

Федеральное агентство железнодорожного транспорта  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Иркутский государственный университет путей сообщения»  
СИБИРСКИЙ КОЛЛЕДЖ ТРАНСПОРТА И СТРОИТЕЛЬСТВА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ПРАКТИЧЕСКИМ И ЛАБОРАТОРНЫМ РАБОТАМ

по дисциплине

ОП.09 Основы алгоритмизации и программирования

по специальности

09.02.01 Компьютерные системы и комплексы

базовая подготовка среднего профессионального образования

Иркутск 2022

Электронный документ выгружен из ЕИС ФГБОУ ВО ИрГУПС и соответствует оригиналу

Подписант ФГБОУ ВО ИрГУПС Трофимов Ю.А.

00a73c5b7b623a969ccad43a81ab346d50 с 08.12.2022 14:32 по 02.03.2024 14:32 GMT+03:00

Подпись соответствует файлу документа



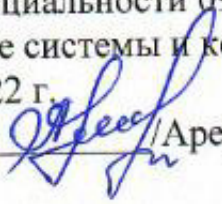
РАССМОТРЕНО:

Цикловой методической

комиссией специальности 09.02.01

Компьютерные системы и комплексы

«08» июня 2022 г.

Председатель:  /Арефьева Н.В.

СОГЛАСОВАНО:

Заместитель директора по УВР

/А.П.Ресельс

«09» июня 2022 г.

Разработчик: Фитисова Н.Н. преподаватель высшей категории Сибирского колледжа транспорта и строительства ФГБОУ ВО «Иркутский государственный университет путей сообщения».

## Оглавление

Практическая работа №1 .....	4
Лабораторная работа №1 .....	11
Практическая работа №2 .....	14
Практическая работа №3 .....	16
Практическая работа №4 .....	18
Практическая работа №5 .....	19
Практическая работа №6 .....	21
Практическая работа №7 .....	22
Практическая работа №8 .....	25
Перечень учебных изданий, интернет - ресурсов, дополнительной литературы:.....	26

## Практическая работа №1

### Формализация поставленной задачи

**Цель работы:** Овладеть техникой составления алгоритмов.

**Задачи работы:** Научиться составлять линейные и разветвляющиеся алгоритмы.

**Задание:** Составить алгоритмы, используя предложенные задания.

Ознакомьтесь с теоретическим материалом, необходимым для выполнения работы:

Алгоритм — совокупность последовательных шагов, схема действий, приводящих к желаемому результату.

Разработать алгоритм означает разбить задачу на определенную последовательность шагов. От разработчика алгоритма требуется знание особенностей и правил составления алгоритмов.

Основные особенности и свойства алгоритмов:

1. Наличие ввода исходных данных.
2. Наличие вывода результата выполнения алгоритма.
3. Дискретность.
4. Формальность.
5. Определенность (точность).
6. Понятность.
7. Результативность (конечность).
8. Корректность.
9. Массовость.
10. Эффективность.

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависит от того, кто будет исполнителем этого алгоритма.

Способы описания алгоритма:

1. Формульная запись
2. Табличная запись
3. Развернутая словесная
4. На алгоритмическом языке
5. Графический (в виде блок-схемы)
6. На языке программирования

Основным элементарным действием в вычислительных алгоритмах является присваивание значения переменной величине. Если значение константы определено видом ее записи, то переменная величина получает конкретное значение только в результате присваивания.

Присваивание – это операция, которая значение выражения, стоящее справа от символа «:=» запоминает в переменной или элементе массива, стоящем слева. При присваивании происходит преобразование типов данных, если они не совпадают.

Линейными называются алгоритмы, в которых все действия осуществляются последовательно друг за другом, при этом каждая команда выполняется только один раз строго после той команды, которая ей предшествует. Линейный алгоритм составляется из команд присваивания, ввода, вывода и обращения к вспомогательным алгоритмам.

Разветвляющимся называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий. Каждое из возможных направлений дальнейших действий называется ветвью. В блок-схемах разветвление реализуется специальным блоком «Решение». Этот блок предусматривает возможность двух выходов. В самом блоке «Решение» записывается логическое условие, от выполнения которого зависят дальнейшие действия.

Циклическим называется алгоритм, в котором некоторая часть операций (тело цикла – последовательность команд) выполняется многократно, т.е. более одного раза.

Различают:

- 1.циклы с известным числом повторений (или со счетчиком);
- 2.циклы с неизвестным числом повторений (циклы с предусловием и циклы с постусловием).

В любом цикле должна быть переменная, которая управляет выходом из цикла, т.е. определяет число повторений цикла.

В циклах со счетчиком известно число повторений цикла, т.е. оно является фиксированным числом. В этом случае переменная, которая считает количество повторений (шагов) цикла, называется счетчиком цикла (или параметром цикла, или управляющей переменной цикла).

Циклы с предусловием – это такие циклы, в которых до начала выполнения тела цикла проверяется условие выполнения следующего шага цикла. Если значение этого условия истинно (т.е. условие выполняется), то выполняется тело цикла.

Цикл с постусловием также используется при неизвестном заранее количестве повторений цикла, но в отличие от цикла с предусловием здесь условие на выход из цикла проверяется после того, как выполнились операторы тела цикла.

**Задание 1:** Составить линейные алгоритмы, используя предложенные задания.

**Рассмотрите примерные упражнения**

1. Составление алгоритма в табличной форме для вычисления выражения :

$$y = \frac{8x - 1}{4x + 2}$$

Исходные данные: x.

Результат: y.

Последовательность шагов и описание действий:

1.  $a := 8x$
2.  $b := a - 1$
3.  $c := 4x$
4.  $d := c + 2$
5.  $y := b/d$

2. Составление алгоритма вычисления площади поверхности цилиндра с диаметром D и высотой H в табличной форме.

Исходные данные: D, H.

Результат: S.

Последовательность шагов и описание действий:

1.  $a := \pi D^2$
2.  $b := a / 4$
3.  $c := \pi D H$
4.  $S := b + c$

3. Вычислить факториал числа n ( $c=n!$ ), который вычисляется по формуле  $c=1*2*3*4*\dots*n$ . Алгоритм представить в словесной форме.

Исходные данные: n.

Результат: c.

Алгоритм:

1. Полагаем  $c=1$  и переходим к следующему пункту.
2. Полагаем  $i=1$  и переходим к следующему пункту.
3. Полагаем  $c=i*c$  и переходим к следующему пункту.
4. Проверяем, равно ли  $i$  числу n. Если  $i=n$ , то вычисления прекращаем. Если  $i$  больше n, то увеличиваем  $i$  на 1 и переходим к пункту 3.

### **Выполнить самостоятельно**

Составьте алгоритмы решения следующих задач:

1. По заданным формулам составить вычислительные алгоритмы в виде таблиц:

$$S = \frac{\pi D^2}{4} + \frac{\pi D L}{2}$$

(S – площадь боковой поверхности конуса, D – диаметр основания, L – образующая);

$$f = \frac{1}{2\pi\sqrt{LC}}$$

( $f$  – частота собственных колебаний в контуре,  $L$  – индуктивность катушки,  $C$  – емкость конденсатора).

2. Дано значение  $x$ . Получить значения  $-2x+3x^2-4x^3$  и  $1+2x+3x^2-4x^3$ . Позаботиться об экономии операций.

3. Дано значение  $a$ . Не используя никаких функций и никаких операций, кроме умножения, получить значение  $a^8$  за три операции и  $a^{10}$  за четыре операции.

4. Составить алгоритм для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде  $v$  км/ч, скорость течения реки  $v_1$  км/ч, время движения по озеру  $t_1$  ч, а против течения реки –  $t_2$  ч. Использовать словесный способ записи алгоритма.

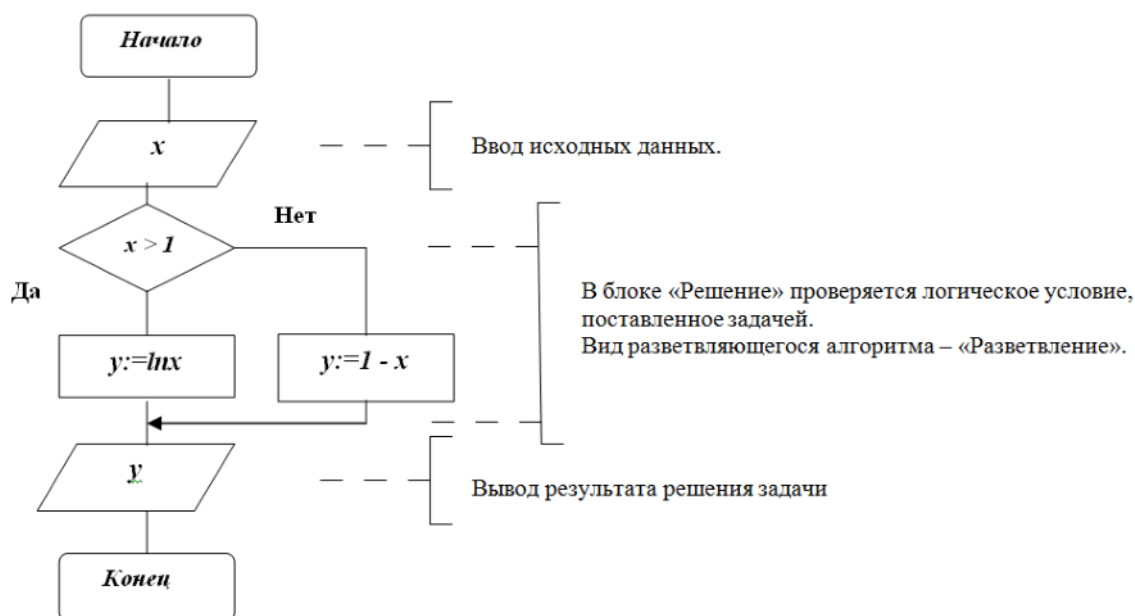
**Задание 2:** Составить разветвляющиеся алгоритмы, используя предложенные задания.

**Рассмотрите примерные упражнения:**

1. Составьте блок-схему вычисления значения функций:

$$y = \begin{cases} \ln x, & x > 1, \\ 1 - x, & x \leq 1 \end{cases}$$

Исходные данные:  $x$ . Результат:  $y$ .



**Выполнить самостоятельно**

Составьте алгоритмы решения следующих задач:

1. Изобразите блок-схему простого диалогового алгоритма, который обращается к пользователю с просьбой ввести сначала строку имя, а затем строку настроение. В результате диалога может появиться следующий совместный текст:

Программа> Здравствуйте! Как Ваше имя?

Пользователь> Гаврик

Программа> Доброе утро, Гаврик! Как настроение?

Пользователь> так себе

Программа> У меня тоже так себе, Гаврик!

2. Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень – отрицательные.

3. Составьте блок-схему вычисления значения функций:

$$y = \begin{cases} 5x^2 + 2, & \text{если } x > -4 \\ \frac{x+4}{3x^2+7}, & \text{если } x \leq -4 \end{cases}$$

4. Составьте программу, заменяющую меньшее из двух данных чисел суммой, а большее – произведением этих чисел.

5. Составить блок-схему алгоритма вычисления площади треугольника 3-мя различными способами на выбор пользователя.

**Задание 3:** Составить циклические алгоритмы, используя предложенные задания.

**Рассмотрите следующие упражнения**

1. Составьте алгоритм вычисления суммы первых 20 членов последовательности с общим членом  $a = S + (k + 1) / (2k + 1)^2$ , используя циклы с предусловием и постусловием. Для циклического накапливания сумм при составлении соответствующих алгоритмов используется предписание стандартного вида:  $\text{сумма} := \text{сумма} + \text{слагаемое}$ . Если повторять такое предписание требуемое количество раз, изменяя соответствующим образом слагаемое, то и будет получена искомая сумма. Понятно, что сумма перед началом работы цикла должна иметь нулевое значение. В схеме, изображенной ниже, роль суммы выполняет переменная S, а роль слагаемого – формула общего члена последовательности  $(k + 1)(2k + 1)$ . Изменение слагаемого достигается увеличением в каждом обороте цикла номера члена k на единицу. Словесная запись этого алгоритма:

Цикл с постусловием:

1. начало
2.  $k := 1; S := 0$
3.  $S := S + (k + 1) / (2k + 1)^2$
4.  $k := k + 1$
5. если  $k \leq 20$  идти к п.3



6. вывод S

7. Конец

Цикл с предусловием:

1. начало

2.  $k := 1; S := 0$

3. если  $k > 20$  идти к п.6

4.  $S := S + (k + 1) / (2k + 1)^2$

5.  $k := k + 1$ ; идти к п.3

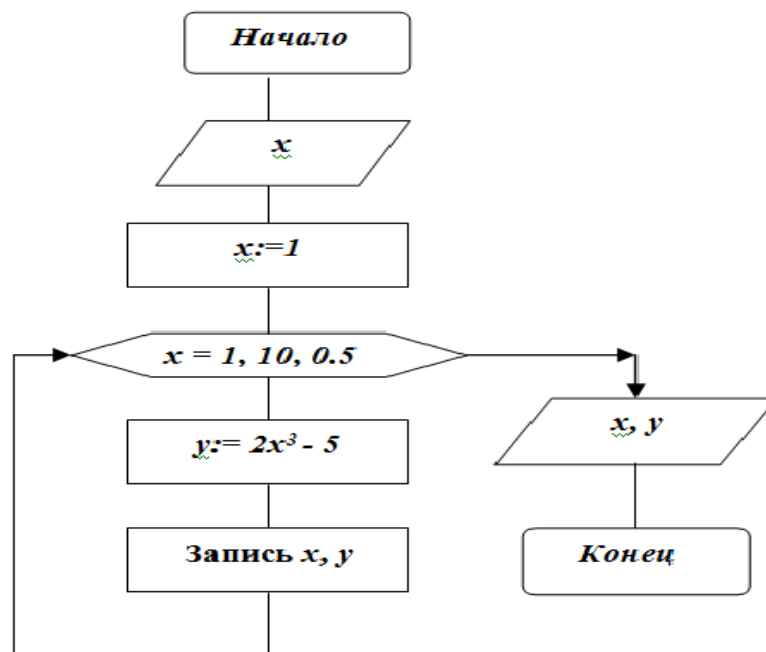
6. вывод S

7. Конец

2. Составьте блок-схему алгоритма для вычисления значений функции для значений  $x$ , начиная с  $x=1$  до  $x=10$  с шагом  $h=0,5$ .

Исходные данные:  $x$ .

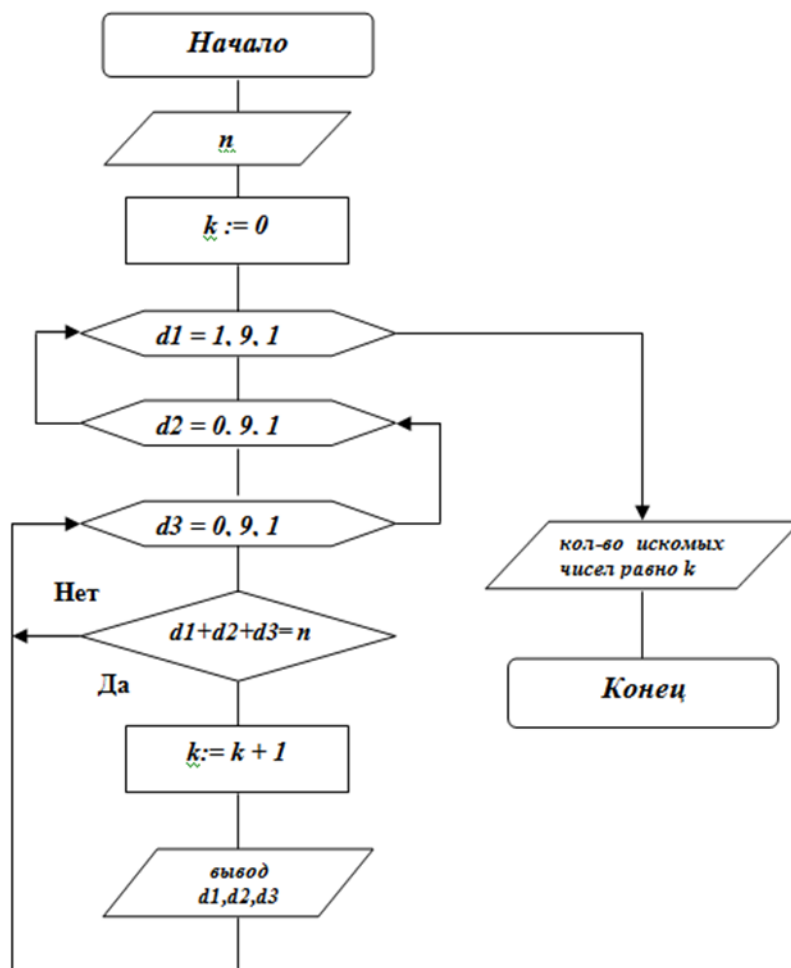
Результат:  $x, y$ .



3. Составьте алгоритм для определения  $k$  – количества трехзначных натуральных чисел, сумма цифр которых равна  $n$  ( $1 < n < 27$ ). Операции деления не использовать. В алгоритме приняты обозначения:  $d1$  - левая,  $d2$  - средняя,  $d3$  – правая цифры числа.

Исходные данные:  $n$ .

Результат:  $k, d1, d2, d3$ .



### Выполнить самостоятельно

Составьте алгоритмы решения следующих задач:

1. Составьте блок-схемы алгоритмов для задачи из Упражнения 1.
2. Составьте блок-схему алгоритма для решения следующей задачи: начав тренировки, спортсмен пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
3. Изобразите блок-схему алгоритма определения максимального числа в последовательности из  $n$  произвольных чисел.
4. Составьте блок-схему алгоритма вычисления суммы квадратов первых  $n$  чисел натурального ряда.
5. Найти сумму первых  $n$  членов ряда

$$y = 1 + \frac{x}{2} + \frac{2 * x}{3} + \frac{3 * x}{4} + \dots \text{при } |x| < 1$$

6. Найти все делители целого положительного числа (кроме 1 и самого числа).

## **Инсталляция и конфигурирование компилятора Pascal. Составление и оформление программы на языке программирования Pascal . Тестирование и отладка программы**

Цель работы: Научиться инсталляции и конфигурированию компилятора Pascal.

Закрепить на практике теоретические знания по созданию программ с линейными алгоритмами, закрепить на практике навыки работы в среде Turbo Pascal:

- 1.научиться записывать на языке Pascal математические выражения
- 2.научиться использовать стандартные функции языка Паскаль
- 3.научиться работать с целыми типами данных
- 4.научиться работать с вещественными типами данных
- 5.научиться работать с операторами ввода;
- 6.научиться работать с операторами вывода;
- 7.форматировать выходную информацию;
- 8.ознакомиться с особенностями текстового режима

### **Задание 1**

Для инсталляции и конфигурирования компилятора Pascal скачать архив. Он содержит все необходимое для запуска Borland Pascal под ОС Windows (32- и 64-разрядными версиями), в том числе русификатор (переход на русский язык и обратно с помощью клавиши Ctrl). Настоятельно не рекомендуется брать Borland или Turbo Pascal из других источников, так как по умолчанию Borland и Turbo Pascal игнорирует многие ошибки в программах, а скомпилированные программы могут не работать на достаточно быстрых компьютерах.

- Если при раскрытии на полный экран (Alt+Enter) искажаются цвета и окно ВР из синего становится красным, то следует в конец файла dos\dosbox.conf добавить строчки:  
[sdl]  
output = opengl
- Если наблюдаются проблемы с работой арифметики с плавающей запятой, то проверьте, что в настройках Options -> Compiler снята галочка с пункта 8087/80287, в разделе Numeric Processing.

Задачи работы 2: Написать в среде Turbo Pascal программы, реализующие следующие линейные алгоритмы, запустить их на выполнение.

$b = \left(1 + \operatorname{tg}^2 \frac{z}{2}\right)$	$b = 1 + \frac{z^2}{3 + z^2 / 5}$
$a = \frac{1 + \sin^2(x + y)}{2 + \left x - 2x / (1 + x^2 y^2)\right } + x$	$a = \ln \left  \left(y - \sqrt{ x }\right) \left(x - \frac{y}{z + x^2 / 4}\right) \right $
$b = \cos^2 \left( \operatorname{arctg} \frac{1}{z} \right)$	$f = 6,673 \cdot 10^{-8} \cdot \frac{m_1 \cdot m_2}{r^2}$
$t = \frac{xyz - 3,3 \left  x + \sqrt[4]{y} \right }{10^7 + \sqrt{\lg 4}}$	$b = e^{ x-y } \ln(1+e) \log_2 \operatorname{tg} 2$
$y = \frac{\beta + \sin^2 \pi^4}{\cos 2 +  \operatorname{ctg} \gamma }$	$y = \sqrt[8]{x^8} + 8^x$

Задачи работы 3 1 Написать в среде Turbo Pascal программы, реализующие следующие линейные алгоритмы, запустить их на выполнение

1. Написать программу, которая для заданного целого числа  $a$  печатает следующую таблицу:

$$\begin{array}{cc}
 a & \\
 a^3 & a^6 \\
 a^6 & a^3 & a
 \end{array}$$

2. Вычислить длину окружности, площади круга и объема шара одного и того же заданного радиуса.

3. Написать программу вычисления периметра и площади прямоугольного треугольника по длинам двух катетов.

4. Написать программу, которая по координатам трех вершин некоторого треугольника вычисляет его площадь и периметр.

5. Найти площадь кольца, внутренний радиус которого равен 20, а внешний – заданному числу  $r$  ( $r > 20$ ).

**Порядок выполнения**

1. Составить в тетради алгоритм
2. Написать программу
3. Отладить программу
4. Протестировать программу
5. Записать в тетради текст программы
6. Записать в тетради входные и выходные данные

## Практическая работа №2

Составление программ разветвляющейся структуры. Применение полученных знания в области математики.

Цель работы:

- 1.научиться работать с оператором условия IF THEN ELSE;
- 2.научиться работать с оператором выбора CASE

Задание: Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Написать программу вычисления Y:

$$y = \begin{cases} x & , x \geq 1 \text{ или } x < -2 \\ 2x + 16 - x & , \text{остальных случаях} \end{cases}$$

2. Написать программу вычисления X:

$$x = \begin{cases} 14 & m = d, \quad l < k \\ 15 & m = d, \quad l \geq k \\ 16 & m \neq d \end{cases}$$

3. Составить программу для подсчета числа букв А, Б, В - в предложении. Предложение вводится отдельными символами до тех пор пока не введен символ (.).

4. В строке введенных символов подсчитать количество символов С, D, Р, считая концом ввода символ (\*).

5. По введенному номеру месяца напечатать наименование времени года и наименование месяца.

6. Улица Байкальская разбита на участки:

с 1 по 100 дом - 1 участок  
с 101 по 203 дом - 2 участок  
с 204 по 308 дом - 3 участок  
остальные дома - 4 участок.

По введенному номеру дома выдать № участка.

7. Лежит ли введенное значение X на отрезке в интервале [4,5.4].

8. В пятиэтажном доме на каждом этаже расположено по четыре квартиры. Составить программу, которая по номеру этажа, печатает номера квартир на этом этаже.

9. Написать программу нахождения наибольшего числа из введенных чисел, считая концом ввода 0.

## Практическая работа №3

### Составление программ циклической структуры

Цель работы: Научиться работать с оператором цикла FOR. Научиться работать с оператором цикла WHILE (цикл с предусловием). Научиться работать с оператором цикла Repeat .... Until(цикл с постусловием).

**Задание 1:** Программирование задач с оператором цикла FOR. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Составить программу определения разрядности введенного целого числа.
2. Подсчитать сумму отрицательных чисел последовательности.
3. Составить программу подсчитывающую сумму цифр вводимого натурального числа.
4. Имеется одномерный массив из 15 чисел ( $a_1, a_2, a_3 \dots a_{15}$ ). Составить программу их ввода. Упорядочить массив по убыванию.
5. Ввести 5-тизначное число. Вывести последовательность цифр начиная с конца.

Пример: ввод - 42891  
вывод - 1 9 8 2 4

**Задание 2:** Программирование задач с оператором цикла WHILE. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение.

1. Подсчитать  $ax^2 + bx + c$ , при  $a=18.5, b=-0.5, c=134$ ,  $x$  изменяется от -1 до 0.6 с шагом 0.2. Значения выдавать на каждом шаге итерации.
2. Подсчитать сумму положительных чисел последовательности.
3. Подсчитать  $w=(a+bx)\sqrt{x+1}$ , при  $a=2.8, b=-0.3$ ,  $x$  изменяется от 1 до 3 с шагом 0.5. Выдавать значения на каждом шаге итерации
4. Имеется одномерный массив из 20 чисел ( $B_1, B_2, B_3 \dots B_{20}$ ). Составить программу их ввода. Упорядочить массив по возрастанию.
5. Подсчитать

$$zz = \begin{cases} fv / (f - 6)x & x \geq 13.6 \\ (f + v)xx / v + 3 & x < 4.8 \\ x / vf & \text{в остальных случаях} \end{cases}$$

Цель работы: Научиться работать с оператором цикла Repeat .... Until(цикл с постусловием).



**Задание.3** Программирование задач с оператором цикла RepeatНаписать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

Перевернуть число 12345 54321

```
var
c,b,x : integer;
begin
writeln('Введите число');
readln(x);
if x < 9 then
writeln('Число должно быть больше 9')
else
repeat
c:=x div 10;
b:=x mod 10;
x:=c;
write(b);
until c < 9;
writeln(c);
readln;
end.
```

1.Подсчитать  $ax^2+bx+c$  , при  $a=18.5$ ,  $b=-0.5$ , $c=134$  ,  $x$  изменяется от -1 до 0.6 с шагом 0.2. Значения выдавать на каждом шаге итерации.

2. Подсчитать сумму положительных чисел последовательности.

3. Подсчитать  $w=(a+bx)\sqrt{x+1}$ , при  $a=2.8$ ,  $b=-0.3$ ,  $x$  изменяется от 1 до 3 с шагом 0.5. Выдавать значения на каждом шаге итерации

4. Имеется одномерный массив из 20 чисел ( $B_1, B_2, B_3 \dots B_{20}$ ). Составить программу их ввода. Упорядочить массив по возрастанию.

5. Подсчитать

$$zz = \begin{cases} fv / (f - 6)x & x \geq 13.6 \\ (f + v)xx / v + 3 & x < 4.8 \\ x / vf & \text{в остальных случаях} \end{cases}$$

## Практическая работа №4

### Обработка одномерных массивов

**Цель работы:** Научиться писать программы с использованием одномерных и двумерных массивов.

- научиться описывать одномерные массивы;
- научиться вводить одномерные массивы;
- научиться генерировать одномерные массивы;
- научиться выводить одномерные массивы;
- осуществлять поиск минимального и максимального элементов массива;
- упорядочивать массивы по возрастанию и убыванию.

**Задание** Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Вычисляя значения переменной  $x=8d+f$  при всех значениях  $d=1,2,3$  и  $f=-3, 3, -6$ . Создать одномерный массив. Вывести значения элементов этого массива и значения  $d, f$ .

2. Создать одномерный массив из случайно сгенерированных чисел (от 1 до 36). Подсчитать сумму нечетных чисел элементов массива. Выдать на экран массив и найденную сумму.

3. Из случайно сгенерированных чисел (от 1 до 25) создать одномерный массив. Выдать массив на экран в виде:

a1			
	a2		
		a3	
			a4

4. Вычисляя значения переменной  $Z=3A/(9+X)$  при всех значениях  $A=0,12,20$  и  $X=-2,-3,-8$ . Создать одномерный массив. Вывести значения элементов этого массива на экран:

$z(a1,x1)$	$z(a1,x2)$	$z(a1,x3)$
$z(a2,x1)$	$z(a2,x2)$	$z(a2,x3)$
$z(a3,x1)$	$z(a3,x2)$	$z(a3,x3)$

5. Ввести данные для формирования одномерного массива  $X[10]$ . Поменять местами элементы массива, введя их номера с клавиатуры.

## Практическая работа №5

### Обработка двумерных массивов

**Цель работы:** Научиться писать программы с использованием одномерных и двумерных массивов.

- научиться описывать двухмерные массивы;
- научиться вводить двухмерные массивы;
- научиться генерировать двухмерные массивы;
- научиться выводить двухмерные массивы;
- осуществлять поиск минимального и максимального элементов массива;
- упорядочивать массивы по возрастанию и убыванию.

**Задание** Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Транспонировать матрицу и подсчитать количество равных элементов

```
program pr1;
const N1=10;N2=12;
var A,A1: array[1..N1,1..N1] of integer;
    B,B1: array[1..N2,1..N2] of integer;
    i,j,k,k1: integer;

begin

randomize;

for I:=1 TO N1 Do begin
for J:=1 TO N1 Do begin
  A[I,J]:=random(10);
  A1[J,i]:=A[i,j];
  end;
end;
for I:=1 TO N2 Do begin
for J:=1 TO N2 Do begin
  B[I,J]:=random(10);
  B1[j,i]:=B[i,j];
  end;
end;
k:=0;
for I:=1 to N2 do begin
for J:=1 to N2 do begin
if B1[i,j]=B[i,j] then inc(k);
end; end;
k1:=0;
for I:=1 to N1 do begin
for J:=1 to N1 do begin
if A1[i,j]=A[i,j] then inc(k1);
end; end;write (k:4,k1:4);
readln;
```

2. Создать трехмерный массив из случайно сгенерированных вещественных чисел, заменяя те элементы массива на 0, которые  $> 55.0$ .
3. Ввести вещественные числа для формирования массива  $A[3,3]$ , подсчитать сумму диагональных элементов этого массива.
4. Ввести вещественные числа для формирования массива  $A[4,4]$ . Переставить колонки этого двумерного массива: 1 на место 2, 2 на место 3, 3 на место 4, 4 на место 1.
5. Ввести данные для формирования двумерного массива  $L[3,3]$ , формируя вначале строки. Поменять местами строки и колонки: 1 строку перенести в 1 колонку, 2 строку перенести во 2 колонку, а 3 строку поместить в 3 колонку.
6. Сгенерировать двумерный массив. Вычислить сумму элементов обратной диагонали.

## Практическая работа №6

Использование стандартных функций и процедур для работы со строками

### Цель работы:

Научиться писать программы по программированию строковых массивов

**Задание** Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

1. Заменить в предложении одно слово на другое

```
program slovo;
var t,a,b,k,s,c:string;
i,j:integer;
begin
write('Введите исходный текст ');
readln(t);
write('Введите слово которое нужно заменить ');
readln(a);
write('Введите слово для замены ');
readln(b);
t:=t+' ';
i:=1;
s:=' ';
while i<>0 do
begin
i:=pos(' ',t);
c:=copy(t,1,i-1);
if c=a then s:=s+b+' ' else s:=s+c+' ';
t:=copy(t,i+1,length(t)-i);
end;
writeln('Новый текст',s:10);
readln;
end.
```

2. Определить, сколько раз в тексте встречается заданная буква.

3. Подсчитать, сколько слов в тексте начинается на заданную букву

## Практическая работа №7

Организация процедур. Использование процедур. Организация функций.  
Использование функций.

Цель работы: Научиться работать с использованием стандартных процедур.  
Задание1. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение

$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$	$z = \frac{ x  -  y }{1 +  xy }$
$a = \frac{3 + e^{y-1}}{1 + x^2  y - \operatorname{tg} z }$	$a = (1 + y) \frac{x + y / (x^2 + 4)}{e^{-x-2} + 1 / (x^2 + 4)}$
$b = 1 +  y - x  + \frac{(y - x)^2}{2} + \frac{ y - x ^3}{3}$	$b = \frac{1 + \cos(y - 2)}{x^4 / 2 + \sin^2 z}$
$a = y + \frac{x}{y^2 + \left  \frac{x^2}{y + x^3 / 3} \right }$	$a = \frac{2 \cos(x - \pi / 6)}{1 / 2 + \sin^2 y}$
$b = \left( 1 + \operatorname{tg}^2 \frac{z}{2} \right)$	$b = 1 + \frac{z^2}{3 + z^2 / 5}$
$a = \frac{1 + \sin^2(x + y)}{2 +  x - 2x / (1 + x^2 y^2) } + x$	$a = \ln \left  \left( y - \sqrt{ x } \right) \right  \left( x - \frac{y}{z + x^2 / 4} \right)$

Задание2. Написать в среде Turbo Pascal программы, реализующие следующие алгоритмы, запустить их на выполнение:

1. Паскаль сам по себе не предоставляет функции, вычисляющей площадь круга по фактическому параметру – диаметру круга. Однако такую функцию легко определить самостоятельно. После этого функцию, например, circle можно использовать в программе точно так же, как ранее использовались функции sqr или trunc.

В первой строке определения функции параметр d как бы говорит «Делай то, что делается со мной, но используй значение, которое будет на моем месте». При вызове функции на место d ставится число, которое таким образом возводится в квадрат, затем результат умножается на 3.14159 и делится на 4.0. Параметр d – это формальный параметр, тогда как конкретное число – фактический параметр.

В основной программе, обращающейся к функции `circle`, можно использовать имя `d` как имя переменной (или любого другого объекта), не опасаясь помех со стороны функции.

Также приведены функции, вычисляющие площади прямоугольника и треугольника; их параметрами являются длины сторон.

Все эти функции могут быть вызваны из программы в любой последовательности и любое количество раз.

Обратите внимание на то, что функции вызываются с фактическими параметрами `x`, `y`, `z`, тогда как формальные параметры в их определениях – `a`, `b`, `c`, `d`. Переменная `a` в основной программе никак не связана с формальным параметром `a` в функциях. Точно так же отсутствует связь переменной `x` в основной программе и локальной переменной `x` в функции `triangle`.

Определенные здесь функции различаются по числу параметров. В функциях может быть любое фиксированное число параметров; определить функцию с переменным числом параметров (например, как в случае `read(a)`, `read(a, b)`, `read(a, b, c)`) нельзя – такой возможностью пользуется только сам Pascal.

Программа на языке Паскаль:

```
var
  choice: char;
  a, x, y, z: real;

function circle(d: real): real;
  const pi = 3.14159;
  begin
    circle := pi * sqr(d) / 4.0;
  end;

function rectangle(a, b: real): real;
  begin
    rectangle := a * b
  end;

function triangle(a, b, c: real): real;
  var x: real;
  begin
    x := (a + b + c) / 2;
    triangle := sqrt(x * (x - a) * (x - b) * (x - c))
  end;

begin
  repeat
    read(choice);
    case choice of
      'c': begin
        readln(x);
        a := circle(x)
      end;
    end;
  end;
```

```

'r': begin
    readln(x, y);
    a := rectangle(x, y)
end;
't': begin
    readln(x, y, z);
    a := triangle(x, y, z)
end
end;
writeln('Square: ', a:8:2);
until choice = 'q';

```

readln

end.

2. Ввести x и вычислить значения всех известных процедур и функций.
3. Определить чётность X через ODD и mod
4. Найти код ASCII заданного символа
5. Найти символ по заданному коду ASCII
6. Написать функцию, которая вычисляет объём цилиндра. Параметрами функции должны быть радиус и высота цилиндра.
7. Написать функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде одного из знаков > , < или =.
8. Написать функцию, которая вычисляет  $ab$  . Числа a и b могут быть дробными положительными числами.
9. Написать функцию Dohod, которая вычисляет доход по вкладу. Исходными данными для функции являются: величина вклада, процентная ставка (годовых) и срок вклада (количество дней)



## Практическая работа №8

### Составление программы на языке Ассемблер

Цель работы: Научиться составлять программу в машинных кодах

#### Задание 1

1. Составить программу в машинных кодах:
  - занести в регистр AX десятичное число  $(134 + \text{№\_машины})$ .
  - прибавить десятичное число  $(103 + \text{№\_машины})$  к AX.
  - переслать содержимое AX в BX.
  - прибавить AX к BX.
  - почистить AX.
  - выход в DOS.
2. Записать программу в машинных кодах в память со смещением  $(100 + Y)$ , где  $Y = \text{№\_машины} + 16$ , Y - десятичное число, 100 - шестнадцатеричное число.
3. Рассмотреть содержимое всех регистров.
4. Рассмотреть записанную программу в памяти.
5. Осуществить пошаговое выполнение созданной программы до команды RET.
6. Вести протокол работы в тетради (краткий).
7. Оформить протокол.

#### Задание 2

1. Составить программу в машинных кодах:
  - переслать слово (первое число) - два байта, начинающиеся в сегменте данных с адреса 04 в регистр AX,
  - прибавить содержимое слова (второе число) - два байта, начинающиеся в сегменте данных с адреса 02 к регистру AX,
  - переслать содержимое регистра AX в слово, начинающиеся в сегменте данных DS с адреса 00,
  - вернуться в DOS.если: первое число =  $-576 + N_{\text{маш}}$ ;  
второе число =  $84 - N_{\text{маш}}$ ;
  - конец сегмента данных заполнить некоторыми символами ? , # , \* , @ по выбору.
2. Рассмотреть содержимое сегмента данных.
3. Рассмотреть записанную программу в памяти.
4. Рассмотреть содержимое всех регистров.
5. Осуществить пошаговое выполнение созданной программы до команды RET.
6. Создать файл с расширением .com.
7. Оформить протокол.

**Перечень учебных изданий, интернет - ресурсов, дополнительной литературы:**  
**Основная литература:**

Голицына О. Л. Основы алгоритмизации и программирования: Учеб. пособие. – М.:ИНФРА-М, 2022. – 432 с.

**Дополнительная литература:**

Алгоритмизация и программирование: Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2022. - 352 с.: ил.; 60х90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0355-1

**Интернет-ресурсы:**

- Программирование на Java, C++, Pascal: <http://www.kufas.ru>
- Основы программирования: [www.infobook.ru/book/book\\_10496\\_0.html](http://www.infobook.ru/book/book_10496_0.html)