

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

## **Методические указания**

по выполнению практических работ  
по дисциплине «Математическая логика и теория алгоритмов»  
для студентов  
Специальности 10.05.03 Информационная безопасность  
автоматизированных систем  
Специализация «Анализ безопасности информационных систем»

Ставрополь

2026

## Содержание

|                                                                                                                                                                                          |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| ВВЕДЕНИЕ .....                                                                                                                                                                           | 3  |
| Практическая работа №1. Высказывания и логические операции.....                                                                                                                          | 4  |
| Практическая работа №2. Логические операции и их свойства.....                                                                                                                           | 10 |
| Практическая работа №3. Логические основы ЭВМ.....                                                                                                                                       | 16 |
| Практическая работа №4. Алгебра логики. Логические функции.....                                                                                                                          | 22 |
| Практическая работа №5. Булевы функции. Таблицы истинности .....                                                                                                                         | 27 |
| Практическая работа №6. Равносильность булевых функций .....                                                                                                                             | 32 |
| Практическая работа №7. Совершенные нормальные формы .....                                                                                                                               | 35 |
| Практическая работа №8. Логика предикатов. Кванторы .....                                                                                                                                | 38 |
| Практическая работа №9. Разработка вычислительных алгоритмов линейной структуры.<br>Разработка алгоритмов разветвляющейся структуры. Разработка алгоритмов циклической<br>структуры..... | 41 |
| Практическая работа №10. Одномерные и многомерные массивы. Сортировка массива методом<br>выбора .....                                                                                    | 49 |
| Практическая работа №11. Сортировка методом вставки. Сортировка методом пузырька. ....                                                                                                   | 53 |
| Практическая работа №12. Рекурсивные функции. ....                                                                                                                                       | 56 |
| Практическая работа №13. Машина Тьюринга. ....                                                                                                                                           | 58 |
| Практическая работа №14. Машина Тьюринга. Построение машин Тьюринга.....                                                                                                                 | 60 |
| Практическая работа №15. Машина Поста. ....                                                                                                                                              | 63 |
| Практическая работа №16. Нормальные алгоритмы Маркова.....                                                                                                                               | 69 |
| РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА .....                                                                                                                                                           | 71 |

## **ВВЕДЕНИЕ**

### **Цель и задачи освоения дисциплины**

Целью освоения дисциплины «Математическая логика и теория алгоритмов» является формирование у будущего специалиста по специальности 10.05.03 Информационная безопасность автоматизированных систем специализация «Анализ безопасности информационных систем» является формирование фундаментальных знаний в области логических исчислений, формальных систем и алгоритмических моделей, необходимых для анализа и обеспечения безопасности информационных систем.

Задачи дисциплины:

- Изучение основ математической логики – пропозициональной и предикатной логики, методов доказательств, теории моделей и теории множеств.
- Освоение формальных языков – понимание синтаксиса, семантики, теории формальных грамматик.
- Анализ алгоритмических моделей – изучение теории алгоритмов (машины Тьюринга, рекурсивные функции, алгоритмическая разрешимость и сложность).

## Практическая работа №1.

### Высказывания и логические операции

**1. Цель занятия:** сформировать понятия: логические высказывания и операции, логические величины; сформировать навыки построения таблиц истинности. Научить студентов приводить сложное логическое выражение к нормальной форме.

#### Задачи занятия:

1. Основные понятия математической логики:
2. Логические выражения.
3. Операциях над логическими выражениями.
4. Правила построения таблиц истинности.
5. Законы математической логики
6. Правила преобразования логических выражений.

## 2. Теоретическое обоснование

**2.1 Основные понятия математической логики** *Математическая логика* — это раздел математики, изучающий высказывания и рассуждения, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

**Логическое высказывание** - это любое утверждение, относительно которого можно сказать истинно оно или ложно, т.е. соответствует оно действительности или нет.

Так, например, утверждение "7 — нечетное число" следует считать высказыванием, так как оно истинное. Утверждение "Рим — столица Франции" тоже высказывание, так как оно ложное.

Очень часто трудно установить истинность высказывания. Так, например, высказывание "площадь поверхности Индийского океана равна 75 млн. кв.км" в одной ситуации можно посчитать ложным, а в другой — истинным. Ложным — так как указанное значение неточное и вообще не является постоянным. Истинным — если рассматривать его как некоторое приближение, приемлемое на практике.

Употребляемые в обычной речи слова и словосочетания "не", "и", "или", "если..., то", "тогда и только тогда" и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**.

Например, из элементарных высказываний "Петров — врач", "Петров — шахматист" при помощи связки "и" можно получить составное высказывание "Петров — врач и шахматист".

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им дают имена. Пусть через **A** обозначено высказывание "Тимур поедет летом на море", а через **B** — высказывание "Тимур летом отправится в горы". Тогда составное высказывание "Тимур летом побывает и на море, и в горах" можно кратко записать как **A и B**. Где "и" — логическая связка, **A**, **B** — логические переменные.

**Логические переменные** — переменные, которые принимают только два значения — "истина" или "ложь", обозначаемые, соответственно, "1" и "0".

Каждая логическая связка рассматривается как **логическая операция** над логическими

высказываниями и имеет свое название и обозначение.

В основе работы современных ЭВМ лежат три основные логические операции НЕ, ИЛИ, И. Иногда эти операции называют "тремя китами машинной логики".

**Операция НЕ**, выражается словом "не", называется **отрицанием** и обозначается знаком  $\neg$  или чертой над логической переменной.

**Операция И**, выражается связкой "и", называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается знаком  $\&$  (может также обозначаться знаками  $\wedge$  или  $*$ ).

**Операция ИЛИ**, выражается связкой "или", называется **дизъюнкцией** (лат. disjunctio — разделение) или логическим сложением и обозначается знаком  $\vee$  или  $+$ . Высказывание  $A \vee B$  ложно тогда и только тогда, когда оба высказывания  $A$  и  $B$  ложны.

Используя операции НЕ и ИЛИ можно получить **операцию ЕСЛИ-ТО**. Которая выражается связками "если...,то", "из...следует", "...влечет...", называется **импликацией** (лат. *implico* — тесно связаны) и обозначается знаком  $\rightarrow$ .

$$A \rightarrow B = \neg A \vee B$$

Используя операции НЕ, ИЛИ, И можно получить **операцию РАВНОСИЛЬНО**. Которая выражаемая связками "тогда и только тогда", "необходимо и достаточно", "... равносильно ...", называется **эквиваленцией** или двойной импликацией и обозначается знаком  $\leftrightarrow$  или знаком  $\sim$ . Высказывание  $A \leftrightarrow B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают.  $A \leftrightarrow B = (\neg A \vee B) \& (\neg B \vee A)$ .

**Приоритет логических операций** по убыванию: операции в скобках, операция отрицания, операция конъюнкции, дизъюнкция, импликация и в последнюю очередь — эквивалентность.

## 2.2 Логические функции и таблицы их истинности.

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической функцией.

**Логической функцией являются:**

- всякая логическая переменная и символы "истина" ("1") и "ложь" ("0");
- составные высказывания:  $\neg A$ ,  $A \& B$ ,  $A \vee B$ ,  $A \rightarrow B$ ,  $A \leftrightarrow B$  (в случае если  $A$  и  $B$  являются логическими функциями).

Используя вышеописанные логические функции можно образовать более сложные функции.

Значения каждой логической функции описывается таблицей истинности. **Таблица истинности** представляет собой таблицу, устанавливающую соответствие между всевозможными наборами значений переменных и значениями функций.

Для  $N$  переменных существует  $2^N$  всевозможных наборов значений переменных.

Например функция, которая содержит две переменные, имеет четыре ( $2^2 = 4$ ) набора значений переменных: (0, 0), (0, 1), (1, 0), (1, 1).

Если функция содержит три переменные, то возможных наборов значений переменных восемь ( $2^3 = 8$ ): (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1).

Составим таблицу истинности основных логических функций двух переменных

Таблица 1 – Таблица истинности элементарных логических функций

| X | Y | $\neg X$ | $X \& Y$ | $X \vee Y$ | $X \rightarrow Y$ | $X \leftrightarrow Y$ |
|---|---|----------|----------|------------|-------------------|-----------------------|
|---|---|----------|----------|------------|-------------------|-----------------------|

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Удобной формой записи при нахождении значений сложной логической функции является таблица, содержащая кроме значений переменных и значений формулы также и значения промежуточных формул.

**Пример 1.** Составить таблицу истинности для функции, которая содержит две переменные  $x$  и  $y$ ,  $\neg x \& y \vee \neg (x \vee y) \vee x$  (таблица 2).

В двух первых столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах — значения промежуточных функций, а в последнем столбце — результат.

Из таблицы 2 видно, что при всех наборах значений переменных  $x$  и  $y$ , функция  $\neg x \& y \vee \neg (x \vee y) \vee x$  принимает значение 1.

Функция, которая принимает значение "истина" для всех наборов значений переменных, называется **тождественно истинной функцией** или **тавтологией**.

Функция, которая принимает значение "ложь" для всех наборов значений переменных, называется **тождественно ложной функцией** или **противоречием**.

Таблица 2 – Таблица истинности для функции  $\neg x \& y \vee \neg (x \vee y) \vee x$ .

| Переменные |     | Промежуточные логические функции |               |            |                   |                          | Результат                                 |
|------------|-----|----------------------------------|---------------|------------|-------------------|--------------------------|-------------------------------------------|
| $X$        | $y$ | $\neg x$                         | $\neg x \& y$ | $x \vee y$ | $\neg (x \vee y)$ | $\neg (x \vee y) \vee x$ | $\neg x \& y \vee \neg (x \vee y) \vee x$ |
| 0          | 0   | 1                                | 0             | 0          | 1                 | 1                        | 1                                         |
| 0          | 1   | 1                                | 1             | 1          | 0                 | 0                        | 1                                         |
| 1          | 0   | 0                                | 0             | 1          | 0                 | 1                        | 1                                         |
| 1          | 1   | 0                                | 0             | 1          | 0                 | 1                        | 1                                         |

Функция, которая принимает для некоторых наборов значений переменных значение "истина", а для других – значение "ложь", называется **выполнимой логической функцией**.

Если две функции  $A$  и  $B$  при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Замена логической функции другой, ей равносильной, называется **равносильным преобразованием** данной формулы.

### 2.3 Основные законы математической логики

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений:

– **переместительный закон:**  $X \vee Y = Y \vee X$ ,

$$X \& Y = Y \& X;$$

- **сочетательный закон:**  $(X \vee Y) \vee Z = X \vee (Y \vee Z),$   
 $(X \& Y) \& Z = X \& (Y \& Z);$
- **распределительный закон:**  $(X \vee Y) \& Z = (X \& Z) \vee (Y \& Z),$   
 $(X \& Y) \vee Z = (X \vee Z) \& (Y \vee Z);$
- **законы де Моргана:**  $\neg (X \vee Y) = \neg X \& \neg Y,$   
 $\neg (X \& Y) = \neg X \vee \neg Y;$
- **закон идемпотенции:**  $X \vee X = X, X \& X = X;$
- **закон поглощения:**  $(X \& Y) \vee X = X, (X \vee Y) \& X = X;$
- **закон склеивания:**  $(X \& Y) \vee (\neg X \& Y) = Y, (X \vee Y) \& (\neg X \vee Y) = Y;$
- **правило операции переменной с ее инверсией:**  $\neg X \vee X = 1, X \& \neg X = 0;$
- **правило операции с константами:**  $X \vee 0 = X,$   
 $X \vee 1 = 1,$   
 $X \& 0 = 0,$   
 $X \& 1 = X;$
- **закон двойного отрицания:**  $\neg \neg X = X.$

## 2.4 Преобразование логических функций

Равносильные преобразования логических функций имеют то же назначение, что и преобразования функций в обычной алгебре. Они служат для упрощения функций или приведения их к определённому виду путем использования основных законов алгебры логики.

**Пример 2.** Упростите логическую функцию  $\neg (x \vee y) \& (x \& \neg y).$

- а) Применяем закон де Моргана:  $\neg (x \vee y) \& (x \& \neg y) = \neg x \& \neg y \& (x \& \neg y).$
- б) Применяем сочетательный закон:  $\neg x \& \neg y \& (x \& \neg y) = \neg x \& x \& \neg y \& \neg y.$  в)

Применяем правило операций переменной с её инверсией:

$$\neg x \& x \& \neg y \& \neg y = 0 \& \neg y \& \neg y.$$

- г) Применяем два раза правило операций с константами:  $0 \& \neg y \& \neg y = 0 \& \neg y = 0.$  **Пример**

**Пример 3.** Упростите логическую функцию  $x \& \neg y \vee \neg x \& y \& z \vee x \& z.$

- а) Вводим вспомогательный логический сомножитель  $(\neg y \vee y): x \& \neg y \vee \neg x \& y \& z \vee x \& z = x \& \neg y \vee \neg x \& y \& z \vee x \& z \& (\neg y \vee y).$
- б) Применяем распределительный закон к последнему слагаемому:  $x \& \neg y \vee \neg x \& y \& z \vee x \& z \& (\neg y \vee y) = x \& \neg y \vee \neg x \& y \& z \vee x \& y \& z \vee x \& \neg y \& z.$
- в) комбинируем два крайних и два средних логических слагаемых:  $x \& \neg y \vee \neg x \& y \& z \vee x \& y \& z \vee x \& \neg y \& z = (x \& \neg y \vee x \& \neg y \& z) \vee (\neg x \& y \& z \vee x \& y \& z).$
- г) используем закон поглощения:  $(x \& \neg y \vee x \& \neg y \& z) \vee (\neg x \& y \& z \vee x \& y \& z) = (x \& \neg y \vee y \& z)$

## 3. Контрольные вопросы

1. Что такое математическая логика, высказывание? Приведите примеры.
2. Дайте определения логической переменной, логической операции, логическому выражению.
3. Назовите логические операции, их обозначение и приведите примеры.
4. Приведите таблицу истинности для логических операций.

## 4. Задания для выполнения

1. Какие правила и законы логики Вы знаете?
2. Какие из предложений являются высказываниями? Определите их истинность или ложность:

1. Петербург – столица нашей Родины.
2. Собака – хищное животное.
3. Я поступил в университет.
4. Как твои дела?
5. Мы сегодня встретимся?
6. Летом я отдыхал на море. Какое из предложений является высказыванием: а) Не можете ли вы передать мне соль?
7. Некоторые лекарства опаснее самих болезней. в) Сегодня солнечно

### Имя и значение

1. Какие из предложений 1) – 24) истинны, а какие ложны?

- 1) 1 – число;
- 2) “1” – число;
- 3) 1 – знак числа;
- 4) “1” – знак числа;
- 5)  $x$  – число;
- 6)  $x$  – переменная;
- 7)  $x$  – знак числа;
- 8)  $x$  – знак переменной;
- 9)  $x$  – переменное число;
- 10)  $x$  – неизвестное число;
- 11) “ $x$ ” – число;
- 12) “ $x$ ” – переменная;
- 13) “ $x$ ” – знак переменной;
- 14) “ $x$ ” – знак числа;
- 15) “ $x$ ” – переменное число;
- 16) “ $x$ ” – неизвестное число;
- 17)  $\emptyset$  – пустое множество;
- 18) “ $\emptyset$ ” – пустое множество;
- 19)  $\emptyset$  – знак пустого множества;
- 20) “ $\emptyset$ ” – знак пустого множества;

2. В каждом из следующих предложений имеется, по крайней мере, одна ошибка, связанная с неправильным употреблением языка. Постарайтесь найти как можно больше ошибок.

- 1) Множество  $\{\emptyset\}$  не пусто, в нем есть один элемент – знак пустого множества.
- 2) Число 144 оканчивается цифрой 4.
- 3) Равенство, содержащее неизвестное число, обозначенное буквой, называется уравнением.
- 4) Корнем уравнения называется число, при подстановке которого в уравнение получается верное равенство.
- 5) Число “3” является корнем уравнения  $x^2 - 9 = 0$ .
- 6) Выражение  $2 + 7$  является корнем уравнения  $x^2 - 4x - 3 = 0$ .
- 7) Рациональным числом называется выражение вида  $\frac{m}{n}$ , где  $m$  и  $n$  – целые числа и  $n \neq 0$ .



8) Линейной функцией называется выражение вида  $ax + b$ , где  $a \neq 0$  и  $b$  – числа,  $x$  – переменная.

9) При  $\alpha = 2\pi$  выражение  $\operatorname{tg} \alpha$  не существует.

10) В теории действительных чисел доказывается теорема 2 есть иррациональное число.

3. Составьте отрицания к данным высказываниям:

а) Все дни в августе были солнечными

б) Не все птицы летают

в) Все растения съедобные

4. Переведите данные высказывания на язык логики.

а) Дождь неожиданно начался и быстро закончился.

б) Обычно в 6 часов я иду гулять с собакой или смотрю телевизор.

в) Если завтра будет холодно, я не пойду гулять.

#### **Составные высказывания**

1. Есть два простых высказывания:  $A$  –

«Число 10 – четное»;

$B$  – «Волк – травоядное животное».

Составить из них все возможные составные высказывания и определите их истинность или ложность. Результаты сведите в таблицу:

| $A \& B$ | $A \vee B$ | $\neg A$ | $\neg B$ | $A \rightarrow B$ | $A \leftrightarrow B$ |
|----------|------------|----------|----------|-------------------|-----------------------|
|          |            |          |          |                   |                       |

2. Постройте таблицу истинности для логического выражения  $A \& B \vee \neg A \& \neg B$ .

3. Упростите выражение:  $F = \neg(A \& B) \vee \neg(B \vee C)$ .

## Практическая работа №2. Логические операции и их свойства

**1. Цель занятия:** сформировать понятия: высказывания и логических операций, свойств логических операций. Сформировать навыки построения таблиц истинности. Научить студентов приводить сложное логическое выражение к нормальной форме.

### **Задачи занятия:**

Данное практическое занятие содержит информацию об основных понятиях математической логики: логических выражениях и операциях над ними, логических операциях и их свойствах, правилах построения таблицы истинности для логических выражений о законах логики,

## **2. Теоретическое обоснование**

**2.1 Основные понятия математической логики** *Математическая логика* — это раздел математики, изучающий высказывания и рассуждения, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

**Логическое высказывание** - это любое утверждение, относительно которого можно сказать истинно оно или ложно, т.е. соответствует оно действительности или нет. Основным (неопределяемым) понятием математической логики является понятие «простого высказывания». Под высказыванием обычно понимают – повествовательное предложение, которое является либо истинным, либо ложным, но не тем и другим сразу. Логика высказываний (пропозициональная логика) изучает правильные способы рассуждений о высказываниях.

Употребляемые в обычной речи слова и словосочетания "*не*", "*и*", "*или*", "*если..., то*", "*тогда и только тогда*" и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**.

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

**Логические переменные** – переменные, которые принимают только два значения – "истина" или "ложь", обозначаемые, соответственно, "*1*" и "*0*".

Каждая логическая связка рассматривается как **логическая операция** над логическими высказываниями и имеет свое название и обозначение.

В основе работы современных ЭВМ лежат три основные логические операции НЕ, ИЛИ, И. Иногда эти операции называют "тремя китами машинной логики".

В алгебре логики все высказывания рассматриваются только с точки зрения их логического значения, а от их житейского содержания отвлекаются. Считается, что каждое высказывание либо истинно, либо ложно и ни одно высказывание не может быть одновременно истинным и ложным

### **2.2 Логические операции над высказываниями.**

**Операция НЕ**, выражается словом "*не*", называется **отрицанием** и обозначается знаком  $\neg$  или чертой над логической переменной.

**Операция И**, выражается связкой "*и*", называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается знаком  $\&$  (может также обозначаться знаками  $\wedge$  или  $*$ ).

**Операция ИЛИ**, выражается связкой "*или*", называется **дизъюнкцией** (лат. disjunctio — разделение) или логическим сложением и обозначается знаком  $\vee$  или  $+$ . Высказывание  $A \vee B$  ложно тогда и только тогда, когда оба высказывания  $A$  и  $B$  ложны.

Используя операции НЕ и ИЛИ можно получить **операцию ЕСЛИ-ТО**. Которая выражается связками "*если..., то*", "*из... следует*", "*...влечет...*", называется **импликацией** (лат. *implico* — тесно связаны) и обозначается знаком  $\rightarrow$ .  
 $A \rightarrow B = \neg A \vee B$

Используя операции НЕ, ИЛИ, И можно получить **операцию РАВНОСИЛЬНО**. Которая выражаемая связками "*тогда и только тогда*", "*необходимо и достаточно*", "*... равносильно ...*", называется **эквиваленцией** или двойной импликацией и обозначается знаком  $\leftrightarrow$  или знаком  $\sim$ . Высказывание  $A \leftrightarrow B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают.  $A \leftrightarrow B = (\neg A \vee B) \& (\neg B \vee A)$ .

**Приоритет логических операций** по убыванию: операции в скобках, операция отрицания, операция конъюнкции, дизъюнкция, импликация и в последнюю очередь — эквивалентность.

### 2.3 Логические функции и таблицы их истинности.

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической функцией.

**Логической функцией являются:**

- всякая логическая переменная и символы "истина" ("1") и "ложь" ("0");
- составные высказывания:  $\neg A$ ,  $A \& B$ ,  $A \vee B$ ,  $A \rightarrow B$ ,  $A \leftrightarrow B$  (в случае если  $A$  и  $B$  являются логическими функциями).

Введём основные операции над высказываниями. Пусть даны два произвольных высказывания  $A$  и  $B$ .

Выражение  $A \& B$  означает высказывание, истинное только тогда, когда  $A$  и  $B$  истинны. Такое высказывание называется **конъюнкцией высказывания  $A$  и  $B$** . Символ  $\&$  означает операцию, называемую конъюнкцией. В обычной речи этой операции соответствует соединение высказываний союзом «И», «А», «ДА». Будем считать, что если  $A$ ,  $B$  истинны, то они соответственно принимают значение 1, ложно — 0.

Таблица истинности данной операции имеет вид

| A | B | $A \& B$ |
|---|---|----------|
| 0 | 0 | 0        |
| 1 | 0 | 0        |
| 0 | 1 | 0        |
| 1 | 1 | 1        |

Символы  $\&$  или  $\&$  означают логические операции над высказываниями, которые называются **конъюнкцией**.

Выражение  $A \vee B$  означает высказывание, истинное, когда, по крайней мере одно из высказываний  $A$  или  $B$  истинно.

Такое высказывание называется *дизъюнкцией высказываний* А и В. Символ  $\vee$  означает операцию, называемой *дизъюнкцией*. В обычной речи этой операции соответствует соединение высказыванием связкой «ИЛИ».

Таблица истинности

| А | В | $A \vee B$ |
|---|---|------------|
| 0 | 0 | 0          |
| 1 | 0 | 1          |
| 0 | 1 | 1          |
| 1 | 1 | 1          |

Выражение  $A > B$  означает высказывание, которое ложно тогда и только тогда, когда А истинно, а В ложно. Такое высказывание называется *импликацией высказываний* А и В. Символ  $>$  означает операцию, называемую *импликацией*. Читается «А влечет В» или «если А, то В»

Таблица истинности  $(A > B) = (\bar{A} \vee B)$

| А | В | $A > B$ |
|---|---|---------|
| 0 | 0 | 1       |
| 0 | 1 | 1       |
| 1 | 0 | 0       |
| 1 | 1 | 1       |

Выражение  $A \sim B$  означает высказывание, которое истинно тогда и только тогда, когда А и В оба истинны, или оба ложны. Такое высказывание называется *эквивалентностью*. В обычной речи этой операции соответствует соединение высказывания «тогда и только тогда, когда».

Таблица истинности  $(AB \vee \bar{A}\bar{B}) = A \sim B$

| А | В | $A \sim B$ |
|---|---|------------|
| 0 | 0 | 1          |
| 0 | 1 | 0          |
| 1 | 0 | 0          |
| 1 | 1 | 1          |

Выражение  $\bar{A}$  означает высказывание, которое истинно, когда А ложно и ложно, когда А – истинно. Такое высказывание называется *отрицанием высказывания А*. Черточка над буквой означает операцию, называемую *отрицанием*.

В обычной речи этой операции соответствует образование нового высказывания В с помощью частицы «НЕ».

| А | В |
|---|---|
| 0 | 1 |
| 1 | 0 |

Операцией *неравнозначности (равноименности)* высказываний А, В называют составное высказывание, обозначаемое  $A \oplus B$ , которое истинно тогда и только тогда, когда значение истинности высказываний А, В противоположны и ложно в противном случае, что отражается таблицей истинности

| А | В | $A \oplus B$ |
|---|---|--------------|
| 0 | 0 | 0            |

|   |   |   |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

В составленном высказывании порядок выполнения логических операций определяется круглыми скобками, а при их отсутствии сначала выполняется *отрицание*, а затем *конъюнкция*, далее *дизъюнкция*, а потом все остальное. При отсутствии скобок порядок операции совпадает с порядком их перечисления.

## 2.4 Таблицы истинности.

Доказательством эквивалентности двух высказываний служит совпадение их таблиц истинности.

Например, определим значение истинности составного высказывания

$$D = A \& (A \& B > \bar{C}) \vee \bar{A} \& B$$

при  $A=0, B=1, C=1$

$$A \& B = 0, A \& B > C = 1$$

$$A \& (A \& B > C) = 0$$

$$\bar{A} \& B = 1,$$

$$D = 1.$$

Рассмотрим обратную задачу. Пусть  $D=0$ . Необходимо найти хотя бы один набор значений высказываний  $A, B, C$ , при которых  $D=0$ .

В составленном высказывании «D» последней логической операции является дизъюнкция, поэтому составные высказывания

$$A \& (A \& B > C) = 0$$

$$\bar{A} \& B = 0.$$



Для того чтобы

$$A \& (A \& B > \bar{C}) = 0, \text{ необходимо чтобы или } A = 0, \text{ или } (A \& B > \bar{C}) = 0$$

Рассмотрим случай, когда

$$(A \& B > \bar{C}) = 0.$$

Последней операцией здесь является импликация, равная 0 в единственном случае, когда  $(A \& B) = 1$ , а  $\bar{C} = 0$ . Отсюда имеем  $C=1, A=1, B=1$ . Ясно, что при  $A=1, B=1$  составное высказывание  $\bar{A} \& B = 0$ . Таким образом требуемый набор значений  $A, B, C$  определен.

## 3. Задания для выполнения

### 1. Высказывания

1. Какое из предложений является высказыванием:

- Не можете ли вы передать мне соль?
- Некоторые лекарства опаснее самих болезней.
- Сегодня солнечно.

2. Составьте отрицания к данным высказываниям:

- Все дни в августе были солнечными
- Не все птицы летают
- Все растения съедобные

Переведите данные высказывания на язык логики.

- Дождь неожиданно начался и быстро закончился.

- б) Обычно в 6 часов я иду гулять с собакой или смотрю телевизор.  
 в) Если завтра будет холодно, я не пойду гулять.
3. Какие правила и законы логики Вы знаете?
4. Какие из предложений являются высказываниями? Определите их истинность или ложность:
1. Петербург – столица нашей Родины.
  2. Собака – хищное животное.
  3. Я поступил в университет.
  4. Как твои дела?
  5. Мы сегодня встретимся?
  6. Летом я отдыхал на море.
5. Есть два простых высказывания: А – «Число 10 – четное»;  
 В – «Волк – травоядное животное».
- Составить из них все возможные составные высказывания и определите их истинность или ложность. Результаты сведите в таблицу:

| A & B | A ∨ B | ¬A | ¬B | A → B | A ↔ B |
|-------|-------|----|----|-------|-------|
|       |       |    |    |       |       |

## 2. Таблицы истинности.

1. Построить таблицы истинности для следующих схем:

- 1)  $(X \rightarrow Y) \vee (X \rightarrow Y \wedge X)$ ;
- 2)  $X \wedge (Y \rightarrow X) \rightarrow \neg X$ ;
- 3)  $((X \wedge \neg Y) \rightarrow Y) \rightarrow (X \rightarrow Y)$ ;
- 4)  $X \rightarrow (Y \rightarrow Z) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z))$ ;
- 5)  $X \wedge (Y \vee \neg X) \wedge ((\neg Y \rightarrow X) \vee Y)$ .

## 3. Тавтологии.

1. Доказать, что следующие высказывательные схемы являются тавтологиями:

- 1)  $(X \rightarrow Y) \vee (Y \rightarrow X)$ ;
- 2)  $((X \rightarrow Y) \rightarrow X) \rightarrow X$ ;
- 3)  $(X \rightarrow Y) \vee (X \rightarrow \neg Y)$ ;
- 4)  $(\neg X \rightarrow \neg Y) \rightarrow ((\neg X \rightarrow Y) \rightarrow X)$ ;
- 5)  $\neg X \rightarrow (X \rightarrow Y)$ ;
- 6)  $X \wedge \neg X \rightarrow (Y \vee Z \rightarrow (Z \rightarrow \neg X))$ ;
- 7)  $X \rightarrow \neg X \leftrightarrow \neg X$ ;
- 8)  $(X \rightarrow Z) \wedge (Y \rightarrow T) \wedge (\neg Z \vee \neg T) \rightarrow \neg X \vee \neg Y$ .

2. Доказать, что следующие высказывательные схемы не являются тавтологиями:

- 1)  $((X \rightarrow Y \wedge Z) \rightarrow (\neg Y \rightarrow \neg X)) \rightarrow \neg Y$ ;
- 2)  $X \vee Y \vee Z \rightarrow (X \vee Y) \wedge (X \vee Y)$ ;
- 3)  $X \vee Y \rightarrow X \wedge Y$ ;
- 4)  $(X \rightarrow Y) \rightarrow (Y \rightarrow X)$ .

## 4. Выполнимость.

1. Доказать, что следующие высказывательные схемы выполнимы:

1)  $\neg X \rightarrow X$ ;

2)  $(X \rightarrow Y) \rightarrow (Y \rightarrow X)$ ;

3)  $(Y \rightarrow X \wedge Z) \wedge (X \vee Z \rightarrow Y)$ ;

4)  $\neg(X \rightarrow \neg X)$ ;

5)  $(X \rightarrow Y) \rightarrow (Y \rightarrow Y)$ ; 6)  $(Y \rightarrow (X \wedge Z)) \wedge \neg((X \vee Z) \rightarrow Y)$ .

#### 5. Совместность

1. Выяснить, выполнимы ли следующие множества высказывательных схем:

1)  $\{X \vee Y \vee Z, X \rightarrow Z, \neg Y\}$ ;

2)  $\{X \vee Y \vee Z, X \rightarrow Z, \neg Z\}$ ;

3)  $\{X \vee Y \vee Z, X \rightarrow Y, \neg Y, \neg Z\}$ ;

4)  $\{X \vee Y, X \rightarrow Y, Y \rightarrow Z\}$ ;

5)  $\{X \vee Y, Y \rightarrow Z, \neg Z\}$ ;

6)  $\{X \vee Y, X \rightarrow Y, Y \rightarrow Z, \neg Z\}$

#### 4. Контрольные вопросы

1. Что такое математическая логика, высказывание? Приведите примеры.

2. Дайте определения логической переменной, логической операции, логическому выражению.

3. Назовите логические операции, их обозначение и приведите примеры.

4. Приведите таблицу истинности для логических операций

## **Практическая работа №3.**

### **Логические основы ЭВМ**

**1. Цель занятия:** сформировать понятия: логические высказывания и операции, логические величины; сформировать навыки построения таблиц истинности. Научить студентов приводить сложное логическое выражение к нормальной форме.

**Задачи занятия:**

Освоить элементы, реализующие базовые логические операции, называемые базовыми логическими элементами или вентилями в ЭВМ. Изучить правила построения логических схем.

## **2. Теоретическое обоснование**

### **2.1 Формы мышления**

**Логика** – это наука о формах и способах мышления. Это учение о способах рассуждений и доказательств.

Законы мира, сущность предметов, общее в них мы познаем посредством абстрактного мышления. Логика позволяет строить формальные модели окружающего мира, отвлекаясь от содержательной стороны.

Мышление всегда осуществляется через понятия, высказывания и умозаключения.

**Понятие** – это форма мышления, которая выделяет существенные признаки предмета или класса предметов, позволяющие отличать их от других.

Например: прямоугольник, проливной дождь, компьютер.

**Высказывание** – это формулировка своего понимания окружающего мира. Высказывание является повествовательным предложением, в котором что-либо утверждается или отрицается.

По поводу высказывания можно сказать, истинно оно или ложно. Истинным будет высказывание, в котором связь понятий правильно отражает свойства и отношения реальных вещей. Ложным высказывание будет в том случае, когда оно противоречит реальной действительности.

Например: истинное высказывание: «Буква «а» – гласная»; ложное высказывание: «Компьютер был изобретен в середине XIX века».

**Умозаключение** – это форма мышления, с помощью которой из одного или нескольких суждений может быть получено новое суждение (знание или вывод).

Например: дано высказывание: «Все углы равнобедренного треугольника равны». Получить высказывание «Этот треугольник равносторонний» путем умозаключений.

Пусть основанием треугольника является сторона  $c$ . Тогда  $a = b$ . Так как в треугольнике все углы равны, следовательно, основанием может быть любая другая сторона, например,  $a$ . Тогда  $b = c$ . Следовательно  $a = b = c$ . Треугольник равносторонний.

### **2.2 Логические выражения и операции**

Алгебра – это наука об общих операциях, аналогичных сложению и умножению, которые выполняются не только над числами, но и над другими математическими объектами, в том числе и над высказыванием. Такая алгебра называется **алгеброй логики**. Алгебра логики отвлекается от смысловой содержательности высказываний и принимает во внимание только истинность или ложность высказывания.

Можно определить понятия логической переменной, логической функции и логической операции.

**Логическая переменная** – это простое высказывание, содержащее только одну мысль. Ее символическое обозначение – латинская буква (например,  $A, B, X, Y$  и т. д.). значениями логической переменной могут быть только константы ИСТИНА и ЛОЖЬ (1 и 0 соответственно).



Составное высказывание – **логическая функция**, которая содержит несколько простых мыслей, соединенных между собой с помощью логических операций. Ее символическое обозначение –  $F(A, B, \dots)$ .

На основании простых высказываний могут быть построены составные высказывания.

**Логические операции** – логическое действие.

Рассмотрим три базовые логические операции – конъюнкцию, дизъюнкцию, и отрицание и дополнительные – импликацию и эквивалентность.

1. **Конъюнкция** – логическое умножение. Обозначение  $A \& B$  или  $A \wedge B$  (союз & соответствует союзу «и» в русском языке, читается А и В).

Например: пусть А – простое высказывание, обозначающее «Число 10 – четное»; В – простое высказывание, обозначающее «Число 10 – отрицательное». Тогда  $A \& B$  – обозначает «Число 10 – четное и отрицательное». Результатом такой операции является ЛОЖЬ.

2. **Дизъюнкция** – логическое сложение. Обозначение  $A \vee B$  (союз V соответствует союзу «или» в русском языке, читается А или В).

Например:  $A \vee B$  обозначает «Число 10 – четное или отрицательное». Результатом такой операции является ИСТИНА.

3. **Инверсия** – отрицание. Обозначение  $\bar{A}$  или  $\neg A$  (символ верхнее подчеркивание и символ  $\neg$  соответствует частице «не» в русском языке)

Например:  $\neg A$  обозначает «Неверно, что число 10 – четное», результат – ЛОЖЬ; или «Неверно, что число 10 – отрицательное», результат – ИСТИНА.

4. **Импликация** – логическое следование. Обозначение  $A \rightarrow B$  (где А – условие, В – следствие). Читается если А, то В.

Например:  $A \rightarrow B$  обозначает «Если число 10 – четное, то оно является отрицательным». Результатом такой операции является ЛОЖЬ.

5. **Эквивалентность** – логическое равенство. Обозначение  $A \equiv B$ , или  $A \leftrightarrow B$ . Читается А тогда и только тогда, когда В.

Например:  $A \leftrightarrow B$  обозначает «Число 10 – четное тогда и только тогда, когда отрицательно». Результатом такой операции является ЛОЖЬ.

Представим таблицу истинности (таблица 4) для вышеперечисленных логических операций. Таблица истинности – это таблица, определяющая значение сложного высказывания при всех возможных значениях простых высказываний.

Таблица 4 – Таблица истинности

| Простое высказывание | Простое высказывание | Конъюнкция | Дизъюнкция | Инверсия | Импликация        | Эквивалентность       |
|----------------------|----------------------|------------|------------|----------|-------------------|-----------------------|
| A                    | B                    | $A \& B$   | $A \vee B$ | $\neg A$ | $A \rightarrow B$ | $A \leftrightarrow B$ |
| 0                    | 0                    | 0          | 0          | 1        | 1                 | 1                     |
| 0                    | 1                    | 0          | 1          |          | 1                 | 0                     |
| 1                    | 0                    | 0          | 1          | 0        | 0                 | 0                     |
| 1                    | 1                    | 1          | 1          |          | 1                 | 1                     |

Примечание:

1. Результат операции «конъюнкция» будет истинным тогда и только тогда, когда оба исходных высказывания истинны.

2. Результат операции «дизъюнкция» будет ложным тогда и только тогда, когда оба исходных высказывания ложны, и истинным в остальных случаях.

3. Результат операции «инверсия» будет ложным, если исходное выражение истинно, и наоборот.

4. Результат операции «импликация» будет ложным тогда и только тогда, когда из истинного основания (А) следует ложное следствие (В).

5. Результат операции «эквивалентность» будет истинным тогда и только тогда, когда оба исходных высказывания одновременно либо ложны, либо истинны.

Если составное высказывание (логическую функцию) выразить в виде формулы, в которую войдут логические переменные и знаки логических операций, то получится **логическое выражение**, значение которого можно вычислить. Значением логического выражения могут быть только ЛОЖЬ или ИСТИНА. При составлении логического выражения необходимо учитывать порядок выполнения логических операций, а именно:

- 1) действия в скобках;
- 2) инверсия, конъюнкция, дизъюнкция, импликация, эквивалентность.

**Контрольный пример.** Записать в виде логического выражения следующее высказывание: «Летом Петя поедет в деревню и, если будет хорошая погода, то он пойдет на рыбалку».

1. Проанализируем составное высказывание.

Оно состоит из следующих простых высказываний: «Петя поедет в деревню», «Будет хорошая погода», «Он пойдет на рыбалку». Обозначим их через логические переменные:

А = Петя поедет в деревню;

В = Будет хорошая погода;

С = Он пойдет на рыбалку.

2. Запишем высказывание в виде логического выражения, учитывая порядок действий. Если необходимо, расставим скобки:

$F = A \& (B \rightarrow C).$

### 2.3 Построение таблицы истинности для логического выражения

При составлении таблицы истинности для логического выражения необходимо:

1. Выяснить количество строк в таблице (вычисляется как  $2^n$ , где  $n$  – количество переменных).

2. Выяснить количество столбцов (определяется как количество переменных + количество логических операций).

3. Установить последовательность выполнения логических операций.

4. Построить таблицу, указывая названия столбцов и возможные наборы значений исходных логических переменных.

5. Заполнить таблицу истинности по столбцам.

**Пример 1.** Построить таблицу истинности для выражения  $F = (A \vee B) \& (\neg A \vee \neg B).$

Количество строк в таблице определяется как  $2^2$  (2 переменных) + 1 (заголовок таблицы) = 5.

Количество столбцов – как 2 логические переменные (А, В) + 5 логических операций (&, V, ¬, →, ↔).

Расставим порядок выполнения операций:

1      5    2    4    3

$(A \vee B) \& (\neg A \vee \neg B)$ .

Построим таблицу истинности для данного логического выражения (таблица 5).

Таблица 5 – Таблица истинности для логического выражения

| A | B | $A \vee B$ | $\neg A$ | $\neg B$ | $(\neg A \vee \neg B)$ | $(A \vee B) \& (\neg A \vee \neg B)$ |
|---|---|------------|----------|----------|------------------------|--------------------------------------|
| 0 | 0 | 0          | 1        | 1        | 1                      | 0                                    |
| 0 | 1 | 1          | 1        | 0        | 1                      | 1                                    |
| 1 | 0 | 1          | 0        | 1        | 1                      | 1                                    |
| 1 | 1 | 1          | 0        | 0        | 0                      | 0                                    |

**Пример 2.** Построить таблицу истинности для логического выражения  $X \vee Y \& \neg Z$ .

Количество строк =  $2^3 + 1 = 9$ .

Количество столбцов = 3 логические переменные + 3 логических операций = 6

Укажем порядок действий:

3    2    1

$X \vee Y \& \neg Z$ .

Нарисуем и заполним таблицу 6:

Таблица 6 – Таблица истинности для логического выражения

| X | Y | Z | $\neg Z$ | $Y \& \neg Z$ | $X \vee Y \& \neg Z$ |
|---|---|---|----------|---------------|----------------------|
| 0 | 0 | 0 | 1        | 0             | 0                    |
| 0 | 0 | 1 | 0        | 0             | 0                    |
| 0 | 1 | 0 | 1        | 1             | 1                    |
| 0 | 1 | 1 | 0        | 0             | 0                    |
| 1 | 0 | 0 | 1        | 0             | 1                    |
| 1 | 0 | 1 | 0        | 0             | 1                    |
| 1 | 1 | 0 | 1        | 1             | 1                    |
| 1 | 1 | 1 | 0        | 0             | 1                    |

## 2.4 Базовые логические элементы. Построение логических схем.

С точки зрения логики электрический ток либо течет, либо не течет; электрический импульс есть или нет; электрическое напряжение есть или нет. Рассмотрим электрические контактные схемы, реализующие логические операции (схемы 1 – 3). На схемах 1 – 3 контакты обозначены латинскими буквами A и B.

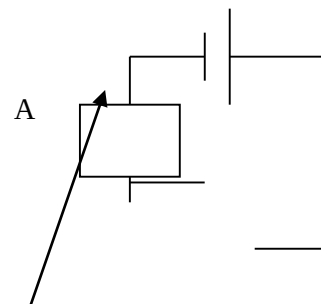
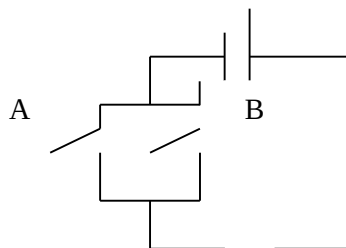
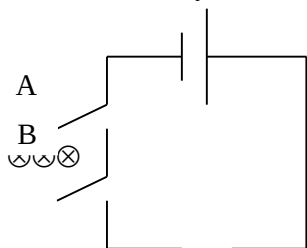


Схема 1 – Конъюнкция

Схема 2 – Дизъюнкция

Схема 3 – Инверсия  
(автоматический ключ)

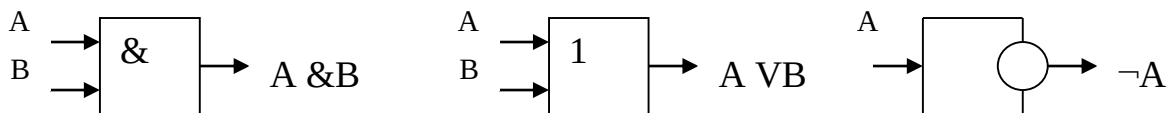


Схема 4 – Конъюнктор

Схема 5 – Дизъюнктор

Схема 6 – Инвертор

Цепь на схеме 1 с последовательным соединением контактов соответствует логической операции «И» и представляется конъюнктором (схема 4). Цепь на схеме 2 с параллельным соединением контактов соответствует логической операции «ИЛИ» и представляется дизъюнктором (схема 5). Цепь на схеме 3 (электромагнитное реле) соответствует логической операции «НЕ» и представляется инвертором (схема 6).

Именно такие электронные схемы нашли свое применение в качестве элементной базы ЭВМ. Элементы, реализующие базовые логические операции, назвали базовыми логическими элементами или **вентильми** и характеризуются они не состоянием контактов, а наличием сигналов на входе и выходе элемента. Их названия и условные обозначения являются стандартными и используются при составлении и описании логических схем компьютера.

Логические схемы необходимо строить из минимально возможного количества элементов, что, в свою очередь, обеспечивает большую скорость работы и увеличивает надежность устройства.

#### Правила построения логических схем:

1. Определить число логических переменных.
2. Определить количество базовых логических операций и их порядок.
3. Изобразить для каждой логической операции соответствующий ей вентиль.
4. Соединить вентили в порядке выполнения логических операций.

**Пример 3.** Пусть  $X = \text{Истина} (1)$ ,  $Y = \text{Ложь} (0)$ . Составьте логическую схему для следующего логического выражения:  $F = XY \vee X$ .

- 1) Две переменные –  $X$  и  $Y$ .
- 2) Две логические операции:  $XY \vee X$  2 1
- 3) Строим схему (рисунок 3).
- 4) Ответ:  $1 \vee 0 \vee 1 = 1$ .

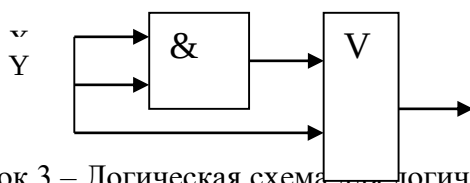


Рисунок 3 – Логическая схема для логического выражения  $F = XY \vee X$

#### 2.5 Задания для выполнения

1. Постройте таблицу истинности для логического выражения  $A \& B \vee \neg A \& \neg B$ .
2. Постройте логическую схему, соответствующую логическому выражению  $F = X \& Y \vee \neg(Y \vee X)$ .
3. Вычислить значения для этого логического выражения, если:  $X = 1$ ,  $Y = 0$ .
4. Постройте таблицу истинности для логических выражений:
  - 1)  $A \& B \vee \neg A \& \neg B$ ,
  - 2)  $(\neg X \vee Z) \wedge (Y \vee Z)$ ;
  - 3)  $X \vee Y \wedge Z$ ;

- 4)  $X \wedge Y \vee Z \wedge T$ ;
- 5)  $(X \wedge \neg Y \vee Z) \wedge (\neg X \vee Z)$ ;
- 6)  $(X \vee Y) \wedge (X \vee Z) \vee \neg Y \wedge (Z \vee \neg Y)$ ;
- 7)  $X \vee Y \vee Z$ ;
- 8)  $(X \vee Y \vee Z) \wedge (X \vee T) \wedge (Z \vee T)$ ;
- 9)  $X \vee Y \vee Z \vee S \vee T$ ;
- 10)  $\neg X \vee X \wedge Y \vee Y \wedge Z \vee Z \wedge T$ ;

5. Пусть  $X$  = Истина (1),  $Y$  = Ложь (0).

Составьте логическую схему для следующего логического выражения:  $F = X \vee Y \wedge X$ . Строим схему (см. рисунок).

Ответ:  $1 \vee 0 \wedge 1 = 1$ .

Составьте логические схемы для логических выражений

- $A \wedge B \vee \neg A \wedge \neg B$ ,
- $(\neg X \vee Z) \wedge (Y \vee Z)$ ;
- $X \vee Y \wedge Z$ ;
- $X \wedge Y \vee Z \wedge T$ ;
- $(X \wedge \neg Y \vee Z) \wedge (\neg X \vee Z)$ ;
- $(X \vee Y) \wedge (X \vee Z) \vee \neg Y \wedge (Z \vee \neg Y)$ ;
- 7)  $X \vee Y \vee Z$ ;

### 3.Контрольные вопросы

1. Какие правила логики Вы знаете?
2. Таблица истинности для логического выражения?
3. Каковы элементы, реализующие базовые логические операции?
4. Назовите правила и последовательность построения логических схем.

## Практическая работа №4.

### Алгебра логики. Логические функции

**1 Цель занятия:** сформировать понятия: логические высказывания и операции, логические величины; сформировать навыки построения таблиц истинности.

#### **Задачи занятия:**

Научить студентов формулировать: понятия алгебры логики, логических функций, приводить сложное логическое выражение к нормальной форме.

#### **2. Теоретическое обоснование**

Рассмотрим множество векторов  $\bar{X} = \{x_1, x_2, \dots, x_n\}$ , пусть координаты этих векторов принимают значения 0 или 1. Таким образом, множество  $X$  состоит из  $2^n$  различных векторов.

Совокупность координат некоторого фиксированного вектора  $\bar{X}^*$  из множества  $\bar{X}$  будем называть двоичным набором или просто набором.

Сопоставим каждому вектору  $X$  число 0 или 1, т.е. произведем однозначное отображение  $X$  на множество  $Y = \{0, 1\}$ .

**2.1 Функцией алгебры логики (ФАЛ)** называется функция, дающая однозначное отображение  $X$  в  $Y$ .

1. Число различных ФАЛ, зависящих от  $n$  аргументов, конечно и равно  $2^{2^n}$ .

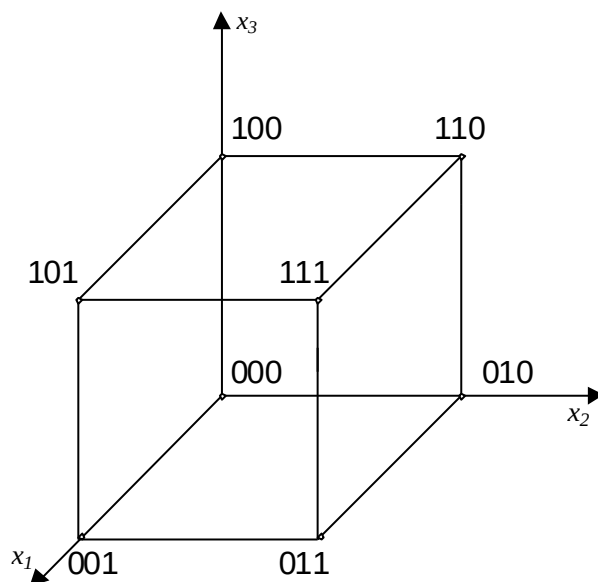
| $x_1, x_2, \dots, x_{n-1}, x_n$ | $f(x_1, x_2, \dots, x_{n-1}, x_n)$ |
|---------------------------------|------------------------------------|
| 0 0 ... 0 0                     | $\alpha_1$                         |
| 0 0 ... 0 1                     | $\alpha_2$                         |
| ... ..                          | ... ..                             |
| 1 1 ... 1 0                     | $\alpha_{2^{n-1}}$                 |
| 1 1 ... 1 1                     | $\alpha_{2^n}$                     |

Если наборам значений аргументов ФАЛ сопоставляет точки  $n$ - мерного пространства, то множество  $2^{2^n}$  наборов определяет множество вершин  $n$ - мерного единичного куба. Таким образом, областью определения ФАЛ, зависящей от  $n$  аргументов, является множество вершин такого куба.

ФАЛ  $f(x_1, x_2, x_3)$  может быть задана геометрически.  $F(101, 110, 010, 011) = 1$ . Значения ФАЛ могут быть заданы не на всех возможных наборах аргументов. На некоторых значения функции могут быть не определены. Такие функции называют не полностью определенными

Если две ФАЛ  $f_1(x_1, x_2, \dots, x_{n-1}, x_n)$  и  $f_2(x_1, x_2, \dots, x_{n-1}, x_n)$  принимают на всех возможных наборах значений аргументов одинаковые значения, то функции  $f_1$  и  $f_2$  называются равными.

ФАЛ  $f(x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{n-1}, x_n)$  существенно зависит от аргумента  $x_i$ , если имеет место соотношение  $f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_{n-1}, x_n) \neq f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_{n-1}, x_n)$  в противном случае



говорят, что от  $x_i$  функция зависит не существенно и  $x_i$  является ее фиктивным аргументом. ФАЛ не изменяется, если к ее аргументам дописать любое число фиктивных аргументов или зачеркнуть те аргументы, которые являются фиктивными.

## 2.2 Элементарные функции алгебры логики

Рассмотрим множество элементарных ФАЛ. Начнем со случая, когда длина слова  $n=0$ . По формуле, определяющей число функций логики, вычисляем, что при  $n=0$ ,  $2^{2^n} = 2$ .

$$f_1=0; \quad f_2=1.$$

Эти две функции совпадают с константами.

В случае  $n=1$  мы имеем две функции, существенно зависящие от аргумента  $x$ , эти две функции определяются таблицей

| $x_1$ | $f_3$ | $f_4$ |
|-------|-------|-------|
| 0     | 0     | 1     |
| 1     | 1     | 0     |

Эти две функции также относят к элементарным, и они записываются следующим образом

$$f_3=x_3; \quad f_4=\neg x_3(\text{читается «не } x\text{»}).$$

Функцию  $f_4$  называют функцией отрицания.

В случае  $n=2$  имеем десять различных функций, существенно зависящих от аргументов  $x_1$  и

$x_2$

|       |       | $x$          | $x$        | $x$          | $x$                 | $x$                | $x$      | $x$            |
|-------|-------|--------------|------------|--------------|---------------------|--------------------|----------|----------------|
|       |       | $1 \vee x_2$ | $1 \& x_2$ | $1 \sim x_2$ | $1 \rightarrow x_2$ | $1 \downarrow x_2$ | $1/x_2$  | $1 \oplus x_2$ |
| $x_1$ | $x_2$ | $f_5$        | $f_6$      | $f_7$        | $f_8$               | $f_9$              | $f_{10}$ | $f_{11}$       |
| 0     | 0     | 0            | 0          | 1            | 1                   | 1                  | 1        | 1              |
| 0     | 1     | 1            | 0          | 0            | 1                   | 0                  | 1        | 0              |
| 1     | 0     | 1            | 0          | 0            | 0                   | 0                  | 1        | 1              |
| 1     | 1     | 1            | 1          | 1            | 1                   | 0                  | 0        | 0              |

Функция  $f_5(x_1, x_2) = x_1 \vee x_2$  (дизъюнкция).

Функция  $f_6(x_1, x_2) = x_1 \& x_2$  (конъюнкция).

Вместо символов  $\&$  часто применяют символ умножения « $\bullet$ » или вообще опускают также, как и в обычной алгебре.

Функция  $f_7$  носит название функции эквивалентности или функции равнозначности  $f_7(x_1, x_2) = x_1 \sim x_2$  и  $f_7(x_1, x_2) = x_1 \equiv x_2$ .

Функция  $f_8$  носит название импликации  $x_1$  в  $x_2$ . Она обозначается  $f_8(x_1, x_2) = x_1 \rightarrow x_2$ .

Функция  $f_9$  носит название функции Вебба (или ее называют еще стрелкой Пирса) и обозначается

$$f_9(x_1, x_2) = x_1 \vee x_2.$$

Функция  $f_{10}$  называется функцией (штрихом) Шеффера и обозначается следующим образом  $f_{10}(x_1, x_2) = x_1 \downarrow x_2$ .

Функция  $f_{11}$  обычно называется функцией сложения по модулю 2.  $f_{11}(x_1, x_2) = x_1 \oplus x_2$ .

Рассмотрение одиннадцати функций позволяют строить новые функции двумя основными способами:

- путем перенумерации аргументов;
- путем подстановки в функцию новых функций вместо аргументов.

Функцию, полученную из функций  $f_1, f_2, f_3, \dots, f_k$  путем применения этих правил будем называть суперпозицией функции  $f_1, f_2, f_3, \dots, f_k$ .

Имея, например, элементарные функции отрицания, дизъюнкции, эквивалентности и импликации, можно составить следующие новые функции ФАЛ.



Используя таблицы, определяющие элементарные функции, можно задать любую функцию ФАЛ, являющуюся суперпозицией этих функций

### 3. Задачи для выполнения

1. Построить таблицы истинности для следующих схем:

- 1)  $(X \rightarrow Y) \vee (X \rightarrow Y \wedge X)$ ;
- 2)  $X \wedge (Y \rightarrow X) \rightarrow \neg X$ ;
- 3)  $((X \wedge \neg Y) \rightarrow Y) \rightarrow (X \rightarrow Y)$ ;
- 4)  $X \rightarrow (Y \rightarrow Z) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z))$ ;
- 5)  $X \wedge (Y \vee \neg X) \wedge ((\neg Y \rightarrow X) \vee Y)$ .

2. Выяснить, является ли знаковосчетание логической функцией:

- 1)  $(X_1 \wedge X_2) X_3 \neg X_4$ ;
- 2)  $(X_1 \wedge X_2) \rightarrow X_3$ ;
- 3)  $((A_2 \rightarrow A_1) \wedge \neg A_3)$ ;
- 4)  $((X_2 \rightarrow X_1) \wedge \neg X_3)$ ;
- 5)  $((\neg X_1) \rightarrow X_2) \rightarrow \neg(X_2 \vee (X_3))$ ;
- 6)  $((X \wedge Y) Z \neg T)$ ;
- 7)  $((X \wedge Y) \rightarrow Z)$ ;
- 8)  $((X \rightarrow Z) \wedge \neg T)$ ;
- 9)  $((\neg T) \rightarrow Y) \rightarrow (\neg(X \vee Y))$ .11

3. Сколько существует способов расстановки скобок в знаковосчетании, чтобы оно стало логической функцией:

- 1)  $X_1 \rightarrow \neg X_2 \vee X_2 \wedge X_3$ ;
- 2)  $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow \neg X_1 \rightarrow \neg X_2$ ?

4. Исключить возможно большее число скобок:

- 1)  $((Y \leftrightarrow (\neg Z \vee (T \wedge X)) \leftrightarrow (Y \rightarrow Y))$ ;
- 2)  $((X \wedge \neg Y) \wedge Z) \vee T$ .

5. Выписать все подсхемы следующих логической функцией:

- 1)  $((X \rightarrow Y) \wedge (Z \rightarrow T)) \rightarrow (\neg Y \vee T)$ ;
- 2)  $((X \rightarrow Y) \rightarrow ((X \rightarrow \neg Y) \rightarrow \neg Y))$ .

### 4. Контрольные вопросы.

1. Назовите логические операции, их обозначение и приведите примеры.

2. Как определить значение логической функции?
3. Как построить более сложные функции, которые могут зависеть от любого числа переменных?
4. Приведите примеры таблиц истинности для логических функций

## Практическая работа №5.

### Булевы функции. Таблицы истинности

**1. Цель занятия:** научить студентов применению в логике математической символики, умению свести операции с логическими заключениями к формальным действиям над символами, формулировать понятия алгебры логики, логических функций и Булевой алгебры.

#### 2. Теоретическое обоснование

**Булевой функцией**  $y = f(x_1, x_2, \dots, x_n)$  от  $n$  переменных  $x_1, x_2, \dots, x_n$  называется любая функция, в которой аргументы и функция могут принимать значение либо 0 либо 1.

Булеву функцию от  $n$  переменных можно задать таблицей истинности, в которой наборы значений аргументов расположены в порядке возрастания их номеров: сначала идет набор, представляющий собой двоичное разложение 0 (этот набор имеет номер 0); затем идет набор, являющийся двоичным разложением 1, потом 2, 3 и т.д.

Отрицание – булева функция одной переменной, которая определяется следующей таблицей истинности

|        |   |   |                   |
|--------|---|---|-------------------|
| $x$    | 0 | 1 | Обозначения       |
| $f(x)$ | 1 | 0 | $\neg x, \bar{x}$ |

Элементарные булевы функции двух переменных приведены в следующих таблицах

Обозначение: 0

1. *Отрицанием* называется булева функция одной переменной, которая определяется следующей таблицей истинности

|        |   |   |
|--------|---|---|
| $x$    | 0 | 1 |
| $f(x)$ | 1 | 0 |

Обозначения:  $\neg x, \bar{x}$ . Запись  $\neg x$  читается «не икс» или «отрицание икс».

*Конъюнкцией* называется булева функция двух переменных, которая определяется следующей таблицей истинности

|           |   |   |   |   |
|-----------|---|---|---|---|
| $x$       | 0 | 0 | 1 | 1 |
| $y$       | 0 | 1 | 0 | 1 |
| $f(x, y)$ | 0 | 0 | 0 | 1 |

Другие названия: логическое умножение (произведение); логическое «и».

Обозначения:  $x \& y$ ,  $x \cdot y$ ,  $x \wedge y$ ,  $\min(x, y)$ .

Запись  $x \& y$  может читаться так: «икс и игрек» или «икс умножить на игрек».

2. *Дизъюнкцией* называется булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 0 | 1 | 1 | 1 |

Другие названия: логическое сложение (сумма); логическое «или».

Обозначения:  $x \vee y$ ,  $\max(x, y)$ .

Запись  $x \vee y$  может читаться так: «икс или игрек» или «сумма икс и игрек».

*Импликацией* называется булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 1 | 1 | 0 | 1 |

Обозначения:  $x \rightarrow y$ ,  $x \Rightarrow y$ ,  $x \supset y$ .

Запись  $x \rightarrow y$  может читаться так: «из икс следует игрек».

*Эквивалентностью* называется булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 1 | 0 | 0 | 1 |

Обозначения:  $x \sim y$ ,  $x \leftrightarrow y$ ,  $x \equiv y$ .

Запись  $x \sim y$  может читаться так: «икс эквивалентно игрек» или «икс равносильно игрек».

*Суммой по модулю 2* называется булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 0 | 1 | 1 | 0 |

Другое название: антиэквивалентность.

Обозначения:  $x \oplus y$ ,  $x + y$ .

Запись  $x \oplus y$  может читаться так: «икс плюс игрек»

*Штрих Шеффера* - это булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 1 | 1 | 1 | 0 |

Другое название: отрицание конъюнкции, логическое «не-и».

Обозначение:  $x|y$ .

Запись  $x|y$  может читаться так: «не икс или не игрек», «икс и игрек несовместны», «икс штрих Шеффера игрек».

*Стрелка Пирса* - это булева функция двух переменных, которая определяется следующей таблицей истинности

|         |   |   |   |   |
|---------|---|---|---|---|
| x       | 0 | 0 | 1 | 1 |
| y       | 0 | 1 | 0 | 1 |
| f(x, y) | 1 | 0 | 0 | 0 |

Другое название: отрицание дизъюнкции, логическое «не-или», функция Вебба.

Обозначение:  $x \downarrow y$ ; для функции Вебба -  $x \vee y$ .

Запись  $x \downarrow y$  может читаться так: «ни икс и ни игрек», «икс стрелка Пирса игрек».

*Замечание.* Символы  $\neg, \wedge, \vee, \rightarrow, \sim, \oplus, \downarrow$  участвующие в обозначениях элементарных функций будем называть связками или операциями.

С помощью суперпозиции элементарных булевых функций можно построить более сложные функции, которые могут зависеть от любого числа переменных. Запись булевых функций через элементарные булевые функции будем называть **формулой**, реализующей данную функцию.

Для более компактной записи сложных функций введем следующие соглашения:

- 1) внешние скобки опускаются;
- 2) сначала производятся операции в скобках;
- 3) считается, что приоритет связок убывает в следующем порядке:

$\neg, \wedge, \downarrow, \vee, \rightarrow, \oplus, \leftrightarrow, (, |, ), , (, ), .$

Для равносильных связок (в скобках) приоритет определим слева направо.

**Пример 1.** Расставить скобки в формулах:

- 1)  $x \vee y \leftrightarrow z \oplus x$ ;
- 2)  $x \downarrow y \vee z$ ;
- 3)  $x \oplus y \leftrightarrow z \rightarrow x \wedge y \vee \neg z$ .

Решение: 1) в формуле  $x \vee y \leftrightarrow z \oplus x$  скобки расставляются следующим образом:  $((x \vee y) \leftrightarrow (z \oplus x))$ ;

2) т.к. операция  $\downarrow$  сильнее операции  $\vee$  согласно замечанию имеем  $((x \downarrow y) \vee z)$ ;

3) в формуле  $(x \oplus y) \leftrightarrow z \rightarrow x \wedge y \vee \neg z$  используя введенное старшинство связок, получаем, что данная формула равносильна  $((x \oplus y) \leftrightarrow (z \rightarrow ((x \wedge y) \vee (\neg z))))$ .

**Пример 2.** Составить таблицы истинности для формулы:

- a)  $x \leftrightarrow \bar{y} \rightarrow (y \oplus x)$ ;

Решение:

- a) используя введенное старшинство связок, получим, что данная формула равносильна  $x \leftrightarrow (y \rightarrow (y \oplus x))$ .

### 3. Задания для выполнения

1. Выясните, в каких случаях приведенные ниже данные противоречивы

1.  $a=1, a \wedge b = 0$ .
2.  $a=0, a \downarrow \bar{b} \leftrightarrow a = 1$ .
3.  $a=0, a \wedge b = 0$
4.  $a=1, a \wedge b = 1$ ,
5.  $a=0, a \oplus b = 1$
6.  $a=0, a \wedge b = 1$ ,
7.  $a=1, a \downarrow b = 0$ ,
8.  $a=1, a \downarrow b = 1$ ,
9.  $a=1, a \vee b = 0$ ,
10.  $a=0, a \leftrightarrow b = 0$ .

2. Найдите логические значения  $x$  и  $y$ , при которых выполняются равенства:

1.  $(1 \rightarrow x) \rightarrow y = 0$ ,
2.  $(x \leftrightarrow 0) \rightarrow y = 0$ ,
3.  $x \vee y = \bar{x}$ ,
4.  $x \leftrightarrow y = 1 \rightarrow y$ ,
5.  $x \downarrow y = y \leftrightarrow x$ ,

$$6. x \vee (y \wedge x) = x,$$

$$7. y \wedge (x \vee y) = y,$$

$$8. \bar{x} \oplus y = 1.$$

3. Определите логическое значение формул:

$$1. x \rightarrow y \vee \bar{x}y, \text{ при } x=1, y=1.$$

$$2. x \wedge (y \leftrightarrow z) \oplus (x \wedge y) \downarrow 0, \text{ при } x = 1, y = 0, z = 0.$$

$$3. x \downarrow (y \rightarrow z) \leftrightarrow (x \vee y) \rightarrow (x \oplus z), \text{ при } x = 0, y = 0, z = 1.$$

4. Составить таблицу истинности данной булевой функции:

$$1. (x \downarrow y) \vee x \rightarrow z,$$

$$2. (xy \rightarrow z) \vee x \oplus y,$$

$$3. (x | y) z \rightarrow \bar{y} \vee x,$$

$$4. (xy \rightarrow z) \vee x \oplus y.$$

$$5. x \vee y \bar{z} \rightarrow xy$$

#### 4. Контрольные вопросы

1. Назовите логические операции, их обозначение и приведите примеры.
2. Приведите примеры таблиц истинности для логических операций.
3. Как определить значение логической формулы?
4. Как построить более сложные функции, которые могут зависеть от любого числа переменных?
5. Свойства элементарных булевых функций?
6. Таблицы истинности для элементарных булевых функций?

## Практическая работа №6. Равносильность булевых функций

**1. Цель занятия:** научить студентов применению основных эквивалентностей булевых функций для: доказательства равносильности высказывательных схем (или их неравносильности), преобразования их к возможно более простой форме.

### 2. Теоретическое обоснование

Булевы функции  $f_1$  и  $f_2$  называются **эквивалентными**, если на всяком наборе  $(a_1, a_2, \dots, a_n)$  нулей и единиц значения функций совпадают. Обозначение эквивалентных (равносильных) функций следующее:  $f_1 = f_2$  или  $f_1 \equiv f_2$ .

**2.1 Основные эквивалентности** (ниже  $H, H_1, H_2, H_3 \dots$  означают некоторые булевы функции):

1. Закон двойного отрицания:

$$H = \overline{\overline{H}}.$$

2. Идемпотентность:

$$HH = H, H \vee H = H.$$

3. Коммутативность:

$$H_1 * H_2 = H_2 * H_1, \text{ где символ } * \text{ означает одну из связок } \&, \vee, \oplus, \sim, |, \downarrow.$$

4. Ассоциативность:

$$H_1 * (H_2 * H_3) = (H_1 * H_2) * H_3, \text{ где символ } * \text{ означает одну из связок } \&, \vee, \oplus, \sim.$$

5. Дистрибутивность:

$$H_1 \& (H_2 \vee H_3) = (H_1 \& H_2) \vee (H_1 \& H_3);$$

$$H_1 \vee (H_2 \& H_3) = (H_1 \vee H_2) \& (H_1 \vee H_3);$$

$$H_1 \& (H_2 \oplus H_3) = (H_1 \& H_2) \oplus (H_1 \& H_3).$$

6. Законы де Моргана:

$$\overline{H_1 \& H_2 \& \dots \& H_n} = \overline{H_1} \vee \overline{H_2} \vee \dots \vee \overline{H_n},$$

$$\overline{H_1 \vee H_2 \vee \dots \vee H_n} = \overline{H_1} \& \overline{H_2} \& \dots \& \overline{H_n}.$$

7. Правила поглощения:

$$H_1 \vee (H_2 \& H_3) = H_1, \quad H_1 \& (H_2 \vee H_3) = H_1$$

8. Законы склеивания:

$$H_1 \& H_2 \vee H_1 \& \overline{H_2} = H_1$$

$$(H_1 \vee H_2) \& (H_1 \vee \overline{H_2}) = H_1.$$

9. Законы инверсий:

$$H \vee \overline{H} = 1, \quad H \& \overline{H} = 0.$$

10. Правила операций с константами:

$$H \vee 1 = 1, \quad H \& 1 = H, \quad H \vee 0 = H, \quad H \& 0 = 0.$$

Все приведенные соотношения легко проверяются по таблицам истинности.

**Пример 1.** Упростить формулу, используя основные законы эквивалентности.

$$1) \quad x\overline{z}\vee\overline{x}\vee\overline{x}\overline{z}\vee\overline{y}z = \overline{z}\vee\overline{x}\vee\overline{y}z = \overline{z}\vee\overline{y}z\vee\overline{x}\vee\overline{y}z = \overline{z}\vee\overline{y}\vee\overline{x}.$$

**Пример 2.** Доказать равносильность  $(x \vee \acute{o}) \wedge (\grave{o} \vee \overline{\acute{o}}) \equiv \grave{o}$ .



$$(x \vee \acute{o}) \wedge (\tilde{o} \vee \overline{\acute{o}}) \equiv \tilde{o} \wedge x \vee \acute{o} \wedge \tilde{o} \vee \tilde{o} \wedge \overline{\acute{o}} \vee \acute{o} \wedge \overline{\acute{o}} \equiv \acute{o} \equiv x \vee \acute{o} \wedge \tilde{o} \vee \tilde{o} \wedge \overline{\acute{o}} \equiv \\ \equiv \tilde{o} \wedge (1 \vee \acute{o} \vee \overline{\acute{o}}) \equiv \tilde{o} \wedge (1 \vee 1) \equiv \tilde{o}.$$

### 3. Задачи для выполнения

1. Доказать равносильность высказывательных схем:

- 1)  $X \rightarrow Y$  и  $Y' \rightarrow X'$  ;
- 2)  $(X' \rightarrow Y) \rightarrow Y$  и  $X \rightarrow Y$ ;
- 3)  $(X \rightarrow Y') \rightarrow X'$  и  $X \rightarrow Y$ ;
- 4)  $X' \vee YZ \vee XY$  и  $X \rightarrow Y$ ;
- 5)  $X \leftrightarrow Y$  и  $XY \vee X' Y'$  ;
- 6)  $X \leftrightarrow Y$  и  $(X \vee Y')(X' \vee Y)$ ;
- 7)  $(X \rightarrow Y)(Y \rightarrow X)$  и  $X \leftrightarrow Y$ ;
- 8)  $XY' \vee X' Z \vee YZ'$  и  $X' Y \vee Y' Z \vee XZ'$  ; 13
- 9)  $X \rightarrow (Y \rightarrow Z)$  и  $Y \rightarrow (X \rightarrow Z)$ ;
- 10)  $X \rightarrow (Y \rightarrow Z)$  и  $XY \rightarrow Z$ ;

2. Доказать неравносильность высказывательных схем:

- 1)  $X \rightarrow Y$  и  $Y \rightarrow X$ ;
- 2)  $X(X \rightarrow Y)$  и  $Y$ ;
- 3)  $X' Y \vee XY$  и  $XY$ ;
- 4)  $X \vee XY' \vee X' Y'$  и  $X \vee Y$ ;
- 5)  $XY \vee X' Y \vee XY'$  и  $XY \vee X' Y'$  ;
- 6)  $X(Y \vee Z)$  и  $X \vee YZ$ ;
- 7)  $(X \rightarrow Y) \rightarrow Z$  и  $X \rightarrow (Y \rightarrow Z)$ ;
- 8)  $Z(X \rightarrow Y)$  и  $ZX \rightarrow ZY$ ;
- 9)  $Z(X \leftrightarrow Y)$  и  $ZX \leftrightarrow ZY$ ;
- 10)  $(X \vee Y)(X \vee Y')(X' \vee Y)(X' \vee Y')$  и  $X' Y$ ;

3. Преобразовать к возможно более простой форме:

- 1)  $\neg(\neg X \vee Y) \rightarrow (X \vee Y \rightarrow X)$ ;
- 2)  $\neg(\neg X \wedge \neg Y) \vee (X \rightarrow Y) \wedge X$ ;
- 3)  $(X \rightarrow Y) \wedge (Y \rightarrow X) \wedge (X \vee Y)$ ;
- 4)  $(X \rightarrow Y) \wedge (Y \rightarrow \neg X) \wedge (Z \rightarrow X)$ ;
- 5)  $X \wedge Z \vee X \wedge \neg Z \vee Y \wedge Z \vee \neg X \wedge Y \wedge Z$ ;
- 6)  $\neg(X \rightarrow Y) \wedge (Y \rightarrow \neg X)$ ;
- 7)  $X \rightarrow X'$  ;
- 8)  $X \leftrightarrow X'$  ;
- 9)  $(X \rightarrow X) \rightarrow X$ ;
- 10)  $X' \rightarrow (X \rightarrow Y)$ ;

4. Найти высказывательную схему, равносильную данной, которая содержит только связи

$\neg$  и  $\wedge$ :

- 1)  $X \vee Y \rightarrow (\neg X \rightarrow Z)$ ;
- 2)  $(\neg X \rightarrow Z) \vee \neg(X \rightarrow Y)$ ;
- 3)  $(X \vee Y \vee Z \rightarrow X) \vee Z$ ;
- 4)  $((X \rightarrow Y) \rightarrow Z) \rightarrow \neg X$ ;

5)  $X \vee (Y \rightarrow Z) \rightarrow X$ .14

5. Найти высказывательную схему, равносильную данной, которая содержит только связки  $\neg$  и  $\vee$ :

- 1)  $(X \rightarrow Y) \rightarrow Y \wedge Z$ ;
- 2)  $\neg X \wedge \neg Y \rightarrow X \wedge Y$ ;
- 3)  $\neg X \wedge \neg Y \vee Z \rightarrow Z \wedge \neg Y$ ;
- 4)  $((X \rightarrow Y \wedge Z) \rightarrow (\neg Y \rightarrow \neg X)) \rightarrow \neg Y$ ;
- 5)  $(X \rightarrow Y) \wedge (Y \rightarrow Z) \rightarrow (X \rightarrow Z)$ .

6. С помощью равносильных преобразований доказать, что высказывательная схема является противоречием:

- 1)  $(X \rightarrow Y) \wedge (Y \rightarrow X) \wedge (X \wedge \neg Y \vee \neg X \wedge Y)$ ;
- 2)  $(X \wedge \neg Y \rightarrow \neg X \vee X \wedge Y) \wedge (\neg X \vee X \wedge Y \rightarrow X \wedge \neg Y)$ ;
- 3)  $(X \rightarrow Y) \wedge (Y \rightarrow Z) \rightarrow \neg(X \rightarrow Z)$ ;
- 4)  $(X \rightarrow Y) \wedge (X \rightarrow \neg Y) \wedge X$ ;
- 5)  $X \wedge \neg Y \vee X \wedge \neg Z \leftrightarrow (X \rightarrow Y) \wedge (X \rightarrow Z)$ .

7. С помощью равносильных преобразований доказать, что высказывательная схема является тавтологией:

- 1)  $((X \rightarrow Y) \rightarrow X) \rightarrow X$ ;
- 2)  $(X \vee X)' \rightarrow Y$ ;
- 3)  $(X \rightarrow Y) \rightarrow ((X' \rightarrow Y) \rightarrow Y)$ ;
- 4)  $(X \rightarrow Y) \rightarrow ((X \rightarrow Y') \rightarrow X')$ ;
- 5)  $(X \rightarrow (X \rightarrow Y)) \rightarrow (X \rightarrow Y)$ ;
- 6)  $X \vee Y \rightarrow ((X \rightarrow Y) \rightarrow Y)$ ;
- 7)  $XY \vee XY' \vee X'Y \vee X'Y'$ ;
- 8)  $X \vee (XY' \rightarrow X' \vee Y')(X \rightarrow Y')$ ;
- 9)  $(X \rightarrow Y) \rightarrow ((Y \rightarrow Z) \rightarrow (X \rightarrow Z))$ ;
- 10)  $(Y \rightarrow Z) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z))$ ;

#### 4.Контрольные вопросы

1. Какие булевы функции называются эквивалентными?
2. Основные эквивалентности булевых функций?
3. Как доказать равносильность (не равносильность) высказывательных схем?
4. Как преобразовать к возможно более простой форме высказывательные схемы?
5. Проверить с помощью таблиц истинности, справедлив ли закон дистрибутивности а)  $|$  относительно  $\downarrow$ ; б)  $\supset$  относительно  $|$ ; в)  $\equiv$  относительно  $\supset$ ; г)  $\downarrow$  относительно  $\equiv$ ; е)  $|$  относительно  $\supset$ ?
6. Проверить с помощью таблиц истинности ассоциативность бинарных связок: а) импликации; б) штриха Шеффера; в) эквиваленции; г) стрелки Пирса?

## Практическая работа №7. Совершенные нормальные формы

**1. Цель занятия:** рассмотрение вопросов: аналитической записи функций алгебры логики, вопросов создания их совершенных конъюнктивных, дизъюнктивных и полиномиальных нормальных форм.

### 2. Теоретическое обоснование

Любая таблично заданная функция ФАЛ может быть задана в следующей аналитической форме

$$(1) \quad f(x_1, \dots, x_n) = F_{i_1} \vee F_{i_2} \vee \dots \vee F_{i_k} = \bigvee_{i_j \in T_1} F_{i_j},$$

где  $T_1$  – множество номеров наборов, на которых функция обращается в единицу.

Мы воспользовались дизъюнкцией характеристических функций. Выясним, нельзя ли вместо дизъюнкции использовать какую-либо другую ФАЛ.

Любая таблично заданная ФАЛ может быть задана в следующей аналитической форме

$$(2) \quad f(x_1, \dots, x_n) = \Phi_{i_1} \& \Phi_{i_2} \& \dots \& \Phi_{i_k} = \big\&_{i_k \in T_0} \Phi_{i_k},$$

где  $T_0$  – множество номеров наборов, на которых функция  $f$  обращается в нуль.

Представление функции в виде (2) называют конъюнктивным представлением.

Можно утверждать, что любая функция алгебры логики, кроме нуля, может быть представлена в следующей форме:

$$(3) \quad f(x_1, x_2, \dots, x_n) = \bigvee_1 x_1^{\alpha_1} \& x_2^{\alpha_2} \& \dots \& x_n^{\alpha_n}.$$

При этом дизъюнкция в правой части берется только по тем номерам наборов аргументов, по которым функция, заданная таблично, обращается в единицу.

Представление ФАЛ в виде (3) называют совершенной дизъюнктивной нормальной формой этой функции (СДНФ).

Теорема позволяет сформулировать алгоритм перехода от табличного задания функции к записи этой функции в виде СДНФ. Этот алгоритм формулируется следующим образом:

1. Выбрать в таблице задания функции все наборы аргументов, на которых функция обращается в единицу.

2. Выписать конъюнкции, соответствующие этим наборам аргументов. При этом, если аргумент  $x_i$  входит в данный набор как «1», он вписывается в конъюнкцию, соответствующую данному набору, без изменения. Если же  $x_i$  входит в данный набор как «0», то в соответствующую конъюнкцию вписывается его отрицание.

3. Все полученные конъюнкции соединяются между собой знаками дизъюнкции.

Итак, можно отметить необходимые и достаточные условия совершенной нормальной дизъюнктивной формулы. СДНФ формулы  $f(x_1, x_2, \dots, x_n)$ , содержащие  $n$  различных переменных, обладают следующими свойствами:

А) в ней нет двух одинаковых слагаемых;

Б) ни одно слагаемое не содержит двух одинаковых множителей;

В) никакое слагаемое не содержит переменную вместе с ее отрицанием;

Г) в каждом слагаемом содержится в качестве множителя либо переменная  $x_i$ , либо ее отрицание  $\bar{x}_i$ , где  $i=1, 2, \dots, n$ .

Д) число множителей равно числу  $n$ .

Если вместо (1) воспользоваться соотношением (2), получим полиномиальную совершенную нормальную функцию (ПСНФ).

Так для функции, рассмотренной в примере, ПСНФ имеет следующий вид:

$$f(x_1, x_2, x_3) = \bar{x}_1 \& x_2 \& x_3 \oplus x_1 \& \bar{x}_2 \& \bar{x}_3 \oplus x_1 \& x_2 \& \bar{x}_3.$$

Кроме представления ФАЛ в СДНФ, возможно представление ее в некоторой другой форме, двойственной по отношению к СДНФ.

Любая ФАЛ, кроме единицы, может быть представлена в следующей форме:

$$(4.8) \quad f(x_1, x_2, x_3) = \&_0 (\bar{x}_1^{\alpha_1} \vee \bar{x}_2^{\alpha_2} \vee \dots \vee \bar{x}_n^{\alpha_n}).$$

Символ  $\&_0$  означает, что конъюнкция берется только по тем наборам,  $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$ , для которых выполняется равенство:

$$f(x_1, x_2, x_3) = 0.$$

Представление ФАЛ в виде (8) носит название совершенной конъюнктивной нормальной формы (СКНФ).

Из теоремы вытекает алгоритм построения СКНФ:

1. Выбрать в таблице задания функции все наборы аргументов, на которых функция обращается в нуль.
2. Выписать дизъюнкции, соответствующие этим наборам аргументов. При этом, если аргумент  $x_i$  входит в данный набор как «0», он вписывается без изменения в дизъюнкцию, соответствующую данному набору. Если же  $x_i$  входит в данный набор как «1», то в соответствующую дизъюнкцию вписывается его отрицание.
3. Все полученные дизъюнкции соединяются между собой знаками конъюнкций. Аналогично СДНФ, можно отметить необходимые и достаточные условия совершенной нормальной конъюнктивной формулы. СКНФ формулы  $f(x_1, x_2, \dots, x_n)$ , содержащие  $n$  различных переменных, обладают свойствами:
  - А) в ней нет двух одинаковых множителей;
  - Б) ни один множитель не содержит двух одинаковых слагаемых;
  - В) ни один множитель не содержит переменной вместе с ее отрицанием;
  - Г) каждый множитель содержит в качестве слагаемого или  $x_i$ , или ее отрицание  $\bar{x}_i$ , где  $i = 1, 2, \dots, n$ .
  - Д) число слагаемых равно числу  $n$ .

### 3. Задачи для выполнения

1. Равносильными преобразованиями привести высказывательную схему к СДНФ:

- 1)  $(\neg X \vee Z) \wedge (Y \vee Z)$ ;
- 2)  $X \wedge Y \vee Y \wedge Z$ ;
- 3)  $X \vee Y \wedge Z$ ;
- 4)  $X \wedge Y \vee Z \wedge T$ ;
- 5)  $(X \wedge \neg Y \vee Z) \wedge (\neg X \vee Z)$ ;
- 6)  $(X \vee Y) \wedge (X \vee Z) \vee \neg Y \wedge (Z \vee \neg Y)$ ;
- 7)  $X \vee Y \vee Z$ ;
- 8)  $(X \vee Y \vee Z) \wedge (X \vee T) \wedge (Z \vee T)$ ;
- 9)  $X \vee Y \vee Z \vee S \vee T$ ;

$$10) \neg X \vee X \wedge Y \vee Y \wedge Z \vee Z \wedge T;$$

2. Равносильными преобразованиями привести высказывательную схему к СКНФ:

$$1) \neg X \wedge Z \vee Y \wedge Z;$$

$$2) (X \vee Y) \wedge Z;$$

$$3) (\neg X \vee Y) \wedge (X \vee Z);$$

$$4) \neg X \wedge Y \vee Z \wedge T;$$

$$5) X \wedge Y \wedge Z \vee T;$$

$$6) X \wedge Y \wedge Z;$$

$$7) X \wedge Y \vee Y \wedge Z \vee Z \wedge T;$$

$$8) X \vee Y \vee \neg Z \wedge T;$$

$$9) X \wedge Y \vee Z;$$

$$10) X \wedge Y \wedge Z \wedge T;$$

$$11) (X \vee \neg Y) \wedge (\neg X \vee Y \vee Z) \wedge \neg Z.$$

3. Равносильными преобразованиями привести высказывательную схему к ДНФ:

$$1) \neg(X \vee Z) \wedge (X \rightarrow Y);$$

$$2) (X \leftrightarrow Y) \wedge \neg(Z \rightarrow T);$$

$$3) (X \vee Y \wedge \neg Z) \wedge (X \vee Z);$$

$$4) ((X \rightarrow Y) \rightarrow (Z \rightarrow \neg X)) \rightarrow (\neg Y \rightarrow \neg Z);$$

$$5) (X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \rightarrow \neg Z) \rightarrow (X \rightarrow \neg Y)).$$

4. Равносильными преобразованиями привести высказывательные схемы из задания 3 к КНФ

5. Равносильными преобразованиями привести высказывательные схемы из задания 3, а также следующие три схемы, к СДНФ:

6. Равносильными преобразованиями привести высказывательные схемы из задания 3, а также следующие три схемы, к СКНФ:

$$6) (X \vee Y) \wedge (Y \vee Z) \wedge (Z \vee T);$$

$$7) X \wedge (Y \vee \neg Z) \wedge (X \vee Y \vee Z);$$

$$8) \neg(X \wedge (Y \vee Z)) \rightarrow X \wedge Y \vee Z.18.$$

7. Для каждой из следующих высказывательных схем построить равносильную ей ДНФ и КНФ:

$$1) ((X \rightarrow Y) \rightarrow (Z \rightarrow \neg X)) \rightarrow (\neg Y \rightarrow \neg Z);$$

$$2) (((X \rightarrow Y) \rightarrow \neg X) \rightarrow \neg Y) \rightarrow \neg Z \rightarrow Z;$$

$$3) (X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \rightarrow \neg Z) \rightarrow (X \rightarrow \neg Y)).$$

#### 4.Контрольные вопросы

1. Записать конъюнктивное представление функций алгебры логики.
2. Записать дизъюнктивное представление функций алгебры логики.
3. Что называют совершенной дизъюнктивной нормальной формой?
4. Что носит название совершенной конъюнктивной нормальной формы?
5. Получить полиномиальную совершенную нормальную функцию.

## Практическая работа №8. Логика предикатов. Кванторы

**1. Цель занятия:** построить конструктивно алгебру логики высказываний. Ввести операции над логики высказываний и изучить их свойства. Ввести предикаты – отображения произвольных множеств во множество высказываний, которые отражают наличие того или иного признака у предмета. Ввести для предикатов кванторы – одноместные предикаты, означающие локальность или всеобщность высказываний.

### 2. Теоретическое обоснование

*Предикат* (лат. praedicatum – сказанное) – то, что высказывается в суждении об объекте. Предикат отображает наличие того или иного признака у предмета. Другими словами, предикаты — отображения произвольных множеств во множество высказываний

Пусть  $x_1, x_2, \dots, x_n$  – символы переменных произвольной природы. Эти переменные будем называть предметными. Пусть наборы переменных  $(x_1, x_2, \dots, x_n)$  принадлежат множеству  $M = (M_1, M_2, \dots, M_n)$ , которое будем называть предметной областью (т.е.  $x_i \in M_i$ , где  $M_i$  называются областью определения переменной  $x_i$ ). Предикатом местности  $n$  ( $n$ -местным предикатом), определенным на предметной области  $M$ , называют логическую функцию, принимающую либо значение И, либо значение Л.

Множество  $n$ -местных предикатов, определенных на  $M$ , образует булеву алгебру предикатов. Т.о., для них справедливы основные эквивалентности.

Пусть  $P(x)$  — одноместный предикат. Поставим ему в соответствие высказывание, обозначаемое  $\forall x P(x)$  (читается «для любого  $x$   $P(x)$ »), которое истинно тогда и только тогда, когда  $P(x)$  – тождественно истинный предикат. О высказывании  $\forall x P(x)$  говорят, что оно получено из предиката  $P$  навешиванием *квантора всеобщности* по переменной  $x$ .

Пусть  $P(x)$  – одноместный предикат. Поставим ему в соответствие высказывание, обозначаемое  $\exists x P(x)$  (читается «существует  $x$   $P(x)$ »), которое ложно тогда и только тогда, когда  $P(x)$  – тождественно ложный предикат. О высказывании  $\exists x P(x)$  говорят, что оно получено из предиката  $P$  навешиванием *квантора существования* по переменной  $x$ .

Разноименные кванторы, вообще говоря, не коммутируют.

Формулы алгебры логики, в которых из логических символов имеются только символы  $\{\neg, \&, \vee\}$ , причем символ « $\neg$ » отрицания встречается лишь перед символами предикатов, будем называть *приведенными формулами*.

### 3. Задачи для выполнения

#### Предикаты

1. Найти какое-нибудь предложение с одной свободной числовой переменной, которое выполняется для всех значений переменной, кроме 3, 5, 17.

2. Изобразить геометрическое место точек координатной плоскости  $Oxy$ , для которых:

- 1)  $x \geq 0$ ;
- 2)  $y \geq 0$ ;
- 3)  $x \geq 0 \wedge y \geq 0$ ;
- 4)  $x \geq 0 \vee y \geq 0$ ;
- 5)  $x \geq 0 \rightarrow y \geq 0$ ;
- 6)  $y \geq 0 \rightarrow x \geq 0$ ;

7)  $x \geq 0 \leftrightarrow y \geq 0$ .

3. Продолжить в том же духе, добавив к уже рассмотренным следующие предложения:

1)  $x = y$ ;

2)  $\tilde{o} + \tilde{o} = \acute{o} + \acute{o}$ ;

3)  $x + y \leq 0$ ;

4)  $x - y \leq 0$ ;

5)  $\{ \mid \mid \mid \} \{ \geq + \leq 1; 4, 2 \} x x y$

6)  $\{ \mid \mid \mid \} \{ \leq + \geq 1; 4, 2 \} x x y$

7)  $xy \leq 0$ ;

8)  $xy \leq 1$ ;

9)  $xy(y - x^2) \geq 0$ ;

10)  $xy(xy - 1) \geq 0$ ;

### Кванторы

Для каждого из следующих предложений постарайтесь наиболее точно передать выражаемую им мысль с помощью предложения, содержащего кванторы.

1. а)  $x$  есть нуль; б) нуль существует.

2. а) Число  $x$  обладает противоположным; б) каждое число обладает противоположным.

3. а)  $x$  меньше некоторого числа; б) каждое число меньше некоторого числа; в) некоторое число меньше  $x$ ; г) некоторое число меньше каждого числа.

4. а) Число  $x$  – рациональное; б) число  $x$  – иррациональное.

5. а) Точки  $A, B$  и  $C$  лежат на одной прямой; б) точки  $A, B$  и  $C$  не лежат на одной прямой.

6. Прямые  $a$  и  $b$  пересекаются.

7. Прямые  $a$  и  $b$  (в пространстве) скрещиваются.

8. Фигуры  $F$  и  $G$  равны.

9. Плоскости  $\alpha$  и  $\beta$  перпендикулярны.

10. Угол между прямыми  $a$  и  $b$  в пространстве равен  $\varphi$ .

. Геометрическая интерпретация кванторов.

1. Изобразить геометрическое место точек (ГМТ)  $P(p; q)$ , для которых  $(\exists x)(x^2 + px + q = 0)$ ;  $(\forall x)(x^2 + px + q = 0)$ .

2. При каких  $p$   $(\exists q)(\exists x)(x^2 + px + q = 0)$ ?

3. Верно ли, что  $(\exists p)(\exists q)(\exists x)(x^2 + px + q = 0)$ ?  $(\forall p)(\exists q)(\exists x)(x^2 + px + q = 0)$ ?

4. При каких  $p$   $(\forall q)(\exists x)(x^2 + px + q = 0)$ ?

5. Верно ли, что  $(\exists p)(\forall q)(\exists x)(x^2 + px + q = 0)$ ?  $(\forall p)(\forall q)(\exists x)(x^2 + px + q = 0)$ ?

6. При каких  $q$   $(\exists p)(\exists x)(x^2 + px + q = 0)$ ?

7. Верно ли, что  $(\forall q)(\exists p)(\exists x)(x^2 + px + q = 0)$ ?  $(\exists q)(\exists p)(\exists x)(x^2 + px + q = 0)$ ?

8. При каких  $q$   $(\forall p)(\exists x)(x^2 + px + q = 0)$ ?

9. Верно ли, что  $(\forall q)(\forall p)(\exists x)(x^2 + px + q = 0)$ ?  $(\exists q)(\forall p)(\exists x)(x^2 + px + q = 0)$ ?

10. Изобразить ГМТ  $P(p; q)$ , для которых  $(\forall x)(x^2 + px + q \geq 0)$ .

4. Докажите, что

1.  $(\forall x)(x^2 + 1 \geq 2x)$ .

2.  $(\forall x)(x > 1 \rightarrow x^2 > x)$ .

3.  $(\forall x)(0 < x < 1 \rightarrow x^2 < x)$ .

4.  $(\forall x)(x^4 - x^3 + x^2 - x + 1 \geq 0)$ .

5.  $(\forall x)(\forall y)(x^2 + 2xy + 3y^2 + 2x + 6y + 4 > 0)$ .
6.  $(\forall x)(\forall y)(x - y \geq 4 \rightarrow x^2 - 5x \geq y - 5)$ .
7.  $(\exists x)(2x + 3 = 7)$ .
8.  $(\exists x)(-2x > 1)$ .
9.  $(\exists x)(x^2 - 2x > 0)$ .
10.  $(\exists x)(x + x^{-1} = 5,2)$ .
11.  $(\exists a)(\forall x)(x < a \rightarrow x^2 + x - 5 > 0)$ .
12.  $(\exists y)(\forall x)(\| 2\| 2 y - x \geq .$
13.  $(\forall p)(\exists q)(\forall x)(x^2 + px + q \geq 0)$ .
14.  $(\forall x)(\exists y)(\forall z)(z^2 + xz + yz \geq 0)$ .28
155.  $(\forall x)(\exists y)(\forall z)(xy + xz + yz \leq 0)$ .
16.  $(\forall x)(\exists y)(\forall z)(1\,999xy + 2\,000xz + 2\,001yz \leq 0)$ .
17.  $(\forall a)(\exists b)(\forall x)(x < b \rightarrow 2x + 3 < a)$ .
18.  $(\forall a)(\exists b)(\forall z)(z < b \rightarrow z^2 > a)$ .
19.  $(\forall a)(\exists b)(\forall x)(x < b \rightarrow (x - 2a)(x + 3a) > 0)$ .
20.  $(\forall x)(\exists y)(2x + 3y < 5)$ .

#### 4.Контрольные вопросы

1. Какие знаете операции логики высказываний и каковы их свойства.?
2. Что такое предикаты?
3. Что такое кванторы?
4. Какого типа бывают кванторы?
5. Коммутируют ли разноименные кванторы? Покажите это.



## Практическая работа №9. Разработка вычислительных алгоритмов линейной структуры. Разработка алгоритмов разветвляющейся структуры. Разработка алгоритмов циклической структуры

**Цель и содержание работы:** научиться составлять алгоритмы линейной структуры.

### Теоретическое обоснование

Алгоритм линейной структуры – это алгоритм, в котором блоки выполняются последовательно друг за другом. Программа линейной структуры реализует линейный алгоритм. Все программы, выполненные в предыдущей работе, имеют линейный алгоритм. Для организации программы линейной структуры используют операторы присваивания, ввода исходных данных и вывода результатов обработки данных.

Чаще всего линейные алгоритмы используются для программирования вычислений по формулам, которые записываются в виде выражений.

**Выражения** состоят из констант, переменных, операций, функций и круглых скобок, определяющих последовательность выполнения действий. Значения выражений обычно присваиваются переменным.

С каждым видом выражений связано определенное множество операций и встроенных (стандартных) функций языка.

Для числовых значений применяются арифметические операции и математические функции.

К арифметическим операциям относятся: сложение (+), вычитание (-), умножение (\*), деление (/), возведение в степень (^), получение целой части от деления одного числа на другое (операция \), получение остатка от деления (операция mod). Например,  $2^3=8$ ,  $7/3=2.3333333$ ,  $7\backslash 3=2$ ,  $7 \bmod 3=1$ .

Математические стандартные функции представлены в таблице 1.

Таблица 1 – Стандартные математические функции

| Название функции    | Обозначение в математике | Запись в Visual Basic | Примечание                                      |
|---------------------|--------------------------|-----------------------|-------------------------------------------------|
| Модуль              | $ x $                    | ABS(x)                |                                                 |
| Синус               | $\sin x$                 | SIN (X)               | x задан в радианах                              |
| Косинус             | $\cos x$                 | COS (X)               | x задан в радианах                              |
| Тангенс             | $\operatorname{tg} x$    | TAN (X)               | x задан в радианах                              |
| Арктангенс          | $\operatorname{arctg} x$ | ATN (X)               |                                                 |
| Корень квадратный   | $\sqrt{x}$               | SQR (X)               | $x \geq 0$                                      |
| Абсолютная величина | $ x $                    | ABS (X)               | x – числовое выражение                          |
| Экспонента          | $e^x$                    | EXP (X)               | e – основание натурального логарифма, $e = 2.7$ |
| Логарифм            | $\ln x$                  | LOG (X)               | $x > 0$                                         |
| Случайное число     |                          | RND                   | выдает случайное число                          |

Порядок вычисления операций в выражении:

1. Выполняется выражение в скобках.
2. Вычисляются стандартные функции.
3. Операции возведение в степень.
4. Умножение, деление.

# 5. Сложение, вычитание.

Запись всех элементов выражений выполняется в одну строку. Поэтому суммы и разности в числителях и знаменателях дробей, а также произведения в знаменателях необходимо заключать в скобки.

## Практическая часть

В соответствии с вариантом задания 2 составить блок-схему.

| № варианта | Расчетные формулы задания 2                                                                                  | Значения исходных данных                |
|------------|--------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 1          | $W = (x^{y+1} + \frac{bx}{8,2 - 2,1bx^4})(b-1)$ $Y = \sqrt[3]{z \ln^2 z}$                                    | $z = 0,7$<br>$b = 2,2$<br>$x = 1,3$     |
| 2          | $Q = \frac{\sqrt[3]{a+b}}{2(a-b)} \sin^2 x$ $R = \frac{\sqrt[4]{b}}{\sin a + b \cos b} - e^{a-b}$            | $a = -0,3$<br>$b = 12,6$<br>$x = 1,1$   |
| 3          | $b = 1 + \frac{z^2}{3 + z^2/5}$ $W =  x^{y/x} - \sqrt[3]{y/x} $                                              | $x = 1,426$<br>$y = 18,25$<br>$z = 3,5$ |
| 4          | $y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }$<br>$s = b \sin(at^2 \cos 2t) - 1$                                   | $a = -0,5$<br>$b = 1,7$<br>$t = 0,44$   |
| 5          | $\omega = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x$<br>$y = \cos^2 x^3 - x/\sqrt{a^2 + b^2}$                       | $a = 1,5$<br>$b = 15,5$<br>$x = -2,9$   |
| 6          | $s = x^3 tg^2(x+b)^2 + a/\sqrt{x+b}$ $q = \frac{bx^2 - a}{e^{ax} - 1}$                                       | $a = 16,5$<br>$b = 3,4$<br>$x = 0,61$   |
| 7          | $R = x^2(x+1)/b - \sin^2(x+a)$<br>$S = \sqrt{xb/a} + \cos^2(x+b)^3$                                          | $a = 0,7$<br>$b = 0,35$<br>$x = 0,5$    |
| 8          | $Y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$ $Z = \frac{x^2}{a} + \cos(x+b)^3$                                       | $a = 1,1$<br>$b = 0,4$<br>$x = 0,2$     |
| 9          | $F = \sqrt[3]{mtgt +  c \sin t }$ $Z = m \cos(bt \sin t) + c$                                                | $m = 2; t = 1,2$<br>$c = -1; b = 0,7$   |
| 10         | $Y = btg^2 x - \frac{a}{\sin^2(x/a)}$ $D = ae^{-\sqrt{a}} \cos(bx/a)$                                        | $a = 3,2$<br>$b = 17,5$<br>$x = -4,8$   |
| 11         | $f = \ln(a+x^2) + \sin^2(x/b)$ $Z = e^{-x} \frac{x + \sqrt{x+a}}{x - \sqrt{ x-b }}$                          | $a = 10,2$<br>$b = 9,2$<br>$x = 2,2$    |
| 12         | $Y = \frac{a^{2x} + b^{-x} \cos(a+b)x}{x+1}$ $R = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x$                        | $a = 0,3$<br>$b = 0,9$<br>$x = 0,61$    |
| 13         | $Z = \sqrt{ax \sin 2x + e^{-2x}}(x+b)$ $W = \cos^2 x^3 - x/\sqrt{a^2 + b^2}$                                 | $a = 0,5; b = 3,1$<br>$x = 1,4$         |
| 14         | $F = 0,23x \sqrt{\frac{1 +  \cos^2 x^3 }{ab} + \frac{a^2 + b^2}{a-b}}$ $Y = \frac{x + tg(x+1)}{\sin^2(x+1)}$ | $a = 12,7;$<br>$x = 1,1$<br>$b = 0,05$  |

|    |                                                                                                                              |                                                  |
|----|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 15 | $G = 0,8e^{2x} - \frac{3a - 2tg^2 x}{\sqrt{1,2 + 3by^2}} \quad Z = \frac{\sin^2 x}{x - 4} + \frac{\cos x^2}{((x - 3)x - 5)}$ | $a = -0,03; x = 1,1$<br>$b = 12,6;$<br>$y = 0,6$ |
| 16 | $G = 1,2a^2 \arctg^3 3x + \sqrt{\frac{ 1 - 3e^{2x} }{abx}} \quad Y = e^{\left  \frac{2(\sin 4x + x)}{3x} \right }$           | $a = 0,07; x = 0,23$<br>$b = 2,02$               |

### Содержание отчёта и его форма

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### Контрольные вопросы и защита работы

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Дайте определение понятию алгоритма.
2. Опишите основные свойства алгоритма.
3. Какие вы знаете способы представления алгоритма.
4. Что такое блок-схема?

### Разработка алгоритмов разветвляющейся структуры

**Цель и содержание работы:** научиться разрабатывать алгоритмы разветвляющейся структуры, получить практические навыки использования основных алгоритмических конструкций.

### Теоретическое обоснование

В зависимости от введенных данных или полученных в процессе работы результатов поток управления может разветвиться на две или более ветвей. Для изображения разветвления в схемах алгоритмов и программ применяется символ «решение». При разветвлении на две ветви используется условный оператор, при разветвлении на большее число ветвей – переключатель.

Применение символа «решение» в условном операторе (разветвление на две ветви) изображено на рисунок 1.

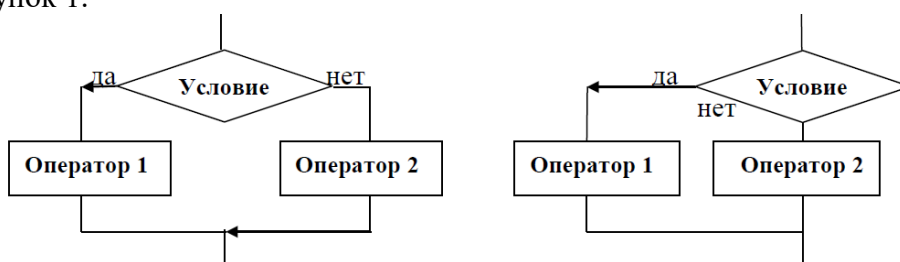


Рисунок 1 – Применение символа «решение»

При выполнении условия, помещенного в символе «решение», поток управления направлен по ветви, при которой стоит слово «да» или знак плюс, при невыполнении условия - по ветви, при которой стоит слово «нет» или знак минус. Линии управления, направленные влево или вверх, обязательно снабжаются стрелками, для направлений вправо или вниз стрелки не обязательны.

Показаны две возможности изображения условного оператора. Если оператор 2 – не пустой оператор, в условном операторе присутствует ветвь else (иначе), имеем полную форму условного оператора. Если же оператор 2 – пустой оператор, то имеем сокращенную форму условного оператора.

Приведем пример алгоритма разветвляющейся структуры с применением условного оператора (рисунок 2) – решение квадратного уравнения  $ax^2+bx+c=0$ .

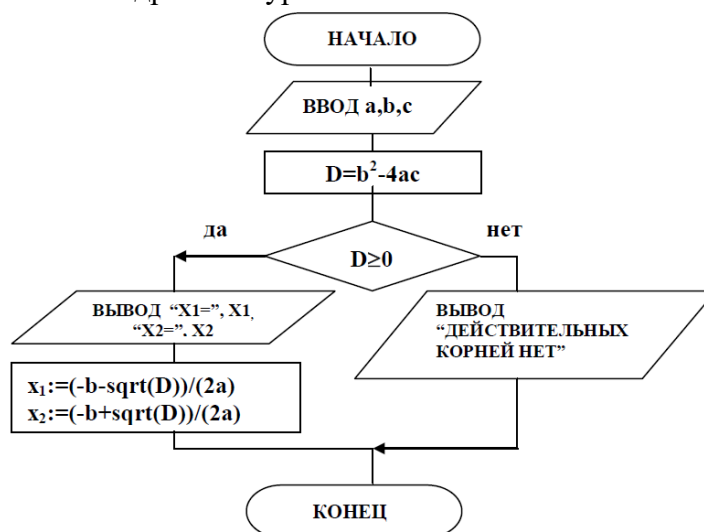


Рисунок 2–Разветвляющийся алгоритм с применением условного оператора

Если дискриминант  $b^2-4ac$  неотрицателен, уравнение имеет действительные корни, в противном случае действительных корней нет.

Иногда в алгоритмах используется безусловная передача управления (оператор безусловного перехода) (рисунок 3).

Оператор безусловного перехода изображается линией управления, прерывающей естественный порядок выполнения операторов сверху вниз, и ведущей к произвольному элементу, который может быть расположен как выше, так и ниже элемента, из которого исходит безусловная передача управления.

В приведенном примере происходит безусловная передача управления от процесса 1 на проверку условия. Переход осуществляется вверх, но возможно и обратное направление. В теории программирования доказано, что можно обойтись без операторов безусловного перехода, однако иногда они оказываются удобным средством и оставлены во многих языках программирования, например, Бейсике, Паскале, Си, Си++.

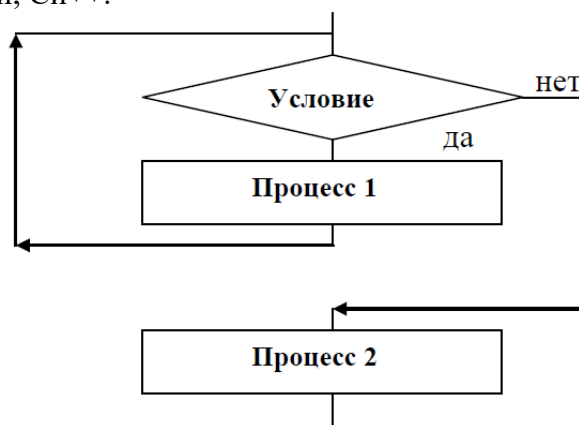


Рисунок 3 – Алгоритм с безусловной передачей управления

### Практическая часть

1. Ознакомиться с разветвляющимися алгоритмами примерах.
2. Разработать алгоритм вычисления значения функции  $y$ , определяемой разными аналитическими выражениями и составить его схему согласно варианту. Исходные данные вводятся, результат выводится. Предусмотреть вывод приглашения к вводу данных.

### Варианты

|   |                                                                                                                                      |   |                                                                                                                                      |    |                                                                                                                                      |
|---|--------------------------------------------------------------------------------------------------------------------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------------------------------------------------------|
| 1 | $y = \begin{cases} 3x^2 - 1 & \text{при } x \leq 1 \\ 2x & \text{при } 1 < x \leq 4 \\ 3x - 4 & \text{при } x > 4 \end{cases}$       | 5 | $y = \begin{cases} 3x^2 - 1 & \text{при } x \leq 1 \\ 2x & \text{при } 1 < x \leq 4 \\ 3x - 4 & \text{при } x > 4 \end{cases}$       | 9  | $y = \begin{cases} 3x^2 - 1 & \text{при } x \leq 1 \\ 2x & \text{при } 1 < x \leq 4 \\ 3x - 4 & \text{при } x > 4 \end{cases}$       |
| 2 | $y = \begin{cases} 3x^3 - 2 & \text{при } x \leq 0 \\ 3x - 2 & \text{при } 0 < x \leq 5 \\ 4x - 7 & \text{при } x > 5 \end{cases}$   | 6 | $y = \begin{cases} 3x^3 - 2 & \text{при } x \leq 0 \\ 3x - 2 & \text{при } 0 < x \leq 5 \\ 4x - 7 & \text{при } x > 5 \end{cases}$   | 10 | $y = \begin{cases} 3x^3 - 2 & \text{при } x \leq 0 \\ 3x - 2 & \text{при } 0 < x \leq 5 \\ 4x - 7 & \text{при } x > 5 \end{cases}$   |
| 3 | $y = \begin{cases} 3x^2 - 3 & \text{при } x \leq -1 \\ 4x + 4 & \text{при } -1 < x \leq 3 \\ 3x + 5 & \text{при } x > 3 \end{cases}$ | 7 | $y = \begin{cases} 3x^2 - 3 & \text{при } x \leq -1 \\ 4x + 4 & \text{при } -1 < x \leq 3 \\ 3x + 5 & \text{при } x > 3 \end{cases}$ | 11 | $y = \begin{cases} 3x^2 - 3 & \text{при } x \leq -1 \\ 4x + 4 & \text{при } -1 < x \leq 3 \\ 3x + 5 & \text{при } x > 3 \end{cases}$ |
| 4 | $y = \begin{cases} 2x^2 - 1 & \text{при } x \leq 2 \\ 8x + 5 & \text{при } 2 < x \leq 3 \\ 7x^3 - 4 & \text{при } x > 3 \end{cases}$ | 8 | $y = \begin{cases} 2x^2 - 1 & \text{при } x \leq 2 \\ 8x + 5 & \text{при } 2 < x \leq 3 \\ 7x^3 - 4 & \text{при } x > 3 \end{cases}$ | 12 | $y = \begin{cases} 2x^2 - 1 & \text{при } x \leq 2 \\ 8x + 5 & \text{при } 2 < x \leq 3 \\ 7x^3 - 4 & \text{при } x > 3 \end{cases}$ |

### Содержание отчёта и его форма

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### Контрольные вопросы и защита работы

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Что такое Блок-схема?
2. Какие основные символы блок-схем Вы знаете?
3. Как составить алгоритм разветвляющейся структуры?
4. Назовите виды алгоритмов разветвляющейся структур

### Разработка алгоритмов циклической структуры

**Цель и содержание работы:** приобрести навыки составления алгоритмов циклической структуры

#### Теоретическая часть

**Базовая структура «цикл»** обеспечивает многократное выполнение некоторой совокупности действий. Повторяющаяся совокупность действий называется – **телом цикла**.

Величина, с которой связано многократное выполнение тела цикла называется – **параметром цикла**. Параметр цикла имеет **начальное** и **конечное** значения.

**Шаг цикла** – величина, на которую изменяется значение параметра цикла при каждом выполнении цикла.

Существует три вида циклов:

- цикл с заранее известным числом повторений (цикл с параметром);
- цикл с предусловием (цикл «пока»);
- цикл с постусловием (цикл «до»).

Структура цикла с заранее заданным числом повторений изображена на рисунке 1.

▲ Параметру цикла  $p$  присваивается начальное значение  $n$  и происходит выполнение тела цикла. Далее значение параметра цикла увеличивается на величину шага  $h$  (шаг, равный единице в схеме можно не указывать) и проверяется условие: (текущее значение параметра цикла должно быть меньше конечного  $k$  значения или равно ему  $p \leq k$ ). Цикл будет повторяться до тех пор, пока это условие истинно. Как только  $p$  станет больше  $k$  ( $p > k$ ) произойдет выход из цикла,

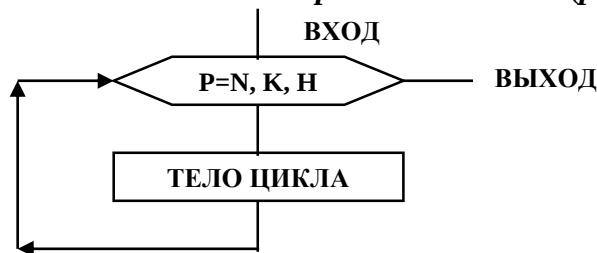


Рисунок 1 – Структура цикла с параметром.

Такая циклическая структура часто используется для вычисления сумм, произведений и т.п.

Кроме циклов с заранее известным числом повторений существуют циклы, число повторений которых заранее неизвестно, а условия выхода из циклов определяются при их выполнении.

В таких циклах проверка условия продолжения цикла может проводиться до выполнения действий цикла или после их выполнения. В циклах с предусловием и с постусловием перед циклом, как правило, выполняется подготовительный процесс (задаются начальное  $n$  и конечное  $k$  значения параметра цикла  $p$  и величина шага  $h$ ), а в теле цикла значение параметра цикла увеличивается на величину шага  $h$ .

Структура **цикла с предусловием** показана на рисунке 2.

Из схемы видно, что после процесса подготовки проверяется условие выполнения цикла: если условие истинно, то выполняется тело цикла. Затем, управление вновь передается на проверку условия.

Выход из цикла происходит в случае невыполнения условия. Если условие ложно изначально, тело цикла не выполнится ни разу.

Из схемы видно, что после процесса подготовки проверяется условие выполнения цикла: если условие истинно, то выполняется тело цикла. Затем управление вновь передается на проверку условия.

Выход из цикла происходит в случае невыполнения условия. Если условие ложно изначально, тело цикла не выполнится ни разу.

Структура **цикла с постусловием** изображена на рисунке 3. Из рисунка видно, что после процесса подготовки выполняется тело цикла, затем проверяется условие выполнения цикла. Если условие истинно, то управление вновь передается на тело цикла.

Выход из цикла происходит в случае невыполнения условия. Если условие ложно изначально, тело цикла обязательно выполнится хотя бы раз.

**Пример 1.** Составить блок-схему задачи. С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности.

Параметром цикла в данной задаче является  $i$  (порядковый номер числа  $X$  в последовательности). Он изменяется с постоянным шагом 1 от начального значения параметра цикла 1 до конечного значения  $N$ .

Схема алгоритма с использованием **цикла с параметром** показана на рисунке 4. Как видно из схемы алгоритма, перед циклом вводится значение  $N$ , а переменной  $S$  присваивается значение 0. Выход из цикла происходит при  $i > N$ .

**Пример 2.** С клавиатуры вводится последовательность из  $N$  чисел. Определить минимальный элемент этой последовательности.

**Пример 3.** Составить блок-схему задачи. С клавиатуры вводится последовательность из  $N$  чисел. Определить сумму положительных элементов этой последовательности.

Параметром цикла в данной задаче является  $i$  (порядковый номер числа  $X$  в последовательности). Он изменяется с постоянным шагом 1 от начального значения параметра цикла 1 до конечного значения  $N$ .

Схема алгоритма с использованием цикла с предусловием показана на рисунке 6.

Решим эту задачу разными способами: используя цикл с предусловием (рисунк 7) и цикл с постусловием (рисунк 8).

Этап подготовки в данной схеме включает в себя: ввод конечного значения параметра цикла  $N$ , задание начального значения  $i$ , обнуление суммы  $S$ .

Цикл начинается с проверки условия выполнения цикла. В данном случае цикл должен выполняться пока значение параметра  $i \leq N$ . В теле цикла вычисляется значение суммы, а далее производится изменение параметра цикла на величину шага равную 1.

Как только условие станет ложным, производится выход из цикла и вывод результата.

Данная схема похожа на схему с использованием предусловия, за исключением месторасположения условия выполнения цикла. Условие  $i \leq N$  проверяется после выполнения тела цикла.

### Практическая часть

| № варианта | Условия задания                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1          | Вводится последовательность из $N$ чисел. Найти произведение и количество положительных среди них чисел.                                           |
| 2          | Вводится последовательность из $N$ чисел. Найти, сколько в ней нулей и их порядковые номера.                                                       |
| 3          | Вводится последовательность из $M$ чисел. Найти наименьшее из всех положительных чисел.                                                            |
| 4          | Вводится последовательность целых чисел, 0 – конец последовательности. Подсчитать, сколько в ней отрицательных, и сколько положительных элементов. |
| 5          | Вводится последовательность целых чисел, 0 – конец последовательности. Найти сумму и количество положительных чисел.                               |
| 6          | Дана последовательность из $M$ чисел. Подсчитать, сколько в ней отрицательных, и сколько положительных (включая нулевые) элементов.                |
| 7          | Дана последовательность из $M$ чисел. Вычислить сумму и количество чисел, которые меньше 5.                                                        |
| 8          | Дана последовательность из $M$ чисел. Вычислить произведение и количество чисел, которые меньше 10.                                                |
| 9          | Вводится последовательность ненулевых чисел, 0 – конец последовательности. Сколько из этих чисел больше 5, но меньше 10. Найти сумму таких чисел.  |
| 10         | Дана последовательность из $M$ чисел. Все отрицательные среди них числа возвести в квадрат и найти их сумму.                                       |
| 11         | Дана последовательность из $M$ чисел. Найти сумму и произведение всех положительных чисел.                                                         |
| 12         | Дана последовательность из $M$ чисел. Найти наибольшее среди отрицательных из них.                                                                 |
| 13         | Дана последовательность из $M$ чисел. Найти наименьшее из всех чисел, больших 10, и его порядковый номер.                                          |
| 14         | Дана последовательность из $M$ чисел, все положительные среди них числа заменить их корнем и найти их сумму.                                       |
| 15         | Дана последовательность чисел, заканчивающаяся 0. Найти сумму чисел меньших 3 и произведение чисел больших 10.                                     |

|    |                                                                                                                     |
|----|---------------------------------------------------------------------------------------------------------------------|
| 16 | Дана последовательность из $M$ чисел. Сколько из этих чисел больше 7, но меньше 20. Найти произведение таких чисел. |
|----|---------------------------------------------------------------------------------------------------------------------|

### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата

A4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Какой алгоритм называется алгоритмом циклической структуры?
2. Какие основные символы блок-схем для организации цикла вы знаете?
3. Какие виды циклов существуют?
4. Что называется телом цикла?
5. Какая величина называется параметром цикла?
4. Нарисуйте блок-схему алгоритма цикла с параметром.
5. Нарисуйте блок-схему алгоритма цикла с предусловием.
6. Нарисуйте блок-схему алгоритма цикла с постусловием.



## Практическая работа №10. Одномерные и многомерные массивы. Сортировка массива методом выбора

**Цель и содержание работы:** приобретение практических навыков в составлении программ с одномерными массивами.

### Теоретическое обоснование

Массивы – структурированный тип данных с элементами одного типа, имеющий одно имя и определенное количество элементов. Количество элементов определяет размер массива. Например, массив составляют номера зачетных книжек студентов, здесь число элементов равно числу студентов; массив образуют набор чисел, их количество равно числу элементов массива. Порядковый номер элемента массива называется его индексом. Массив может иметь не один, а большее число индексов. Число индексов называется размерностью массива, например, массив с двумя индексами называется двумерным массивом. Строка символов является массивом символов, вектор – массив вещественных чисел, матрица – массив векторов. Обработка массивов выполняется следующим образом: описание, ввод или инициализация элементов массива, преобразование и вывод.

При описании массивов после имени в квадратных скобках указывается число его элементов. В языке Си индекс элемента массива начинается с нуля, поэтому индекс последнего элемента массива на 1 меньше числа элементов в данном массиве.

Массив, имеющий больше одного индекса, называется многомерным. Если индексов два, то это двумерный массив или матрица. Многомерные массивы задаются указанием каждого измерения в квадратных скобках.

Например, команда `matr [6] [8];` задает описание матрицы из 6 строк и 8 столбцов. Нумерация строк и столбцов начинается с 0. По каждому из индексов устанавливается размер массива. Такой массив имеет  $6 \cdot 8 = 48$  элементов, первый индекс изменяется от 0 по 5 (включая), второй индекс – от 0 по 7. В памяти такой массив располагается в последовательных ячейках построчно, при переходе к следующему элементу быстрее всего изменяется последний индекс. При обработке многомерных массивов используются вложенные циклы, т.е. цикл по столбцам располагается внутри цикла по строкам.

Такими способами задается двумерный массив `mas` целых чисел из трех строк по два элемента в каждом. Причем индексы элементов массива начинаются с нуля, т.е. `mas00=1`, `mas01=2`, `mas10=3`, `mas11=4`, `mas20=5`, `mas21=6`.

Если при инициализации задано меньше значений, чем объявлено в массиве, то остальные элементы задаются нулями. Если при инициализации указывается больше элементов, чем объявлено, то при компиляции появится сообщение об ошибке.

### Практическая часть

#### Варианты заданий

##### Задание 1. Составить блок-схемы алгоритмов следующих задач.

1. Ввести массив `A` из 15 элементов, вывести его до преобразования и после.
2. Найти сумму элементов, больших 5 и меньших 10. Разделить каждый элемент массива на эту сумму.
3. Найти наибольший элемент массива `A` и сравнить его с наименьшим элементом.
3. Найти наименьший элемент и переставить его с последним элементом.
4. Найти произведение положительных элементов массива `A` и сравнить его с произведением отрицательных.
5. Найти произведение отрицательных элементов и сумму положительных.
6. Найти сумму положительных элементов массива `A` и количество равных 0.
7. На сколько каждый элемент массива `A` отличается от среднеарифметического значения элементов этого массива.
8. Все элементы массива `A`, большие 7 и меньшие 20, переписать в новый массив.
9. Переписать все элементы массива `A` наоборот, и сохранить их в этом же массиве.
10. Получить новый массив, элементы которого равны элементам массива `A`, деленным на наибольший элемент массива `A`.

11. Из элементов массива А, получить массив положительных и массив отрицательных элементов.

12. Проверить, является ли массив А возрастающей последовательностью.

**Задание 2. Составить блок-схемы алгоритмов следующих задач.**

1. Даны матрица А размером  $m \times n$  и вектор В размером  $n$ . Записать на главную диагональ элементы вектора, а в вектор – элементы главной диагонали.

2. Выбрать максимальный элемент матрицы С (размер  $m \times n$ ), элементы четных строк разделить на максимальный элемент, а к элементам нечетных прибавить максимальный элемент.

3. Найти минимальный элемент матрицы С (размер  $m \times n$ ), и поменять его местами с первым элементом.

4. Дана матрица D размером  $m \times n$ . Вычислить суммы элементов каждого столбца. Определить наибольшее значение этих сумм и номер соответствующего столбца.

5. В матрице К размером  $m \times n$  найти в каждом столбце произведение отрицательных элементов и количество нулевых элементов в матрице.

6. Даны две матрицы А и В одинаковой размерности  $m \times n$ . Получить матрицу

$C = \max(a_{ij}, b_{ij})$ , и матрицу  $D = \min(a_{ij}, b_{ij})$ .

7. Дана матрица Р размером  $m \times n$ . Найти сумму минимальных элементов каждого столбца матрицы.

**Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

**Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Что такое массив? индекс элемента массива?
2. Как ввести и вывести элементы вектора в строку и в столбец?
3. Что такое многомерный массив?
4. Как ввести и вывести элементы матрицы в строку и в столбец?
5. Привести пример инициализации матрицы.
6. Как выполняется форматный вывод элементов матрицы?

**Сортировка массива методом выбора**

**Цель и содержание работы:** изучить алгоритмы сортировки массивов и научиться использовать их при обработке данных.

**Теоретическое обоснование**

Выполнение практической работы направлено на освоение основных приемов использования массивов, методов доступа к элементам массивов, их реорганизации и модификации. В качестве практической проблемы, требующей решения, рассматривается известная задача сортировки (упорядочивания) массива в порядке возрастания его элементов. При решении этой задачи требуется исходный массив, содержащий произвольные целые числа, преобразовать к виду, когда каждый элемент массива находится перед другим элементом этого массива, если его значение меньше (больше), чем значение сравниваемого элемента.

В данной практической работе необходимо изучить ряд известных алгоритмов сортировки и создать комплекс программ, реализующий:

- метод сортировки выбором;
- метод сортировки пузырьком;

- метод сортировки включением.

### Метод сортировки выбором

Исходный массив длиной  $N$  разбивается на две части: итог и остаток. Участок массива, называемый **итогом**, располагается с начала массива и должен быть упорядоченным, а участок массива, называемый **остатком**, располагается вплотную за итогом и содержит исходные числа не отсортированной части исходного массива.

Текстуальный алгоритм сортировки выбором:

Шаг 1. Полагается  $i=0$ , т.е. считается, что итоговый участок - пуст.

Шаг 2. В остатке массива ищется минимальный элемент, и он меняется местом с первым элементом остатка ( $i$ -ым элементом массива). После чего значение  $i$  увеличивается на единицу, тем самым расширяя итоговый участок массива (отсортированную часть исходного массива).

Шаг 3. Если  $i < N$ , то повторяется Шаг 2. В противном случае - конец алгоритма, т.к. итог становится равным всему массиву.

Конец алгоритма.

Схема алгоритма методом сортировки выбора представлена на рис. 1.

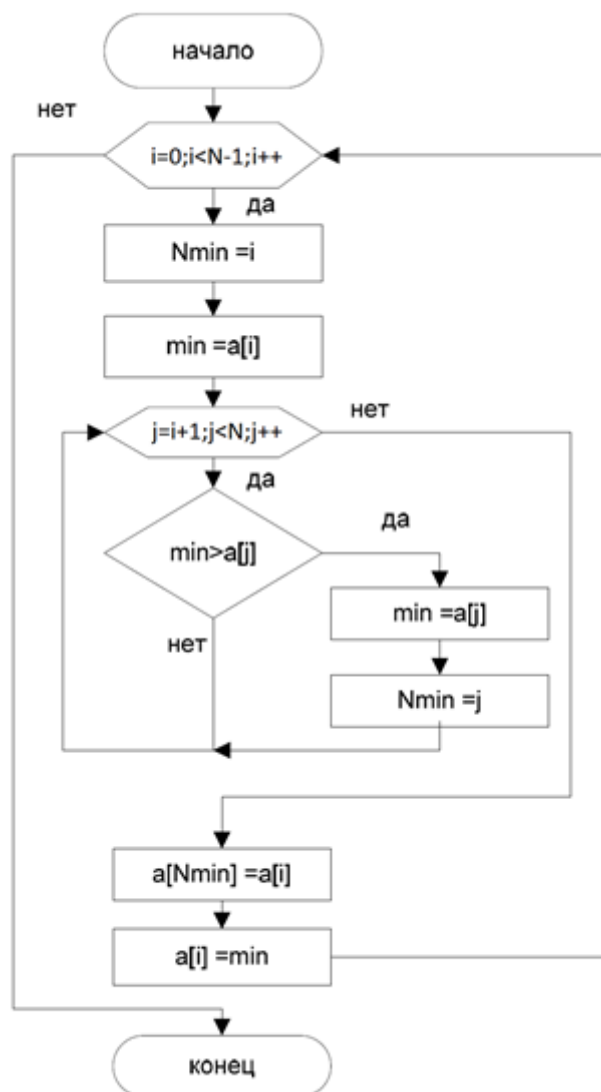


Рисунок 1. Схема алгоритма сортировки методом выбора

### Практическая часть

Используя один из изученных алгоритмов сортировки, составить блок-схему.

1. Выполнить сортировку только четных элементов массива (нечетные элементы остаются на своих местах).
2. Выполнить сортировку элементов, записанных на нечетных местах

### Содержание отчёта и его форма

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

#### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Опишите принцип сортировки массива методом выбора
2. Продемонстрируйте на рисунке любого простого примера сортировки массива методом выбора.
3. Нарисуйте блок-схему алгоритма сортировки массива методом выбора

## Практическая работа №11. Сортировка методом вставки. Сортировка методом пузырька.

**Цель и содержание работы:** сформировать знания сортировки элементов массива методом вставки и умения составлять алгоритм сортировки массива методом вставки.

### Теоретическое обоснование

Происходит такое же разбиение массива на отсортированную и не отсортированную части, но перемещение первого элемента остатка на принадлежащее ему место в итоге делается не сравнением двух соседних элементов, а с помощью метода двоичного поиска, который удобно оформить в виде отдельной процедуры.

Текстуальный алгоритм методом включением:

Шаг 1. Пусть  $i=1$ , т.е. итоговый участок состоит из одного элемента.

Шаг 2. Берется первый элемент остатка и перемещается в отсортированную часть массива так, чтобы итоговый участок остался упорядоченным.

Шаг 3. После того, как первый элемент остатка переместился в итоговый участок, увеличивается на единицу значение переменной  $i$ , тем самым увеличивая отсортированную часть массива. Если  $i < N$ , то управление передается на Шаг 2, в противном случае - работа алгоритма завершена.

Конец алгоритма.

Схема алгоритма методом сортировки включением представлена на рис. 3.

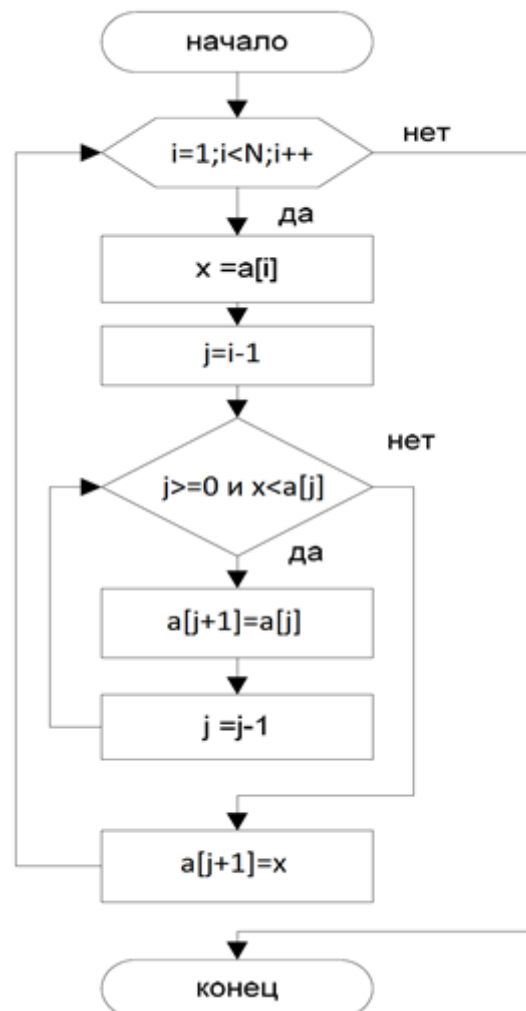


Рисунок 3. Блок-схема алгоритма метода вставки

### Практическая часть

Используя один из изученных алгоритмов сортировки, составить блок-схему.

1. Выполнить сортировку только четных элементов массива (нечетные элементы остаются на своих местах).

2. Выполнить сортировку элементов, записанных на нечетных местах

### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата

A4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Опишите принцип сортировки массива методом вставки
2. Продемонстрируйте на рисунке любого простого примера сортировки массива методом вставки.
3. Нарисуйте блок-схему алгоритма сортировки массива методом вставки

### **Сортировка методом пузырька**

**Цель и содержание работы:** сформировать знания сортировки элементов массива методом пузырька и умения составлять алгоритм сортировки массива методом пузырька

#### **Теоретическое обоснование**

#### **Метод сортировки пузырька**

Аналогично, как и в методе выбора, исходный массив длиной  $N$  разбивается на две части: отсортированную (итог) и не отсортированную (остаток). Первый элемент остатка является  $i$ -ым элементом массива.

Текстуальный алгоритм сортировки пузырьком:

Шаг 1. Пусть  $k=N-1$ , т.е. итоговый участок состоит из одного элемента.

Шаг 2. Берется первый элемент остатка и перемещается на место в итоговый участок массива так, чтобы итог остался упорядоченным. Первый элемент остатка назовем перемещаемым. Перемещение выполняется путем сравнения перемещаемого элемента с последующим элементом. Если последующий элемент больше сравниваемого элемента, то процесс перемещения этого элемента закончен.

Шаг 3. После того, как первый элемент остатка переместился в итоговый участок, уменьшается на единицу значение переменной  $k$ , тем самым увеличивая отсортированную часть массива. Если  $k>0$ , то управление передается на Шаг 2, в противном случае - работа алгоритма завершена.

Конец алгоритма.

Схема алгоритма методом сортировки пузырька представлена на рис. 2.

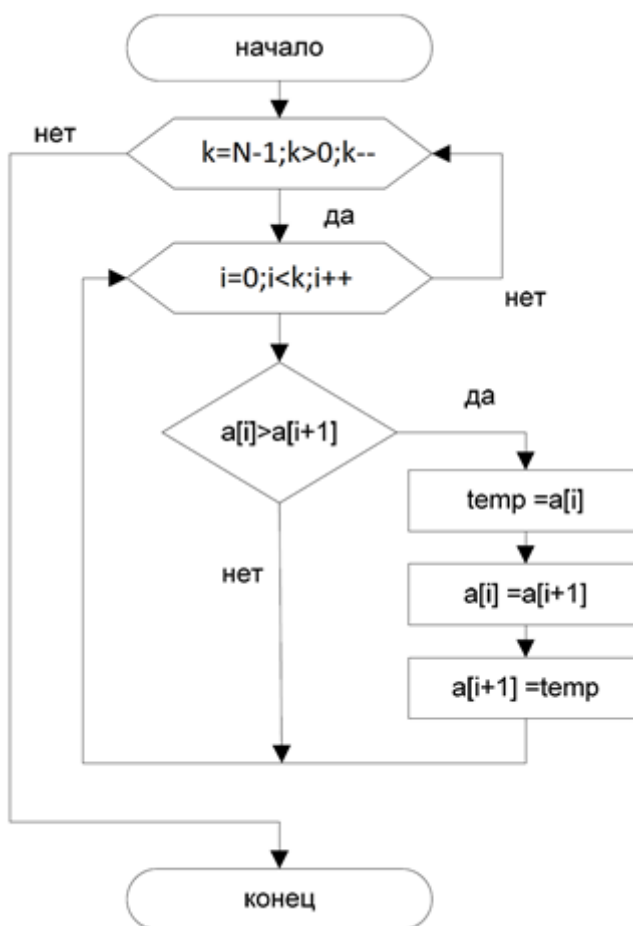


Рисунок 2. Блок-схема алгоритма сортировки методом пузырька

### Практическая часть

Используя один из изученных алгоритмов сортировки, составить блок-схему.

1. Выполнить сортировку только четных элементов массива (нечетные элементы остаются на своих местах).
2. Выполнить сортировку элементов, записанных на нечетных местах

### Содержание отчёта и его форма

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата

A4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### Контрольные вопросы и защита работы

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Опишите принцип сортировки массива методом пузырька
2. Продемонстрируйте на рисунке любого простого примера сортировки массива методом пузырька.
3. Нарисуйте блок-схему алгоритма сортировки массива методом пузырька

## Практическая работа №12. Рекурсивные функции.

**Цель работы:** получение навыков в написании программ с использованием рекурсивных функций.

### Теоретическое обоснование

Рекурсивной (самовызываемой или самовызывающей) называют функцию, которая прямо или косвенно вызывает сама себя.

При каждом обращении к рекурсивной функции создается новый набор объектов автоматической памяти, локализованных в коде функции.

Возможность прямого или косвенного вызова позволяет различать прямую или косвенную рекурсии. Функция называется косвенно рекурсивной в том случае, если она содержит обращение к другой функции, содержащей прямой или косвенный вызов первой функции. В этом случае по тексту определения функции ее рекурсивность (косвенная) может быть не видна. Если в функции используется вызов этой же функции, то имеет место прямая рекурсия, т.е. функция по определению рекурсивная.

Рекурсивные алгоритмы эффективны в задачах, где рекурсия использована в самом определении обрабатываемых данных. Поэтому изучение рекурсивных методов нужно проводить, вводя динамические структуры данных с рекурсивной структурой. Рассмотрим вначале только принципиальные возможности, которые предоставляет язык Си для организации рекурсивных алгоритмов.

В рекурсивных функциях необходимо выполнять следующие правила.

1. При каждом вызове в функцию передавать модифицированные данные.
2. На каком-то шаге должен быть прекращен дальнейший вызов этой функции, это значит, что рекурсивный процесс должен шаг за шагом упрощать задачу так, чтобы для нее появилось нерекурсивное решение, иначе функция будет вызывать себя бесконечно.
3. После завершения очередного обращения к рекурсивной функции в вызывающую функцию должен возвращаться некоторый результат для дальнейшего его использования.

### Практическая часть

Написать программы решения следующих задач, используя рекурсивную функцию.

1. Найти наименьшую цифру в десятичной записи заданного натурального числа.
2. Подсчитать количество цифр в заданном натуральном числе.
3. Вычислить наибольший общий делитель двух натуральных чисел.
4. Найти число, которое образуется из заданного натурального числа при записи его цифр в обратном порядке. Например, для числа 1234 получаем результат 4321.
5. Вычислить сумму:  $1! + 2! + 3! + \dots + n!$  ( $n \leq 15$ ).
6. Вычислить сумму:  $2! + 4! + 6! + \dots + n!$  ( $n \leq 16$ ,  $n$  – четное).
7. Логическая функция возвращает 1, если ее аргумент – простое число.
8. Вычислить функцию Аккермана для всех неотрицательных целых аргументов  $m$  и  $n$ :

$$\begin{aligned} & \begin{cases} A(0, n) = n + 1, \\ A(m, n) = \begin{cases} A(m, 0) = A(m - 1, 1), & m > 0, \\ A(m, n) = A(m - 1, A(m, n - 1)), & m, n > 0 \end{cases} \end{cases} \end{aligned}$$

9. Найти количество нечетных цифр в десятичной записи заданного натурального числа.
10. Найти количество цифр, кратных 3, в десятичной записи заданного натурального числа.

$$C_n^k = \frac{n!}{k!(n - k)!} \quad (\text{значение } 0! = 1).$$

11. Вычислить значение
12. Вычислить произведение четного количества  $n$  ( $n \in \mathbb{Z}$ ) сомножителей следующего вида:



$$y = \left( \frac{2}{1} - \frac{2}{3} \right) \cdot \left( \frac{4}{3} - \frac{4}{5} \right) \cdot \left( \frac{6}{5} - \frac{6}{7} \right) \cdot \dots$$

13. Вычислить  $y = x^n$  по следующему правилу:  $y = (x^{n/2})^2$ , если  $n$  четное и  $y = x \oplus y^{n-1}$ , если  $n$  нечетное.

14. Вычислить значение  $y(n) = \sqrt{1 + \sqrt{2 + \dots + \sqrt{n}}}$ .

15. Вычислить значение  $x = \sqrt{a}$ , используя рекуррентную формулу  $x_n = \frac{1}{2} \left( x_{n-1} + \frac{a}{x_{n-1}} \right)$ , в качестве начального значения использовать  $x_0 = 0,5 \oplus (1 + a)$ .

#### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

#### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Что такое нуль-функция, функция следования, функция проекции?
2. Что такое суперпозиция функций.
3. Опишите схема примитивной рекурсии.
4. Опишите операцию минимизации.
5. Построение дерева вызовов рекурсии

## **Практическая работа №13. Машина Тьюринга.**

### **Перемещение автомата, замена символов, анализ символов.**

**Цель и содержание работы:** развить у студентов математическую интуицию, повысить уровень их математической культуры путем закрепления навыков работы на Машине Тьюринга.

#### **Теоретическое обоснование**

Формальные определения алгоритма появились в тридцатых - сороковых годах нашего века. Одним из первых было определение английского математика А. Тьюринга, который в 1936 году описал схему некоторой гипотетической (абстрактной) машины и предложил называть алгоритмами то, что умеет делать такая машина. При этом определении, если что-то не может быть сделано машиной Тьюринга, это уже не алгоритм. Иначе говоря, Тьюринг формализовал правила выполнения действий при помощи описания работы некоторой конструкции.

Вычислительные машины - это тоже конструкции для выполнения алгоритмов, но это реальные устройства, тогда как машина Тьюринга является абстракцией, которая никогда не была реализована. (Как мы увидим дальше, ее и нельзя реализовать.) Поэтому алгоритмы для машины Тьюринга должны выполняться другими средствами. Например, за нее должен, работать человек, действуя так, как действовала бы машина Тьюринга, если бы она существовала. Польза от машины Тьюринга в том, что, говоря об этой воображаемой конструкции, мы можем доказывать существование или несуществование алгоритмов решения различных задач.

Исходя из этого, Тьюринг искал как можно более простую, «бедную» алгоритмическую схему, лишь бы она была универсальной. Важно, чтобы было удобно доказывать, что она не может решить задачу, которая, как нам кажется, вообще неразрешима. Когда же речь идет о вычислительной машине, то, наоборот, мы больше всего ценим ее удобство, богатство ее возможностей; требуем, чтобы человеку было легко с ней работать. В этом нас сдерживает только цена желаемых удобств и трудность разработки оборудования.

Принципиальное отличие машины Тьюринга от вычислительных машин состоит в том, что ее запоминающее устройство представляет собой бесконечную ленту, тогда как у вычислительной машины может быть очень большое, но уж во всяком случае, не бесконечное запоминающее устройство. Машину Тьюринга нельзя физически реализовать именно из-за ее бесконечной ленты. В этом смысле она мощнее любой вычислительной машины.

*Лента* машины Тьюринга разделена на *ячейки*. В каждой ячейке может находиться одна из букв какого-нибудь алфавита. Алфавиты могут быть разные, но для каждой конкретной машины Тьюринга выбирается какой-либо один алфавит. Кроме ленты, у машины Тьюринга есть *автомат*, который может двигаться вдоль ленты и по очереди «обозревать» содержимое ячеек. Входное слово размещается на ленте по одной букве в расположенных подряд ячейках и занимает конечное число ячеек. Слева и справа от входного слова на ленте находятся только пустые ячейки.

#### **Практическая часть**

1. Выполнить на Машине Тьюринга прилагаемые в файле примеры задач согласно вариантов, заданных в табл. 1. В качестве исходных данных использовать последнюю цифру в зачетной книжке, заданную в двоичной или десятичной системах счисления, либо символы, согласно условия задачи.

Таблица 1. Варианты заданий

| № варианта | Функции, выполняемые программой                                                             |
|------------|---------------------------------------------------------------------------------------------|
| 1          | Доработать программу wolf.tur, исключив из нее состояние Q4 за счет применения состояния Q0 |
| 2          | Прибавление 1 к десятичному числу                                                           |
| 3          | Записать на ленте поздравление с Новым годом!                                               |

|    |                                     |
|----|-------------------------------------|
| 4  | Вычитание 1 из десятичного числа    |
| 5  | Записать на ленте свою фамилию      |
| 6  | Записать на ленте шифр своей группы |
| 7  | Записать на ленте слово Turing      |
| 8  | Записать на ленте слово Alan        |
| 9  | Записать на ленте слово Metison     |
| 10 | Записать на ленте слово Wiener      |

Запустить программу Algo2000.exe и при помощи команды ИНТЕРПРЕТАТОР перевести ее в режим работы Машина Тьюринга. Используя файлы wolf.tur, add\_bit.tur и minus\_bit.tur, изучить принцип программирования на машине Algo2000.exe.

**Задание.** Разработать программу, решающую шараду:

1. с “Б” смертельная бываю;
- с “М” меха я пожираю;
- с “Р” актеру я нужна;
- с “С” для повара важна;
- б) с “К” в полях меня найдете;
- с “Г” романсы запоете;
- с “В” в косички заплетете;
- в) я с “У” - далекая планета;
- а с “И” - я в Азии страна.

#### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата

A4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

#### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Краткое описание машины Тьюринга
2. Реализация алгоритма в машине Тьюринга.
3. Основная гипотеза теории алгоритмов.

## Практическая работа №14. Машина Тьюринга. Построение машин Тьюринга.

**Цель практической работы:** приобрести навыки разработки программы вычисления функции машины Тьюринга

### Теоретическая часть

Рассмотрим решение следующей задачи.

**Построить машину Тьюринга для следующей задачи:**

Пусть даны два числа в троичной системе исчисления. Надо найти их сумму.

|                | q <sub>0</sub>                  | q <sub>2</sub>    | q <sub>3</sub>    | q <sub>4</sub>                  | q <sub>5</sub>                  |
|----------------|---------------------------------|-------------------|-------------------|---------------------------------|---------------------------------|
| 0              | q <sub>0</sub> 2L               | q <sub>2</sub> 0L | q <sub>4</sub> 1R | q <sub>4</sub> 0R               |                                 |
| 1              | q <sub>2</sub> 0L               | q <sub>2</sub> 1L | q <sub>4</sub> 2R | q <sub>4</sub> 1R               |                                 |
| 2              | q <sub>2</sub> 1L               | q <sub>2</sub> 2L | q <sub>3</sub> 0L | q <sub>4</sub> 2R               | q <sub>5</sub> a <sub>0</sub> R |
| +              | q <sub>5</sub> a <sub>0</sub> R | q <sub>3</sub> +L |                   | q <sub>4</sub> +R               |                                 |
| a <sub>0</sub> |                                 |                   |                   | q <sub>0</sub> a <sub>0</sub> L | q <sub>1</sub> a <sub>0</sub>   |

Здесь **внутренний алфавит**

q<sub>0</sub> - начальное состояние системы;

q<sub>1</sub> - конечное состояние системы;

q<sub>2</sub>, q<sub>3</sub>, q<sub>4</sub>, q<sub>5</sub> - промежуточные состояния.

**внешний алфавит:**

0, 1, 2, +, a<sub>0</sub>.

Возьмем два конкретных числа. Например, 102<sub>3</sub> и 211<sub>3</sub>.

Информационная лента будет выглядеть следующим образом:

|                |   |   |   |   |   |   |   |                |
|----------------|---|---|---|---|---|---|---|----------------|
| a <sub>0</sub> | 1 | 0 | 2 | + | 2 | 1 | 1 | a <sub>0</sub> |
| q <sub>0</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 0 | 2 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>2</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 0 | 2 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>2</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 0 | 2 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>2</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 0 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>3</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>3</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>4</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>4</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>4</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>4</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |
| q <sub>4</sub> |   |   |   |   |   |   |   |                |
| a <sub>0</sub> | 1 | 1 | 0 | + | 2 | 1 | 0 | a <sub>0</sub> |

|       |   |   |   |   |   |   |   |       |
|-------|---|---|---|---|---|---|---|-------|
| $q_0$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 1 | 0 | + | 2 | 1 | 2 | $a_0$ |
| $q_0$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 1 | 0 | + | 2 | 0 | 2 | $a_0$ |
| $q_2$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 1 | 0 | + | 2 | 0 | 2 | $a_0$ |
| $q_2$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 1 | 0 | + | 2 | 0 | 2 | $a_0$ |
| $q_3$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 1 | 1 | + | 2 | 0 | 2 | $a_0$ |
| $q_4$ |   |   |   |   |   |   |   |       |
| $a_0$ | 1 | 2 | 0 | + | 2 | 0 | 2 | $a_0$ |
| $q_4$ |   |   |   |   |   |   |   |       |

и т.д.

2) По аналогии с заданием № 1 построить машину Тьюринга для сложения 2-х чисел, причем 1-е дано в восьмеричной системе, второе в троичной системе счисления.

Результат записать в виде программы команд.

Приведенный пример можно интерпретировать и таким образом: пусть необходимо сконструировать машину Тьюринга, которая сможет выступить в роли троично-восьмеричного дешифратора. Для этого достаточно левое слагаемое считать равным нулю. Тогда, уменьшая правое троичное число единица за единицей, мы построим слева от знака «+» новое число, равное уничтоженному, но записанное уже в восьмеричной системе.

|       | $q_0$     | $q_2$   | $q_3$   | $q_4$     | $q_5$     |
|-------|-----------|---------|---------|-----------|-----------|
| 0     | $q_02L$   | $q_20L$ | $q_41R$ | $q_40R$   |           |
| 1     | $q_20L$   | $q_21L$ | $q_42R$ | $q_41R$   |           |
| 2     | $q_21L$   | $q_22L$ | $q_43R$ | $q_42R$   | $q_5a_0R$ |
| 3     |           |         | $q_44R$ | $q_43R$   |           |
| 4     |           |         | $q_45R$ | $q_44R$   |           |
| 5     |           |         | $q_46R$ | $q_45R$   |           |
| 6     |           |         | $q_47R$ | $q_46R$   |           |
| 7     |           |         | $q_30L$ | $q_47R$   |           |
| +     | $q_5a_0R$ | $q_3+L$ |         | $q_4+R$   |           |
| $a_0$ |           |         |         | $q_0a_0L$ | $q_1a_0$  |

3) Построить машину Тьюринга, которая правильно вычисляет функцию  $f(x) = x + 1$  (составить таблицу).

4) На информационной ленте Тьюринга в трех секциях в произвольном порядке записаны три различные буквы: А, В, С. Каретка в состоянии  $q_0$  обозревает букву, расположенную справа (крайнюю справа). Необходимо составить функциональную схему машины Тьюринга, которая сумеет поменять местами крайние буквы.

5) Построить машину Тьюринга для вычисления функции  $f(x) = n + 2m$

6) Какую функцию вычисляет данная машина Тьюринга:

|       | $q_0$   | $q_2$    |
|-------|---------|----------|
| 0     | $q_21L$ | $q_20L$  |
| 1     | $q_00L$ | $q_21L$  |
| $a_0$ |         | $q_1a_0$ |

**Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

#### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Реализация алгоритма в машине Тьюринга.
2. Основная гипотеза теории алгоритмов.
3. Тезис Тьюринга. Проблема останова.

## Практическая работа №15. Машина Поста.

**Цель и содержание работы:** овладеть умениями составлять программы для машины Поста

### Теоретическое обоснование

#### Устройство машины Поста

Машина Поста - абстрактная машина, которая состоит из ленты и каретки (считывающая и записывающая головка). Лента бесконечна и разделена на секции одинакового размера. В каждую секцию ленты заносится один символ двоичной информации, который подлежит обработке. Один из символов двоичного алфавита - метка "V", другой - пустота. Если в секцию занесена "V" - секция отмеченная, если в секции "V" нет - секция пустая или неотмеченная.

Каретка может передвигаться вдоль ленты влево и вправо. Когда она неподвижна, она стоит против ровно одной секции ленты; говорят, что каретка обозревает эту секцию. А такая секция называется текущей или обозреваемой.

За единицу времени, которая называется шагом, каретка может сдвинуться на одну секцию влево или вправо. Кроме того, каретка может также распознать стоит или нет метка в обозреваемой ею секции, может заносить метку в пустую секцию и может удалять метку из отмеченной ячейки.

Команды, по которым каретка должна занести метку в отмеченную секцию или удалить метку из пустой секции являются недопустимыми.

#### Команды машины Поста

**Формат команды:**  $nKt$ , где:

$n$  - номер текущей команды,

$K$  - команда из системы команд машины Поста,

$t$  - отсылка - номер команды, которая будет выполняться следующей.

Последовательность команд из системы команд - программа, если:

- 1) на  $n$  - ом месте этой программы будет стоять команда с номером  $n$ .
- 2) отсылке  $t$  соответствует реальная команда в программе.

#### Система команд машины Поста:

1).  $a \rightarrow b$  Сдвиг каретки вправо, содержимое ленты не меняется.  
2).  $a \leftarrow b$  Сдвиг каретки влево, содержимое ленты не меняется.  
3).  $a \ V \ b$  В обозреваемую секцию ставится метка "V". Выполнение этой команды возможно только в том случае, если обозреваемая секция пустая, в противном случае команда считается невыполнимой.

4).  $a \Leftrightarrow b$  Каретка стирает метку в обозреваемой секции. Выполнение этой команды возможно только в том случае, если обозреваемая секция содержит метку, в противном случае команда считается невыполнимой.

5).  $a? \ b1, \ b2$  Команда передачи. Проверяется содержимое текущей секции, если метки нет, то происходит передача управления команде с номером  $b1$ , иначе, если метка есть - команде с номером  $b2$ . Содержимое ленты не меняется.

6).  $a! \ [b]$  Команда останова машины. Содержимое ленты не меняется. У команды останова отсылка не обязательна.

### Практическая часть

1. На ленте расположен массив в  $N$  отмеченных секций. Необходимо справа от данного массива через одну пустую секцию разместить массив вдвое больший (он должен состоять из  $2N$  меток). Предложите программу для решения данной задачи. Исходный массив может быть стерт.

2. На ленте расположен массив, состоящий из  $2N-1$  меток. Составить программу отыскания средней метки и стирания ее.

3. На ленте дан массив, содержащий  $2N$  меток. Составить программу, действуя по которой машина раздвинет на расстояние в одну секцию две половины данного массива.

4. На ленте расположен массив из  $N$  меток. Составить программу, действуя по которой машина выяснит, делится ли число  $N$  на 3.

5. На ленте два массива расположены на расстоянии в одну секцию. Левый массив содержит  $M$  меток, правый -  $N$  меток. Левый больше правого. Найти разность. По окончании работы машины на ленте должно остаться  $M - N$  меток.

6. На ленте два массива в  $M$  и  $N$  меток. Составить программу выяснения, одинаковы ли массивы по длине.

7. На ленте расположены два массива. Составить программу стирания большего.

### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4.

В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Как нумеруются секции на ленте машины Поста?
2. Что записывается в секции на ленте в машине Поста?
3. Как может двигаться каретка в машине Поста?
4. Сколько секций обзорекает каретка за один раз в машине Поста?
5. Сколько команд у машины Поста?
6. Как называется число, стоящее в конце команды машины Поста?

### **Машина Поста**

**Цель и содержание работы:** развить у студентов математическую интуицию, повысить уровень их математической культуры путем закрепления навыков работы на Маchine Поста.

### **Теоретическое обоснование**

#### **Таблица машины Поста**

В нее непосредственно вписывается алгоритм решения задачи. В столбце "Номер" формируются номера команд алгоритма, начиная с 1, с шагом 1. В столбце "Команды" указываются команды машины Поста. Они выбираются из выпадающего по нажатию клавиши "Enter" или щелчку мыши списка.

В столбце "Отсылка" указываются номера отсылок, т.е. номер строки, которая должна выполняться следующей. Значения в этом столбце не должны быть меньше 1 и больше номера последней команды. В столбце "Комментарии" можно писать различные пояснения, а можно ничего не писать. При выполнении этот столбец игнорируется. Чтобы удалить, вставить, добавить строки, очистить строки или столбцы воспользуйтесь соответствующими пунктами главного или всплывающего меню или панелью инструментов.

| Таблица 1. Система команд машины Поста |                                                     |
|----------------------------------------|-----------------------------------------------------|
| $a \rightarrow b$                      | Сдвиг каретки вправо, содержимое ленты не меняется. |
| $a \leftarrow b$                       | Сдвиг каретки влево, содержимое ленты не меняется.  |



|                |                                                                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $a \vee b$     | В обозреваемую секцию ставится метка "V". Выполнение этой команды возможно только в том случае, если обозреваемая секция пустая, в противном случае команда считается невыполнимой.                                   |
| $a \neq b$     | Каретка стирает метку в обозреваемой секции. Выполнение этой команды возможно только в том случае, если обозреваемая секция содержит метку, в противном случае команда считается невыполнимой.                        |
| $a ? b_1, b_2$ | Команда передачи. Проверяется содержимое текущей секции, если метки нет, то происходит передача управления команде с номером $b_1$ , иначе, если метка есть - команде с номером $b_2$ . Содержимое ленты не меняется. |
| $a ! [ b ]$    | Команда останова машины. Содержимое ленты не меняется. У команды останова ссылка не обязательна.                                                                                                                      |

### Задания

2.1. Выполнить на Машине Поста функцию сложения двух чисел, заданных в позиционной системе счисления. В качестве исходных данных использовать две последние цифры в зачетной книжке, заданные в позиционной системе счисления. Пример программы сложения двух чисел, заданных в позиционной системе счисления для машины Поста приведен в табл. 2.

| Таблица 2. Пример программы сложения двух чисел для машины Поста |        |   |    |                                                           |
|------------------------------------------------------------------|--------|---|----|-----------------------------------------------------------|
|                                                                  | --     |   |    | Поиск начала первого числа: сдвигаемся влево              |
| 2                                                                | ?      | 3 | 1  | до тех пор, пока не встретим неотмеченную ячейку.         |
| 3                                                                | ->     | 4 |    | Сдвигаемся вправо, на 1-ую метку первого числа и          |
| 4                                                                | $\neq$ | 5 |    | удаляем ее                                                |
| 5                                                                | -->    | 6 |    | ищем конец первого числа: сдвигаемся вправо,              |
| 6                                                                | ?      | 7 | 5  | пока каретка не встанет на неотмеченную ячейку и          |
| 7                                                                | V      | 8 |    | ставим метку                                              |
| 8                                                                | ->     | 9 |    | проверяем, заполнился ли промежуток между числами         |
| 9                                                                | ?      | 1 | 10 | если не заполнился - на первую строку, иначе - переход на |
| 10                                                               | !      |   |    | Конец                                                     |

2.2. Для машины Поста написать программу, которая ставит метку на позиции справа от текущей позиции каретки.

2.3. Записать в секции ленты 12 меток и составить программу, которая стирает:

- а) все нечетные метки справа налево;
- б) все четные метки справа налево;
- в) все нечетные метки слева направо;

г) все четные метки слева направо.

2.4. Найти на ленте драгоценный камень по методу Успенского.

### **Порядок работы машины Поста**

Работа машины Поста состоит в том, что каретка передвигается вдоль ленты и печатает или стирает метки. Чтобы машина Поста работала, надо задать некоторую программу и некоторое состояние машины (т.е. нужно как-то расставить метки по секциям ленты, в частности, можно все секции оставить пустыми, и поставить каретку против одной из секций).

Работа машины на основании заданной программы происходит следующим образом. Машина приводится в начальное состояние и приступает к выполнению первой команды программы. Эта команда выполняется за один шаг, после чего машины приступает к выполнению той команды, номер которой равен отсылке первой команды. Эта команда также выполняется за один шаг, после чего начинается выполнение той команды, номер команды которой равен отсылке предыдущей команды.

Вообще каждая команда выполняется за один шаг, а переход от выполнения одной команды к выполнению другой происходит по следующему правилу: пусть на  $k$ -ом шаге выполнялась команда с номером  $a$ , тогда:

1) если эта команда имеет единственную отсылку  $b$ , то на  $(k + 1)$ -ом шаге выполняется команда с номером  $b$ ;

2) если эта команда имеет две отсылки  $b_1$  и  $b_2$ , то на  $(k + 1)$ -ом шаге выполняется одна из двух команд - с номером  $b_1$  или с номером  $b_2$ ;

3) если же выполняющаяся на  $k$ -ом шаге команда вовсе не имеет отсылок, то на  $(k + 1)$  - ом шаге и на всех последующих шагах не выполняется никакая команда – машина останавливается.

Возможные случаи останова машины Поста:

1) в ходе выполнения программы машина дойдет до выполнения невыполнимой команды; выполнение программы прекращается, машина останавливается - происходит безрезультатная остановка.

2) в ходе выполнения программы машина дойдет до выполнения команды остановки;

программа в этом случае считается выполненной, машина останавливается - происходит результативная остановка.

3) в ходе выполнения программы машина не дойдет до выполнения ни одной из команд, указанных в первых двух вариантах; выполнение программы при этом никогда не прекращается, машина никогда не останавливается - процесс работы машины происходит бесконечно.

### **Информационная лента машины Поста**

Лента состоит из 1999 ячеек, нумерация от -999 до 999. Ячейка с толстой рамкой, находящаяся в центре ленты - каретка. Всплывающее меню ленты вызывается при щелчке правой кнопкой мыши на ленте. При получении фокуса лентой на ней появляется курсор (синий прямоугольник). Его можно сдвигать по ленте вправо и влево клавишами управления курсором.

Удерживая клавишу Shift и нажимая клавиши "Влево"/ "Вправо" можно выделить несколько ячеек. Также ячейки можно выделить мышью: удерживая левую кнопку и выделив нужные. Метки ставятся/ удаляются в ячейках, которые выделены. Это можно сделать несколькими способами (лента должна быть сфокусирована):

- клавишей "Пробел"
- щелкнуть мышью на кнопку "V" на панели инструментов
- выбрать пункт главного или всплывающего меню "поставить/ удалить метки"
- двойным щелчком мыши (в этом случае изменение происходит только в одной ячейке).

Клавиши управления курсором "Вверх" или "Вниз" ставят курсор в каретку.

"Home" - курсор в начало видимой части ленты.

"End" - курсор в конец видимой части ленты.

"Ctrl" + "Влево" - сдвиг каретки влево

"Ctrl" + "Вправо" - сдвиг каретки вправо

Кнопки со стрелкой слева и справа от ленты - прокрутить каретку соответственно влево и вправо. Кнопки со стрелкой и вертикальной палочкой слева и справа от ленты - прокрутить каретку соответственно до крайней левой отмеченной ячейки и до крайней правой отмеченной ячейке. Номера этих ячеек указываются в строке состояния как мин и макс соответственно. Можно непосредственно поставить каретку в нужную ячейку, указав ее номер в редакторе над кареткой. Ленту можно запомнить (пункт меню Команда/ Запомнить ленту) и затем восстановить (Команда/ Восстановить ленту).

### **Исполнение алгоритма машины Поста**

Запуск алгоритма на исполнение осуществляется:

- пунктом главного меню Пуск/ Запустить
- кнопкой Запустить на панели инструментов

При этом выполнение программы будет идти до тех пор, пока не встретится команда стоп или не возникнет какая-нибудь ошибка.

Приостановить выполнение можно:

- пунктом главного меню Пуск/ Пауза;
- кнопкой Пауза на панели инструментов.

Если выполнение не было приостановлено, то оно всегда начинается с первой команды алгоритма. Если была нажата пауза, то можно продолжить выполнение с той команды, на которой машина остановилась. Программу можно выполнить по шагам:

- пунктом главного меню Пуск/ Пошагово;
- кнопкой Пошагово на панели инструментов.

При этом выполнится очередная команда и выполнение приостановится. Полностью прервать выполнение программы можно:

- пунктом главного меню Пуск/ Прервать;
- кнопкой Прервать на панели инструментов.

Регулировать скорость выполнения можно:

- пунктом главного меню Скорость

### ***Ошибки машины Поста***

При возникновении этих ошибок происходит прерывание исполнения алгоритма.

- Не указана команда
- Не указана отсылка
- Неверно указан номер отсылки
- Команды с таким номером не существует
- Нельзя поставить метку, где она уже есть
- Нельзя стереть метку, где ее нет
- Ошибка чтения файла
- Ошибка записи файла

### **Содержание отчёта и его форма**

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата

A4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### **Контрольные вопросы и защита работы**

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Реализация алгоритма в машине Поста.
2. Основная гипотеза теории алгоритмов.
3. Из чего состоит машина Поста?

## Практическая работа №16. Нормальные алгоритмы Маркова

**Цель и содержание работы:** овладеть умениями составлять нормальные алгоритмы Маркова

### Теоретическое обоснование

В алгоритмической системе Маркова нет понятия ленты и осуществляется непосредственный доступ к различным частям преобразуемого слова.

Нормальный алгоритм Маркова представляет собой упорядоченный набор формул подстановки. Формула подстановки служит для замены подслов в преобразуемом слове. Заменяемое и замещающее подслова в формуле разделяются одной из стрелок  $\rightarrow$  или  $|>$ . В левой и правой частях формулы подстановки могут содержаться пустые слова.

В качестве примера приведем алгоритм получения частного и остатка при делении на три:

(1)  $*\$ \$ \$ \rightarrow \$ *$

(2)  $* |> *$

(3)  $\rightarrow *$

Выполнение алгоритма распадается на шаги. Каждый шаг включает в себя поиск первой по порядку формулы подстановки, применимой к преобразуемому слову, и выполнение, соответствующей этой формуле, замены. Процесс выполнения алгоритма заканчивается в двух случаях:

- либо все формулы оказались неприменимы;
- либо на последнем шаге была применена формула, в которой левое и правое подслова разделяет стрелка  $|>$ .

Применим вышеприведенный алгоритм к данному входному слову. Например, к неотрицательному целому числу  $N$ , представленному в виде последовательности из  $N$  символов  $\$$  [\$\$\$ \$\$\$\$].

| Шаг | Формула                          | Преобразуемое слово |
|-----|----------------------------------|---------------------|
| 1   | (3) $\rightarrow *$              | *\$ \$\$\$\$ \$     |
| 2   | (1) $*\$ \$ \$ \rightarrow \$ *$ | \$*\$ \$\$\$\$      |
| 3   | (1) $*\$ \$ \$ \rightarrow \$ *$ | \$*\$ \$            |
| 4   | (2) $ > *$                       | \$*\$ \$            |

Т.е. в результате выполнения алгоритма у нас получилось частное 2 ( $*\$ \$ *$ ) и в остатке тоже 2 ( $*\$ \$ *$ ).

### Практическая часть

#### Задание 1.

**Составить нормальный алгоритм Маркова для решения следующих задач:**

1. Уменьшить на 1 число в двоичной системе счисления.
2. Увеличить на 1 число в двоичной системе счисления.
3. Вычислить количество нулевых разрядов в двоичном числе.
4. Вычислить количество единичных разрядов в троичной системе счисления.
5. Вычислить сумму цифр числа в троичной системе счисления.
6. Вычислить сумму цифр двоичного числа.
7. Преобразовать число из двоичной системы счисления в троичную.

## Задание 2.

1. Определите нормальный алгоритм, который уменьшает число на единицу.
2. Определите нормальный алгоритм сложения двух двоичных чисел методом уменьшения одного числа на 1 и увеличением другого числа на 1 до тех пор, пока уменьшаемое число не станет равным 0.
3. Определите нормальный алгоритм логического сложения двух двоичных чисел.
4. Определите нормальный алгоритм логического умножения двух двоичных чисел.
5. Определите нормальный алгоритм сложения по модулю 2 двух двоичных чисел.
6. Определите нормальный алгоритм поразрядного сложения двух двоичных чисел.
7. Определите нормальный алгоритм вычитания двоичных чисел.
8. Определите нормальный алгоритм умножения двух двоичных чисел столбиком.
9. Определите нормальный алгоритм деления двух двоичных чисел с определением частного и остатка.
10. Определите нормальный алгоритм вычисления наибольшего общего делителя двух двоичных чисел.
11. Определите нормальный алгоритм вычисления наименьшего общего кратного двух двоичных чисел.

### Содержание отчёта и его форма

Отчёт предоставляется в письменной или печатной форме в тетради или на листах формата А4. В отчёте указываются:

1. Порядковый номер и название практической работы.
2. Цель и содержание работы.
3. Письменные ответы на контрольные вопросы.
4. Описание решения выполненных заданий практической работы

### Контрольные вопросы и защита работы

Защита практической работы осуществляется по отчёту, представленному студентом и демонстрации задания на компьютере.

1. Определите нормальный алгоритм, который уменьшает число на единицу.
2. Определите нормальный алгоритм сложения двух двоичных чисел методом уменьшения одного числа на 1 и увеличением другого числа на 1 до тех пор, пока уменьшаемое число не станет равным 0.
3. Определите нормальный алгоритм логического сложения двух двоичных чисел.
4. Определите нормальный алгоритм логического умножения двух двоичных чисел.
5. Определите нормальный алгоритм сложения по модулю 2 двух двоичных чисел.
6. Определите нормальный алгоритм поразрядного сложения двух двоичных чисел.
7. Определите нормальный алгоритм вычитания двоичных чисел.
8. Определите нормальный алгоритм умножения двух двоичных чисел столбиком.
9. Определите нормальный алгоритм деления двух двоичных чисел с определением частного и остатка.
10. Определите нормальный алгоритм вычисления наибольшего общего делителя двух двоичных чисел.
11. Определите нормальный алгоритм вычисления наименьшего общего кратного двух двоичных чисел.

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### Основная литература:

1. Лавров, И. А. Задачи по теории множеств, математической логике и теории алгоритмов / И. А. Лавров, Л. Л. Максимова. - Москва: Физматлит, 2021. - 256 с. - ISBN 978-5-9221-1721-5.
2. Мальцев, А. И. Алгоритмы и рекурсивные функции / А. И. Мальцев. - Москва: Лань, 2020. - 368 с. - ISBN 978-5-8114-4888-4.
3. Судоплатов, С. В. Математическая логика и теория алгоритмов: учебник / С. В. Судоплатов, Е. В. Овчинникова. - Москва: ИНФРА-М, 2022. - 432 с. - ISBN 978-5-16-015789-2.
4. Шень, А. Лекции по математической логике и теории алгоритмов / А. Шень. - Москва: МЦНМО, 2023. - 288 с. - ISBN 978-5-4439-1675-8.

### Дополнительная литература:

1. Брыкалова А.А. Теория алгоритмов [Электронный ресурс]: лабораторный практикум / А.А. Брыкалова. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 134 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69439.html>
2. Макоха А.Н. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / А.Н. Макоха, А.В. Шапошников, В.В. Бережной. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 418 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69397.html>
3. Теория алгоритмов: учебное пособие / Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет»; сост. А.А. Брыкалова. - Ставрополь: СКФУ, 2016. - 129 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=467402>
4. Брыкалова А.А. Теория алгоритмов [Электронный ресурс]: лабораторный практикум / А.А. Брыкалова. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 134 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69439.html>
5. Макоха А.Н. Математическая логика и теория алгоритмов [Электронный ресурс]: учебное пособие / А.Н. Макоха, А.В. Шапошников, В.В. Бережной. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2017. — 418 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69397.html>
6. Теория алгоритмов: учебное пособие / Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет»; сост. А.А. Брыкалова. - Ставрополь: СКФУ, 2016. - 129 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=467402>
7. Перемитина, Т.О. Математическая логика и теория алгоритмов: учебное пособие / Т.О. Перемитина; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск: ТУСУР, 2016. - 132 с.: ил. - Библиогр.: с.130.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=480886>
- 8.