

**Принципы обработки информации компьютером. Логические основы работы компьютера. Алгоритмы и способы их описания.**



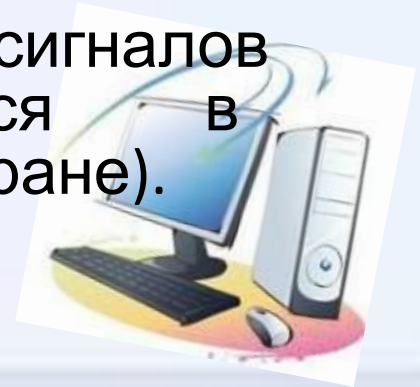
# **Компьютер и его функциональное устройство**

**Компьютер** – это техническое средство преобразования информации, в основу работы которого заложены те же принципы обработки электрических сигналов, что и в любом электронном устройстве:

1. Входная информация, представленная различными физическими процессами, как электрической, так и неэлектрической природы (буквами, цифрами, звуковыми сигналами и т.д.), преобразуется в электрический сигнал;

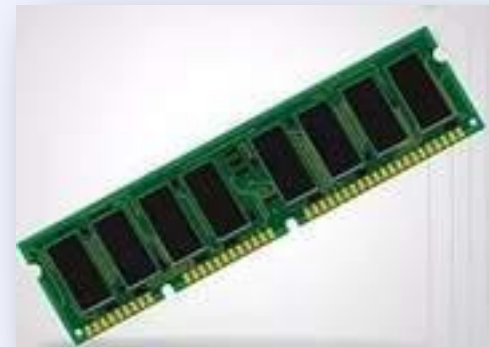
2. Сигналы обрабатываются в блоке обработки;

3. С помощью преобразователя выходных сигналов обработанные сигналы преобразуются в неэлектрические сигналы (изображения на экране).



С позиции функционального назначения компьютер – это система, состоящая из 4-х основных устройств, выполняющих определенные функции: запоминающего устройства или памяти, которая разделяется на оперативную и постоянную, арифметико-логического устройства (АЛУ), устройства управления (УУ) и устройства ввода-вывода (УВВ).

Запоминающее устройство (память) предназначается для хранения информации и команд программы в ЭВМ. Информация, которая хранится в памяти, представляет собой закодированные с помощью 0 и 1 числа, символы, слова, команды, адреса и т.д.



### **Характеристики памяти :**

- 1) емкость памяти – максимальное количество хранимой информации в байтах;
- 2) быстродействие памяти – время обращения к памяти, определяемое временем считывания или временем записи информации.

- Виды памяти

- Внутренняя

- ОЗУ

- ПЗУ

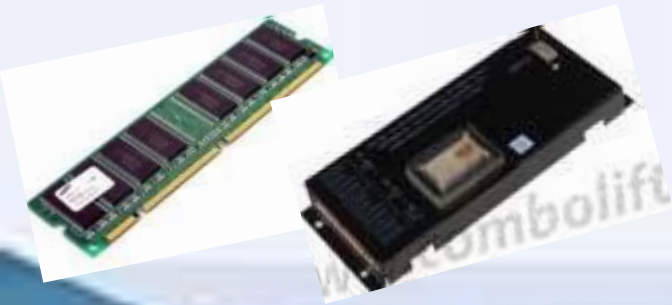
- Внешняя

- Диски

- Флешки

- Дискеты

- Магнитные ленты

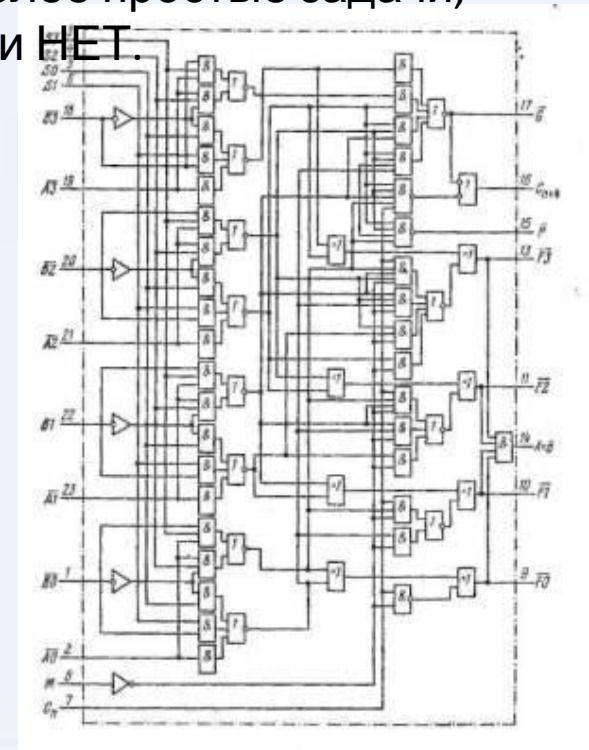


## Арифметико-логическое устройство (АЛУ).

Производит арифметические и логические действия.

Следует отметить, что любую арифметическую операцию можно реализовать с использованием операции сложения.

Сложная логическая задача раскладывается на более простые задачи, где достаточно анализировать только два уровня: ДА и НЕТ.



**Устройство управления (УУ)** управляет всем ходом вычислительного и логического процесса в компьютере, т.е. выполняет функции "регулирующего движителя" информации. УУ читает команду, расшифровывает ее и подключает необходимые цепи для ее выполнения. Считывание следующей команды происходит автоматически.

Фактически УУ выполняет следующий цикл действий:

1. формирование адреса очередной команды;
2. чтение команды из памяти и ее расшифровка;
3. выполнение команды.





- В современных компьютерах функции УУ и АЛУ выполняет одно устройство, называемое центральным процессором.



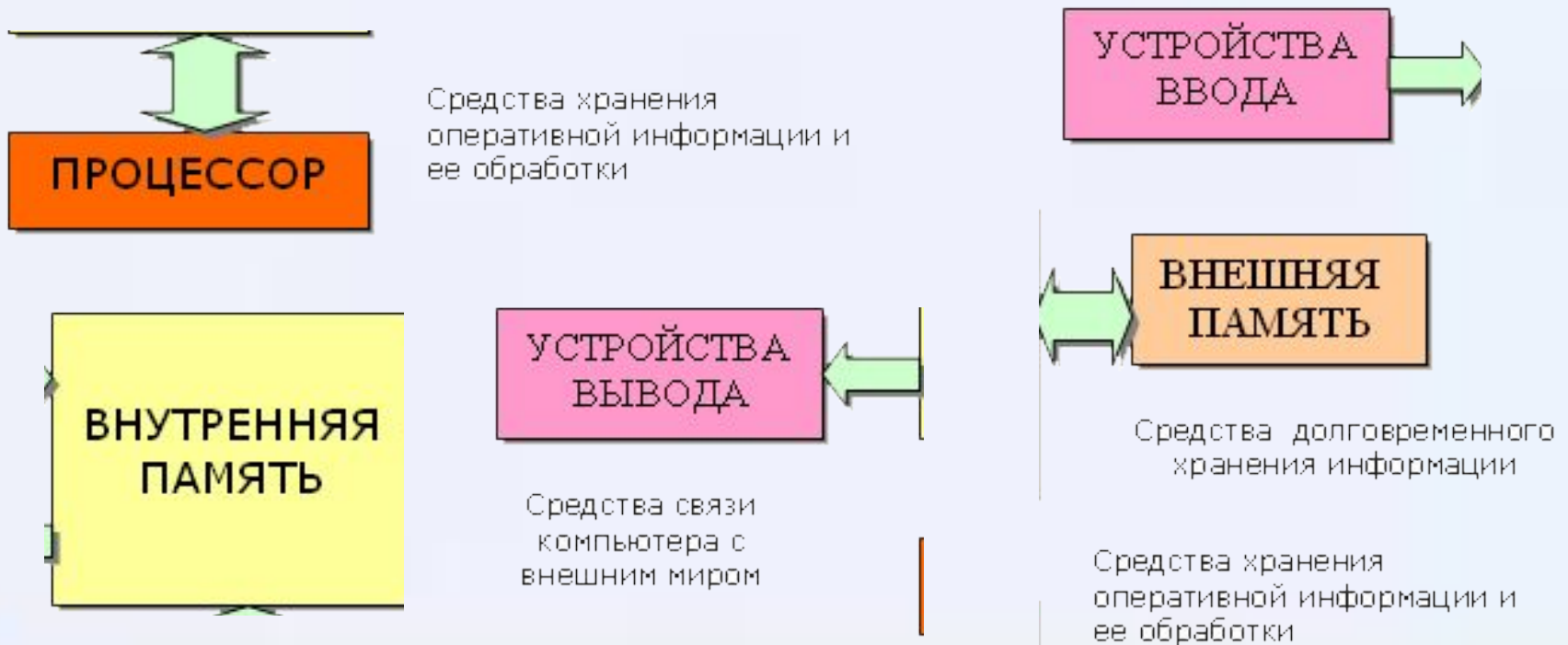
**Устройства ввода и вывода** - устройства взаимодействия компьютера с внешним миром: с пользователями или другими компьютерами.

**Устройства ввода** позволяют вводить информацию в компьютер для дальнейшего хранения и обработки.

**Устройства вывода** - получать информацию из компьютера.

# Задание 1.

- Из данных блоков составьте функциональную схему компьютера



# Решение



# **АРИФМЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ ЭВМ**

Правила выполнения арифметических действий над двоичными числами задаются таблицами сложения, вычитания и умножения.

| <b>Сложение</b> | <b>Вычитание</b> | <b>Умножение</b> |
|-----------------|------------------|------------------|
| $0+0 = 0$       | $0 - 0 = 0$      | $0 \times 0 = 0$ |
| $0+1 = 1$       | $1 - 0 = 1$      | $0 \times 1 = 0$ |
| $1+0 = 1$       | $1 - 1 = 0$      | $1 \times 0 = 0$ |
| $1+1 = 10$      | $10 - 1 = 1$     | $1 \times 1 = 1$ |

- **Система счисления** – это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых цифрами.
- Все системы счисления делятся на две большие группы: позиционные и непозиционные системы счисления.
- В позиционных СС количественное значение цифры зависит от ее положения в числе.
- Рассмотрим вначале позиционные СС, например, десятичную СС. Число 579. Цифра 5 обозначает пять сотен, 7 – семь десятков, 9 – девять единиц.
- Если поменять местами цифры, например, 5 и 7, то цифра 5 – станет обозначать пять десятков, 7 – семь сотен.
- Одним из примеров непозиционных СС является римская СС (римские числа). Давайте подробнее рассмотрим, по какому принципу образуются числа в римской СС.

- Вся информация в компьютере представлена в виде двоичного кода. Компьютер переводит информацию (числовую, текстовую, графическую, звуковую, видео) в последовательность нулей и единиц. Давайте посмотрим, как можно перевести числа из привычной нам десятичной СС в двоичную СС.
- **Перевод целых чисел из десятичной СС в двоичную:**
- Последовательно выполнять деление исходного целого десятичного числа и получаемых целых частных на основание системы (на 2) до тех пор, пока не получится частное, меньшее делителя, то есть меньшее 2.
- Записать полученные остатки в обратной последовательности.



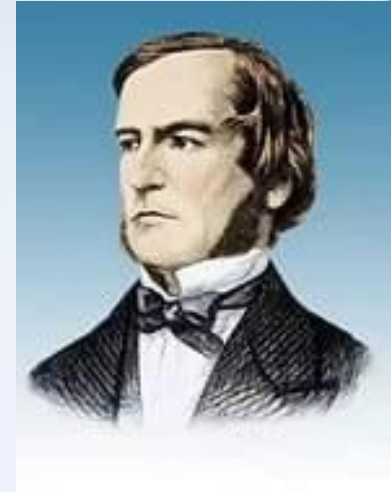
- Алгоритм перевода:
- Двоичное число записать в развернутой форме. Давайте вернемся в курс математики и вспомним, как записывается число в развернутой форме. Запишем число 579 в десятичной СС в развернутой форме.
- Мы уже с вами выяснили, что в э том числе цифра 5 означает 5 сотен, 7 – семь десятков, 9 – девять единиц. Число 579 записано в привычной для нас свернутой форме. Мы настолько привыкли к такой форме записи, что уже не замечаем, как в уме умножаем цифры числа на различные степени числа 10.
- В развернутой форме записи числа такое умножение записывается в явной форме.
-

# Задание 2

- 1. Переведите числа из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.
  - а) 948;
  - б) 763;
  - в) 994,125;
  - г) 523,25;
  - д) 203,82.
- 2. Переведите числа в десятичную систему счисления.
  - а)  $111000111_2$ ;
  - б)  $100011011_2$ ;
  - в)  $1001100101,1001_2$ ;
  - г)  $1001001,011_2$ ;
  - д)  $335,7_8$ ;
  - е)  $14C,A_{16}$ .

# Алгебра логики

Для описания логики функционирования аппаратных и программных средств ЭВМ используется или, как ее часто называют, *булева алгебра* (по имени основоположника этого раздела математики – Дж. Буля).



Булева алгебра оперирует логическими переменными, которые могут принимать только два значения: *истина* или *ложь* (true или false), обозначаемые соответственно 1 и 0.

**Логической функцией** называется функция, которая может принимать только 2 значения – истина или ложь (1 или 0). Любая логическая функция может быть задана с помощью *таблицы истинности*. В левой ее части записываются возможные наборы аргументов, а в правой – соответствующие им значения функции.

**Логическая операция ИНВЕРСИЯ** (операция отрицания)  
— новое высказывание, которое ложно, когда высказывание истинно и истинно, когда само высказывание ложно.  
Соответствует частице **НЕ**, обозначается:

$$\neg A$$
$$\overline{A},$$

**Таблица истинности**

| A | $\neg A$ |
|---|----------|
| 0 | 1        |
| 1 | 0        |

# Логическая операция **КОНЪЮНКЦИЯ**

Конъюнкция двух переменных истинна тогда и только тогда, когда оба высказывания истинны.

Соответствует союзу **И**, обозначается знаками  $\&$ ,  $\cdot$ ,  $*$ ,  $\wedge$

Таблица истинности

| <b>A</b> | <b>B</b> | <b>A ∧ B</b> |
|----------|----------|--------------|
| 0        | 0        | 0            |
| 0        | 1        | 0            |
| 1        | 0        | 0            |
| 1        | 1        | 1            |

# Логическая операция ДИЗЬЮНКЦИЯ

Дизъюнкция двух переменных ложна тогда и только тогда, когда оба высказывания ложны.

Соответствует союзу **ИЛИ**, обозначается знаками  $\vee$ ,  $+$ .

Таблица истинности

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

# Алгорит

# М



**Алгоритм** – система точных и понятных предписаний (команд, инструкций, директив) о содержании и последовательности выполнения конечного числа действий, необходимых для решения любой задачи данного типа.

В качестве **исполнителя** алгоритмов можно рассматривать человека, любые технические устройства, среди которых особое место занимает компьютер.

**Система команд исполнителя (СКИ)** – набор действий, которые может совершить исполнитель

## Свойства алгоритма

- **Дискретность** (от лат. discretus – разделенный, прерывистый) указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке.
- **Детерминированность** (от лат. determinate – определенность, точность) указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.
- **Конечность** определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.
- **Результативность** означает, при точном исполнении всех команд процесс решения задачи должен прекратиться за конечное число шагов и при этом должен быть получен определенный постановкой задачи результат (ответ).
- **Массовость**. Это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными, т. е. применять при решении всего класса задач данного типа, отвечающих общей постановке задачи.

# Типовые конструкции алгоритмов:

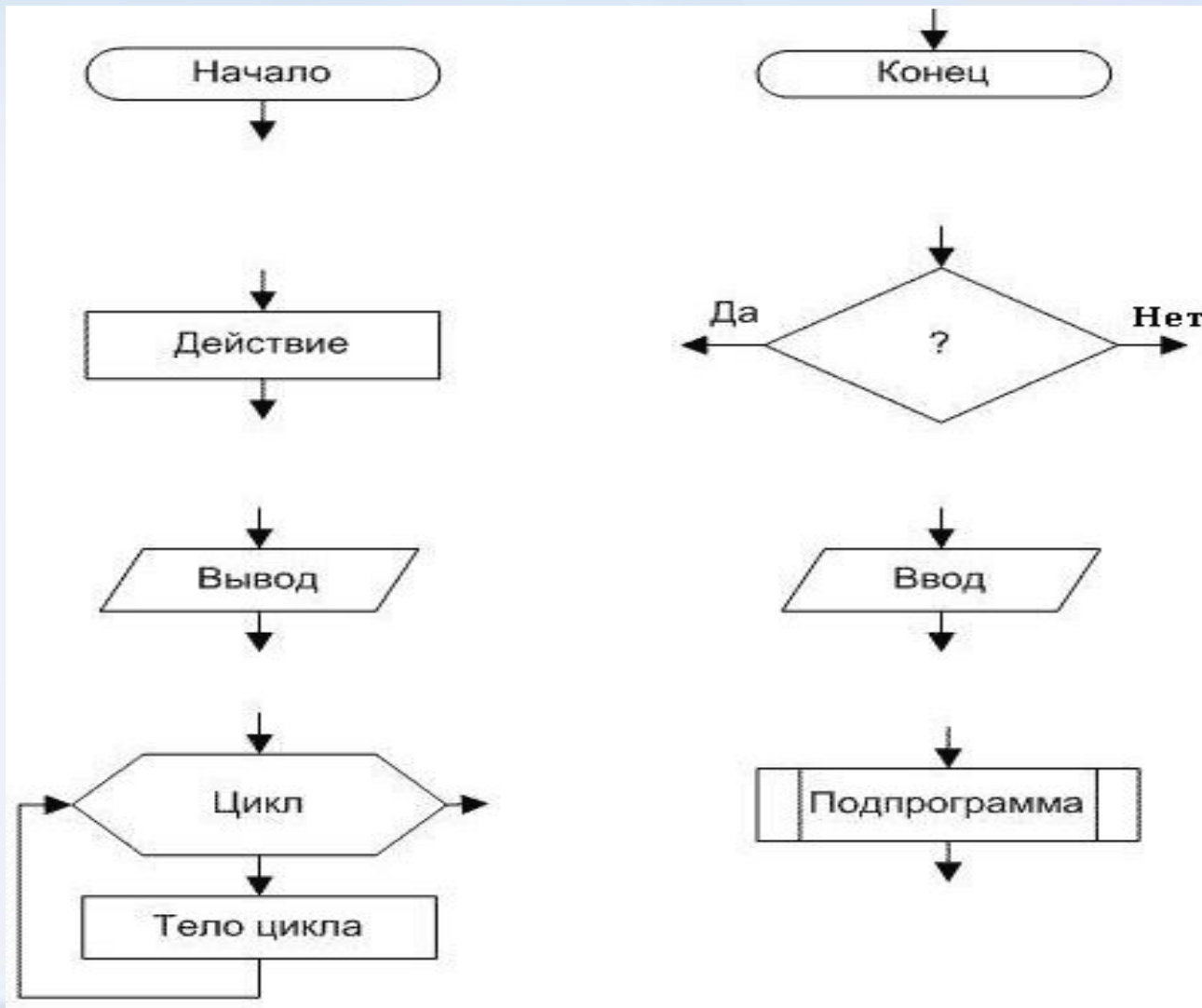
- Линейный.
- Циклический.
- Разветвляющийся.
- Вспомогательный.

- **Линейный** (последовательный) алгоритм – описание действий, которые выполняются однократно в заданном порядке.
- **Циклический** – описание действий или группы действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Совокупность повторяющихся действий – тело цикла.
- **Разветвляющийся** – алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий. Условие – выражение, находящееся между словом «если» и словом «то» и принимающее значение «истина» (ветвь «да») или «ложь» (ветвь «нет»). Возможна полная и неполная форма ветвления.
- **Вспомогательный** – алгоритм, который можно использовать в других алгоритмах, указав только его имя. Вспомогательному алгоритму должно быть присвоено имя.

# Способы описания алгоритмов.

- на естественном языке;
- на специальном (формальном) языке;
- с помощью формул, рисунков, таблиц;
- с помощью стандартных графических объектов (геометрических фигур) – блок-схемы.

# Основные элементы блок

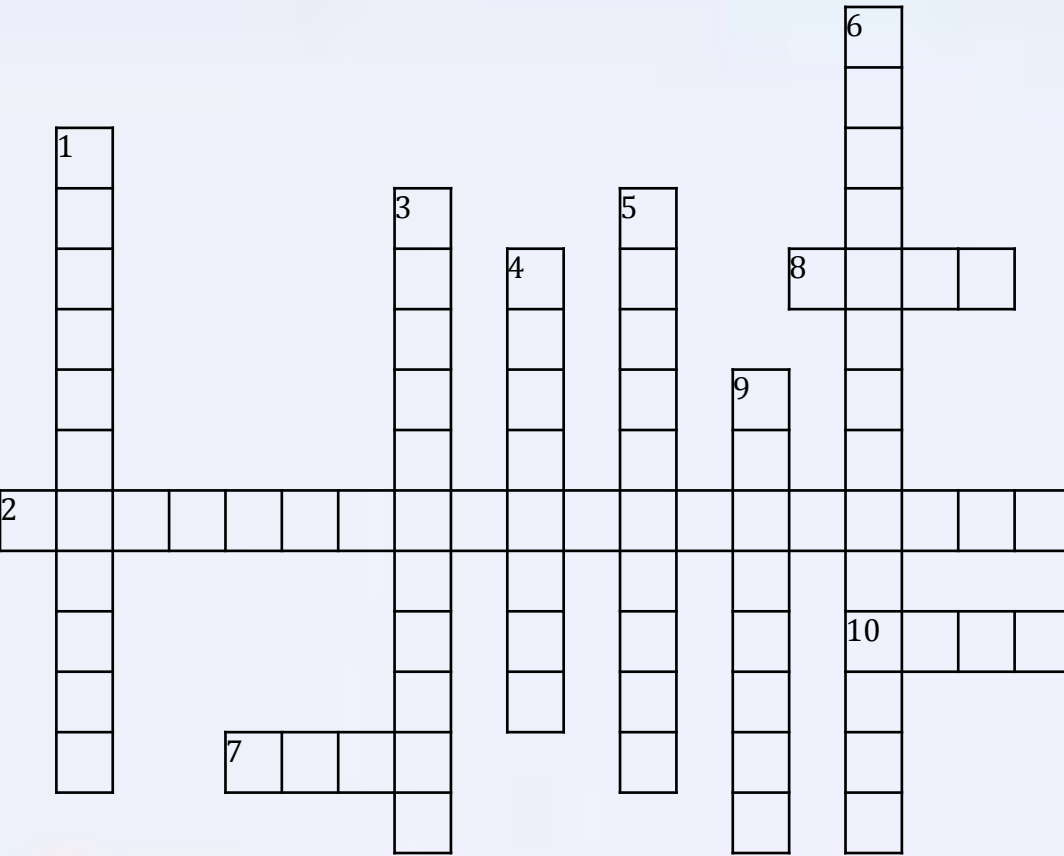


# Задание 3

Составьте блок-схему для решения  
полного квадратного уравнения  
 $ax^2+bx+c=0/$

# Задание 4

## Разгадайте кроссворд



### *По горизонтали:*

2. Свойство алгоритма, означающее однозначность действий.
7. Повторяющаяся последовательность действий.
8. Синоним слову алгоритм.
10. Фигура, в которой записывается условие в блок-схеме.

### *По вертикали:*

1. Способ описания алгоритма.
3. Объект, умеющий выполнять определенный набор действий.
4. Строго определенная последовательность действий при решении задачи.
5. Свойство, показывающие, что алгоритм можно применять для решения класса задач.
6. Фигура ввода-вывода данных.
9. Алгоритм, действия в котором выполняются однократно в заданном порядке.