



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра «Естественные науки»

ИНФОРМАТИКА

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PASCAL ABC

Учебно-методическое пособие для иностранных слушателей
дополнительных общеобразовательных программ

Автор
Моренко Б.Н.

Ростов-на-Дону, 2021



Аннотация

Учебно-методическое пособие по программированию на языке Pascal ABC является частью учебно-методического комплекса по информатике. Оно предназначено для использования на практических занятиях и для самостоятельной работы иностранных слушателей дополнительных общеобразовательных программ. На минимально достаточном лексико-грамматическом материале и в доступной для понимания форме изложены теория и практика выполнения работ по программированию на языке Pascal ABC. Ввод и отработка фонетики, лексики и грамматических моделей соотнесены с программой обучения русскому языку как иностранному.

Учебно-методическое пособие рекомендуется для использования на практических занятиях по информатике, а также для самостоятельной работы при подготовке к текущему контролю и итоговой аттестации.

Автор:

Моренко Б.Н. – к.т.н., доцент кафедры
«Естественные науки»



Оглавление

	стр.
Аннотация	2
Введение	4
Работа в среде программирования Pascal ABC	5
1. Линейные программы	7
2. Разветвляющиеся программы	31
3. Оператор выбора	44
4. Циклические программы	54
5. Массивы	73
Литература и интернет-источники	81



ВВЕДЕНИЕ

Учебно-методическое пособие по программированию на языке Pascal ABC является частью учебно-методического комплекса по информатике. Учебно-методическое пособие предназначено для использования на практических занятиях по информатике и для самостоятельной работы иностранных слушателей дополнительных общеобразовательных программ инженерно-технической, технологической, экономической и естественнонаучной направленностей обучения. Основной целью Учебно-методического пособия является изложение в доступной для понимания форме технологии выполнения лабораторно-практических работ по информатике на минимально достаточном лексико-грамматическом материале. Введение и накопление необходимого объёма лексики и грамматических конструкций способствует успешному формированию коммуникативной компетенции иностранных слушателей. Использование лексики соотнесено с программой обучения русскому языку как иностранному.

Особое место на практических занятиях по информатике занимают профильно-ориентированные задания, которые носят деятельностный характер, моделируют практическую ситуацию и строятся на актуальном для иностранных слушателей учебном материале. Эти задания способствуют формированию практических навыков, развивают способность иностранных слушателей действовать, находить и применять на практике полученные знания, умения и навыки при решении учебных и прикладных задач по физике, химии, математике и другим дисциплинам.



РАБОТА В СРЕДЕ ПРОГРАММИРОВАНИЯ PASCAL ABC

1. Включите компьютер. Дождитесь полной загрузки операционной системы Microsoft Windows.
2. Щёлкните два раза левой клавишей мыши на ярлыке «Pascal ABC». Ярлык находится на Рабочем столе компьютера. Откроется окно программирования Pascal ABC.
3. В окне программирования Pascal ABC напишите текст программы. Для переключения между русским и латинским языками используйте клавиши **Shift** и **Alt**.
4. После набора текста программы её следует сохранить в файле. Для этого:
 - нажмите на клавишу F2 или выберите в главном меню пункт File, а затем Save;
 - в появившемся диалоговом окне «Save file as...» («Сохраните файл как...») напишите имя файла, в котором будет сохранена ваша программа; имя файла нужно писать в формате «*****_01.pas», где пять символов «*****» – это первые пять букв вашего имени, записанного латинскими буквами;
 - нажмите клавишу «Enter» или кнопку «Сохранить» в диалоговом окне «Save file as...».
5. Проверьте текст программы на наличие ошибок. Для этого нажмите клавишу F9. Компьютер покажет ваши ошибки. Исправьте обнаруженные компьютером ошибки.
6. Для запуска программы на выполнение расчётов используйте комбинацию клавиш F9.
7. Выполните предложенные преподавателем задания, ответьте на поставленные вопросы.
8. Порядок окончания работы:
 - сохраните все изменения в программе. Для этого в меню «Файл» выберите «Сохранить»;
 - выйдите из среды программирования Pascal ABC. Для этого в меню «Файл» выберите «Выход»;



11. После окончания работы выключите компьютер. Для этого:
- закройте все открытые программы;
 - отключите и вытащите из USB-разъёма компьютера Flash-накопитель;
 - выполните последовательно следующие действия:
Пуск → Выключение → Ок;
- Внимание!** Если вы делаете небольшой перерыв в работе, то лучше оставить компьютер включённым!
- выключите питание монитора и всех периферийных устройств;
- Внимание!** Компьютер нельзя выключать путём его отключения от электрической сети! Для аварийного (быстрого) выключения компьютера или невозможности его корректного выключения (см. пункт 11) нужно нажать клавишу «Power» и удерживать её в нажатом состоянии 6-8 секунд.

1. ЛИНЕЙНЫЕ ПРОГРАММЫ

Перед выполнением лабораторно-практических работ повторите тему «Алгоритмы», а также изучите теоретический материал. Для этого используйте конспект теоретической части занятия и рекомендуемую преподавателем литературу. При самостоятельном изучении материала обратите внимание на следующие понятия: алгоритм, свойства и способы описания алгоритмов, типы алгоритмов, графическое описание и блок-схемы алгоритмов, программа, линейная программа, программирование.

Запомните: алгоритм – это сформулированная на некотором языке последовательность действий и правил для решения задачи.

Алгоритм - это сформулированная на некотором языке последовательность действий и правил решения задачи.

Алгоритмы делятся на три вида: линейные (следование), разветвляющиеся (ветвления) и циклические.



Программы с линейной структурой (линейные программы) являются простейшими и используются для простых вычислений по формулам. При выполнении линейных алгоритмов операторы выполняются последовательно друг за другом в том порядке, в котором они перечислены в программе.

Перед выполнением лабораторно-практических работ повторите теоретический материал по работе с языком программирования Pascal ABC. Для этого используйте конспект занятия и методические указания, которые находятся в электронной библиотеке Управления цифровых образовательных технологий [электронных ресурсов университета](#) по

адресу (<http://skif.donstu.ru/>) и рекомендуемую литературу.



Блок-схема линейного алгоритма показана на рисунке. Блок-схема – это графическое изображение структуры алгоритма в виде отдельных геометрических фигур или блоков, соединённых между собой линиями связи в порядке выполнения операций. Форма блока (прямоугольник, параллелограмм, ромб и т.д.) определяет вид действий.

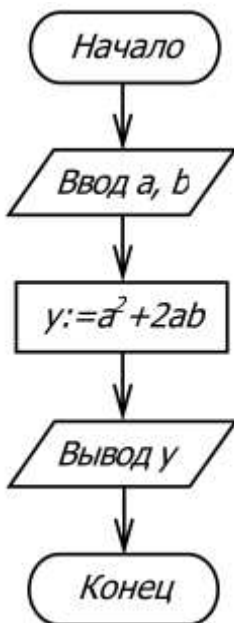
Обратите внимание на понятие «программа». Программа – это последовательность команд, которые определяют операции и порядок их выполнения при решении задачи. Процесс составления алгоритма решения задачи и программы на алгоритмическом языке называется программированием.

Задача 1.1. Составить блок-схему алгоритма и программу расчёта функции Y :

$$y = a^2 + 2ab$$

Цель работы. Приобрести навыки построения блок-схем алгоритмов, разработки программ и расчёта на персональном компьютере функции Y при некоторых значениях a и b .

Блок-схема алгоритма расчёта функции Y :



Программа расчёта функции Y :

```

program Line_Programme_1;
var
    a, b, y: integer;
begin
    write ('Введите число a, a=');
    readln (a);
    write ('Введите число b, b=');
    readln (b);
    y:= a*a + 2*a*b;
    writeln ('Значение функции y=', y);
end.
    
```

Наберите (напишите) на компьютере программу расчёта функции Y и рассчитайте её значения по заданным a и b .

Результаты расчётов запишите в таблицу.

№№ п/п	Значение a	Значение b	Значение функции Y
1.	25	47	
2.	85	94	
3.	95	113	
4.	110	150	
5.	63,5	53,7	

Дополнительные задания.

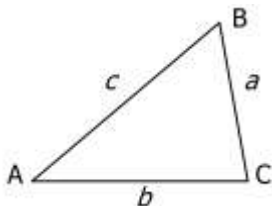
- Внесите изменения в программу, которые позволят вывести на экран монитора заголовок программы в виде: «Практическая работа по информатике. Расчёт функции Y ».
- Внесите изменения в программу, которые обеспечивают вывод на экран монитора результаты расчёта в виде: «При $a=5$ и $b=3$ значение функции $y=55$ ».

Задача 1.2. Составить блок-схему алгоритма и программу расчёта площади треугольника по заданным размерам трёх его сторон.

Цель работы. Приобрести навыки построения блок-схемы, разработки программы и расчёта на персональном компьютере площади треугольника по формуле Герона.

Примечание.

Площадь треугольника S по заданным размерам трёх его сторон a , b , c можно вычислить по формуле Герона:

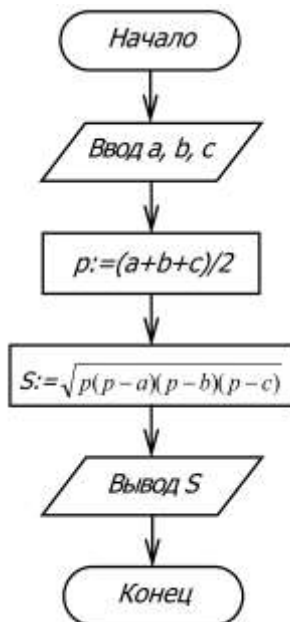


$$S := \sqrt{p(p-a)(p-b)(p-c)},$$

где p – полупериметр треугольника,

$$p = \frac{a+b+c}{2}$$

Блок-схема алгоритма решения задачи:



Программа расчёта площади треугольника по формуле Герона:

```

program Line_Programme_2;
var
  a, b, c, p, S: real;
begin
  write ('Введите размер стороны a, a=');
  readln (a);
  write ('Введите размер стороны b, b=');
  readln (b);
  write ('Введите размер стороны c, c=');
  readln (c);
  p:= (a+b+c)/2;           {Полупериметр треугольника}
  S:= sqrt(p*(p-a)*(p-b)*(p-c)); {Площадь треугольника}
  write ('Площадь треугольника S=', S:6:2);
end.
    
```



Рассчитайте площадь S треугольника по заданным размерам его сторон a , b , c . Результаты расчётов запишите в таблицу.

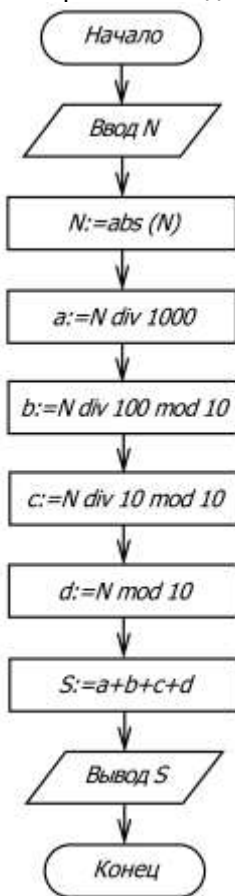
№№ п/п	Размер a	Размер b	Размер c	Площадь треугольника, S
1.	5	9	12	
2.	23,5	46,3	53,8	
3.	45,32	23,46	50,295	
4.	194	395	620	
5.	435	138	573	

Дополнительные задания.

1. Объясните результаты расчётов, полученные в п.п. 4 и 5.
2. Какие изменения нужно внести в программу, чтобы после численного значения площади треугольника указывалась единица её измерения, например, «145 квадратных единиц»?
3. Какие изменения нужно внести в программу для увеличения точности результатов расчёта до 0,00001? Уменьшения точности результатов расчётов до 0,1?

Задача 1.3. Составить блок-схему алгоритма и программу расчёта суммы цифр четырёхзначного числа.

Блок-схема алгоритма решения задачи:



Программа расчёта суммы цифр четырёхзначного числа:

```

program Line_Programme_3;
var
    N: integer;
    a, b, c, d: integer;
    S: integer;
    
```

begin

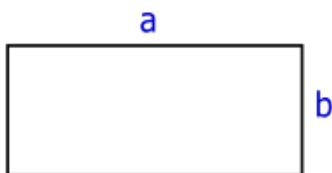
```
write ('Введите четырёхзначное число => ');  
readln (N);  
N:= Abs(N);  
    a:= N div 1000;  
    b:= N div 100 mod 10;  
    c:= N div 10 mod 10;  
    d:= N mod 10;  
writeln ('Цифры числа:');  
writeln ('Первая цифра - ', a);  
writeln ('Вторая цифра - ', b);  
writeln ('Третья цифра - ', c);  
writeln ('Четвертая цифра - ', d);  
    S:=a+b+c+d;  
writeln ('Сумма цифр числа ', N, ' равна ', S);  
end.
```

Дополнительные задания.

1. Какие изменения нужно внести в программу для расчёта произведения всех цифр заданного четырёхзначного числа?
2. Какие изменения нужно внести в программу для записи цифр заданного четырёхзначного числа в обратном порядке?
3. Составить программу расчёта суммы и произведения цифр трёхзначного числа.
4. Составить программу расчёта суммы и произведения цифр двузначного числа.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задача 1. Составить блок-схему алгоритма и написать программу расчёта площади и периметра прямоугольника по заданным размерам его сторон a и b .



Формула площади прямоугольника:

$$S=ab$$

Формула периметра прямоугольника:

$$P=2(a+b)$$

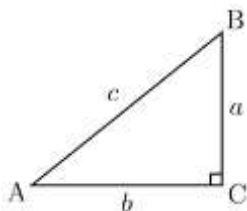
где a и b – стороны прямоугольника.

Составьте (начертите в тетради) блок-схему алгоритма и напишите на языке Pascal ABC программу расчёта площади S и периметра P прямоугольника по заданным размерам его сторон a и b .

Результаты расчётов запишите в таблицу.

№№ п/п	Размер стороны a	Размер стороны b	Площадь прямоугольника	Периметр прямоугольника
1.	6	3		
2.	50	20		
3.	10,5	4,25		
4.	400	300		
5.	8000	5000		

Задача 2. Составить блок-схему алгоритма и написать программу расчёта гипотенузы c прямоугольного треугольника по заданным размерам его катетов a и b .



Определить размер гипотенузы c прямоугольного треугольника можно по формуле Пифагора:

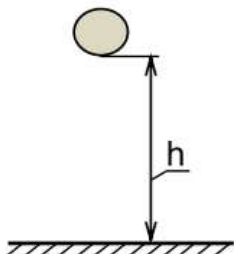
$$c = \sqrt{a^2 + b^2}$$

где a и b – размеры катетов треугольника.

Составьте (начертите в тетради) блок-схему алгоритма и напишите на языке Pascal ABC программу расчёта размеров гипотенузы c прямоугольного треугольника по заданным размерам его катетов a и b . Результаты расчётов запишите в таблицу.

№№ п/п	Размер катета a , см	Размер катета b , см	Размер гипотенузы c , см
1.	3	4	
2.	60	90	
3.	15,4	29,25	
4.	300	300	
5.	25000	40000	

Задача 3. Составить блок-схему алгоритма, написать программу и рассчитать время свободного падения тела с высоты h .



Определить время свободного падения тела с высоты h можно по формуле:

$$t = \sqrt{\frac{2h}{g}}$$

где t – время падения тела, сек,
 h – высота падения тела, м,
 g – ускорение свободного падения, мс^{-2} .

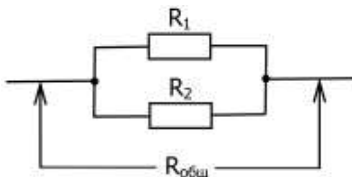
Составьте (начертите в тетради) блок-схему алгоритма и напишите на языке Pascal ABC программу расчёта времени t свободного падения тела с высоты h . Вычислите значения времени падения тела по заданным значениям высоты h .

Результаты расчётов запишите в таблицу.

№№ п/п	Высота падения h , метров	Время падения t , секунд
1.	100	
2.	324,8	
3.	1500	
4.	3500	
5.	20000	

Задача 4. Составить блок-схему алгоритма и написать программу расчёта общего электрического сопротивления двух параллельно соединённых резисторов R_1 и R_2 .

Примечание. Общее электрическое сопротивление двух параллельно соединённых резисторов можно определить по формуле:



$$R_{\text{общ}} = \frac{R_1 R_2}{R_1 + R_2}$$

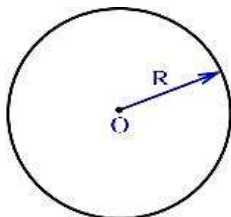
где $R_{\text{общ}}$ – общее электрическое сопротивление резисторов.

Составьте (начертите в тетради) блок-схему алгоритма и напишите на языке Pascal ABC программу расчёта общего сопротивления двух параллельно соединённых резисторов R_1 и R_2 . Рассчитайте общее сопротивление резисторов R_1 и R_2 и результаты расчётов запишите в таблицу.

№№ п/п	Сопротивление резистора R_1	Сопротивление резистора R_2	Общее сопротивление, ом
1.	10 Ом	20 Ом	
2.	100 Ом	5 Ом	
3.	25,2 Ом	50,5 Ом	
4.	2,5 кОм	1 кОм	
5.	1 кОм	400 Ω	

3

задача 5. Составить блок-схему алгоритма и написать программу расчёта диаметра D , длины окружности L и площади круга S , радиус которого равен R .



Круг

Диаметр круга D можно вычислить по формуле:

$$D=2R,$$

где R – радиус окружности.

Длину окружности L можно вычислить по формуле:

$$L=2\pi R$$

Площадь круга S можно вычислить по формуле:

$$S= \pi R^2$$

Составьте (начертите в тетради) блок-схему алгоритма и напишите на языке Pascal ABC программу расчёта диаметра D , длины окружности L и площади S круга, радиус которого равен R .

Вычислите значения D , L и S по заданным значениям радиуса R . Результаты расчётов запишите в таблицу.



№№ п/п	Радиус круга, R	Диаметр круга, D	Площадь круга, S	Длина окружности, L
1.	5			
2.	20			
3.	35,2			
4.	80,05			
5.	250			

РАСЧЕТ СОПРОТИВЛЕНИЯ ЭЛЕКТРИЧЕСКОЙ ЦЕПИ

Цель работы. Приобрести навыки построения блок-схем алгоритмов, разработки программ и расчёта на персональном компьютере сопротивления электрической цепи, состоящей из нескольких резисторов.

При повторении раздела «Электричество» курса физики обратите внимание, что за единицу измерения сопротивления в Международной системе единиц (СИ) принят **Ом** (международное обозначение: Ω – омега).

Сопротивление – это физическая величина, которая характеризует свойства проводника препятствовать прохождению электрического тока.

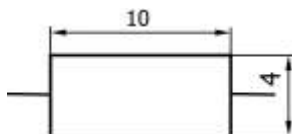
1 Ом равен электрическому сопротивлению проводника, между концами которого возникает напряжение 1 вольт при силе тока 1

На практике используют и другие производные единицы измерения сопротивления, например, килоом (кОм , $\text{k}\Omega$) и мегаом (МОм , $\text{M}\Omega$).

$$1 \text{ килоом (кОм, k}\Omega) = 1\,000 \text{ Ом} = 10^3 \text{ Ом}$$

$$1 \text{ мегаом (МОм, M}\Omega) = 1\,000\,000 \text{ Ом} = 10^6 \text{ Ом}$$

Запомните! Резисторы на электрической схеме изображают в виде прямоугольника размером 10 x 4 мм.



Дополнительные задания

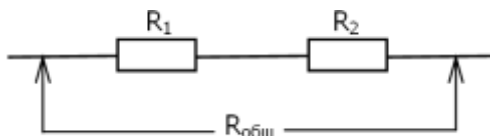
1. Выведите формулу для расчёта сопротивления трёх последовательно соединённых резисторов.
2. Выведите формулу для расчёта сопротивления трёх параллельно соединённых резисторов.

Задача 1. Составить блок-схему алгоритма и написать программу расчёта общего сопротивления электрической цепи, которая состоит из двух последовательно соединённых резисторов (Рис. 2).

Примечание. Общее сопротивление двух последовательно соединённых резисторов определяется по формуле:

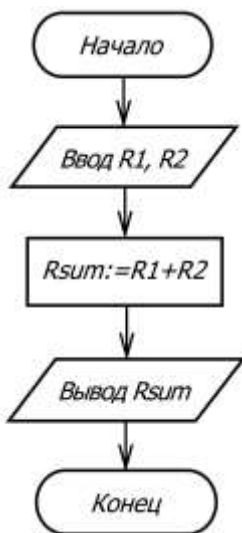
$$R_{\text{общ}} = R_1 + R_2$$

где $R_{\text{общ}}$ – общее сопротивление электрической цепи,
 R_1 и R_2 – сопротивления резисторов.



Последовательное соединение резисторов

Блок-схема алгоритма расчёта сопротивления электрической цепи.



где: R1 – сопротивление резистора R1

R2 – сопротивление резистора R2

Rsum – общее сопротивление электрической цепи

Программа расчёта электрического сопротивления цепи.

```

program Circuit_Resistance_1;
var
  R1, R2, Rsum : integer;
begin
  write (' Введите сопротивление резистора R1, R1=' );
  readln (R1);
  write (' Введите сопротивление резистора R2, R2=' );
  readln (R2);
  Rsum:= R1 + R2;
  writeln (' Общее сопротивление цепи Rsum=', Rsum );
end.
    
```

Напишите на компьютере программу расчёта общего сопротивления электрической цепи и рассчитайте его значение по заданным значениям R_1 и R_2 . Результаты расчётов запишите в Таблицу.

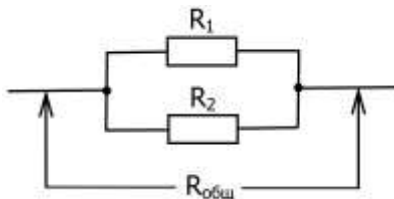
Таблица.

№№ п/п	Значение R_1	Значение R_2	Общее сопротивление электрической цепи, Ом
1.	250 Ом	800 Ом	
2.	10 кОм	1200 Ом	
3.	95 килоом	5 МОм	
4.	110,5 Ω	150,4 Ом	
5.	500 Ом	-500 Ом	

Дополнительные задания.

1. Объясните результаты расчётов в п. 5 таблицы. Где ошибка?
2. Внесите изменения в программу, которые обеспечивают вывод на экран монитора заголовка программы в виде: *«Практическая работа по информатике и физике. Расчёт электрического сопротивления цепи»*.

Задача 2. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления цепи, которая состоит из двух параллельно соединённых резисторов.

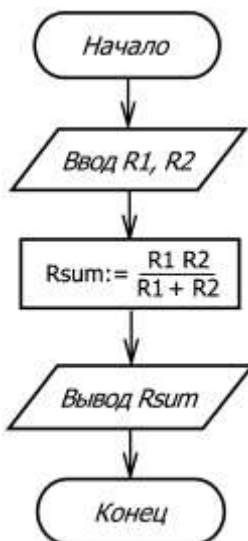


Параллельное соединение резисторов

Примечание. Общее электрическое сопротивление двух параллельно соединённых резисторов определяется по формуле:

$$R_{\text{общ}} = \frac{R_1 R_2}{R_1 + R_2}$$

Блок-схема алгоритма расчёта сопротивления электрической цепи.



Программа расчёта сопротивления электрической цепи.

```

program Circuit_Resistance_2;
var
    R1, R2, Rsum : real;
begin
    write (' Введите сопротивление резистора R1, R1=' );
    readln (R1);
    write (' Введите сопротивление резистора R2, R2=' );
    Rsum := (R1*R2)/(R1 + R2);
    writeln (' Общее сопротивление цепи Rsum=', Rsum);
end.
    
```

Напишите на компьютере текст программы расчёта общего сопротивления электрической цепи и рассчитайте его значение по заданным значениям R_1 и R_2 . Результаты расчётов запишите в Таблицу 2.

Таблица 2.

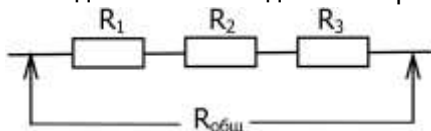
№№ п/п	Значение R_1	Значение R_2	Общее электрическое сопротивление цепи, Ом
1.	200 Ом	200 Ом	
2.	5 кОм	1000 Ом	
3.	400 килоом	1 МОм	
4.	1,5 Ω	50,2 Ом	
5.	-100 Ом	-500 Ом	

Дополнительные задания

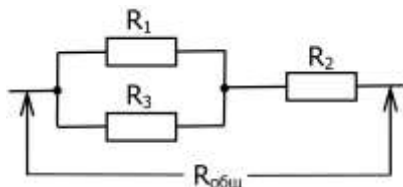
1. Объясните результаты расчётов, полученные в п. 5 таблицы. Где ошибка?
2. Внесите изменения в текст программы, которые обеспечат вывод на экран монитора заголовка программы в виде: *«Лабораторно-практическая работа по информатике и физике. Расчёт электрического сопротивления цепи».*

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

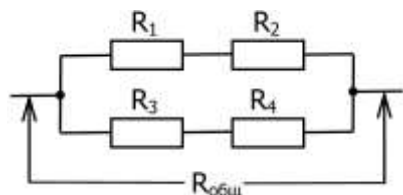
Задача №3. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления цепи, которая состоит из трёх последовательно соединённых резисторов:



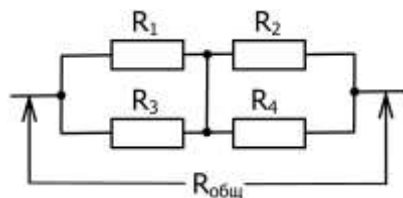
Задача №4. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов:



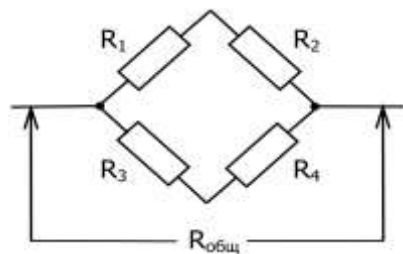
Задача №4. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов.



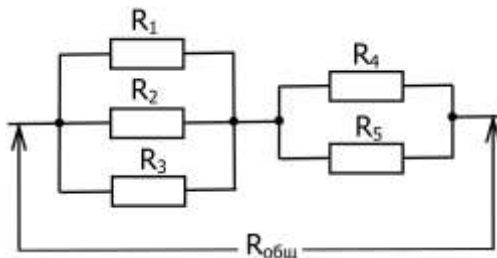
Задача №5. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов.



Задача №6. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов.



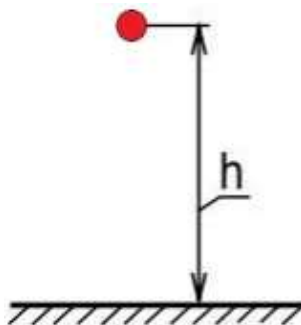
Задача №7. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов:



РАСЧЕТ ВРЕМЕНИ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Цель работы: Приобрести навыки построения блок-схемы алгоритма, разработки текста программы и расчёта на персональном компьютере времени свободного падения тела с высоты h .

Задача 1. Составить блок-схему алгоритма, написать программу и выполнить на компьютере расчёт времени t свободного падения тела с заданной высоты h .



Свободное падение тела

Определить время свободного падения тела с высоты h можно по формуле:

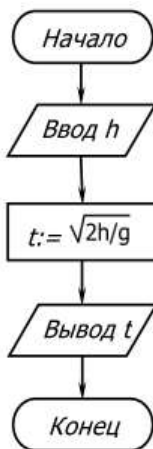
$$t = \sqrt{\frac{2h}{g}}$$

где t – время падения тела, сек,

h – высота падения тела, м,

g – ускорение свободного падения, мс^{-2} .

Блок-схема алгоритма расчёта времени свободного падения тела.



1. Программа расчёта времени свободного падения тела:

```
program Body_fall;
```

```
var
```

```
    h: real;
```

```
begin
```

```
    write (' Введите высоту падения тела h, h=' );
```

```
    readln (h);
```

```
    t:= sqrt (2*h/9.8);
```

```
    writeln (' Время падения тела t=', t, ' секунд');
```

```
end.
```



Вычислите время свободного падения тела по заданным значениям высоты h . Результаты расчётов запишите в таблицу.

№№ п/п	Значение h , метров	Значение t , секунд
1.	100	
2.	324,8	
3.	1500	
4.	35000	

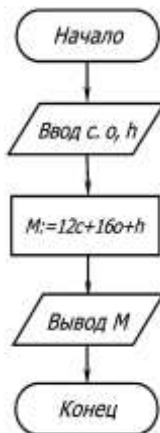
Ответьте на вопросы:

1. Зависит ли время падения тела от его массы?
2. Тело падает с высоты h . Где оно упадёт быстрее: на Земле или на Луне?
3. Чему равно ускорение свободного падения тела вблизи поверхности Земли?
4. Куда направлено ускорение свободного падения тел?
5. От чего зависит ускорение свободного падения?
6. Что называется свободным падением?
7. К какому виду движения относится свободное падение?

ОПРЕДЕЛЕНИЕ МОЛЯРНОЙ МАССЫ ВЕЩЕСТВА

Задача 1. Составить блок-схему алгоритма и написать программу расчёта молярной массы вещества (г/моль), которое состоит из углерода (С), водорода (Н) и кислорода (О).

Блок-схема алгоритма расчёта молярной массы (г/моль) вещества.



Программа расчёта молярной массы (г/моль) вещества.

```

program Molar_Mass;
var
    C, O, H, M: integer;
begin
    write (' Введите количество атомов углерода C, C=' );
    readln (C);
    write (' Введите количество атомов кислорода O, O=' );
    readln (O);
    write (' Введите количество атомов водорода H, H=' );
    readln (H);
    M:=C*12+O*16+H;
    writeln (' Молярная масса вещества равна ', M, ' г/моль');
end.
    
```

Выполните расчёт молярной массы (г/моль) некоторых химических веществ.

Название вещества	Химическая формула	Молярная масса вещества, г/моль
Вода	H_2O	
Метан	CH_4	
Пропан	C_3H_8	
Бутан	C_4H_{10}	
Уксусная кислота	CH_3COOH	
Этиловый спирт	C_2H_5OH	
Метиловый спирт	CH_3OH	
Угольная кислота	H_2CO_3	

Ответьте на вопросы по теме «Язык программирования Pascal ABC»:

- Что такое алфавит языка программирования?
- Что входит в состав алфавита языка Pascal ABC?
- Какие знаки арифметических действий используют в Pascal ABC?
- Что такое линейная программа?
- Что такое разветвляющаяся программа?
- Какие числа называются вещественными?
- Какие существуют правила записи чисел в Pascal ABC?
- Какие существуют правила записи арифметических выражений?
- Какие арифметические функции вы знаете?

2. ПРОГРАММЫ ВЕТВЛЕНИЯ В PASCAL ABC (РАЗВЕТВЛЯЮЩИЕСЯ ПРОГРАММЫ)

Перед выполнением лабораторно-практических работ повторите теоретический материал по языку программирования Pascal ABC. Для этого используйте конспект занятия, методические указания, размещённые в электронной библиотеке университета, и рекомендуемую литературу. Обратите внимание на то, что при решении многих теоретических и прикладных задач порядок вычислений зависит от некоторых условий: исходных данных или результатов промежуточных расчётов.

Условие – это логическое выражение, которое может принимать два значения: «Да» («True») – если условие верно, и «Нет» («False») – если условие неверно. Элемент схемы, который соответствует логическому условию, изображают в виде ромба с одним входом и двумя выходами. Для записи разветвляющихся алгоритмов (ветвлений) в языке Pascal ABC используют соответствующие инструкции, например, *if...then*, *case...of* и др.

Инструкцию *if...then* используют в программах, которые требуют выбора одного из двух действий в зависимости от некоторого условия. В общем случае блок-схема разветвляющегося алгоритма (ветвления) может быть изображена, как показано на рисунке.



Фрагмент блок-схемы разветвляющегося алгоритма
(полная форма)

Полная форма разветвляющегося алгоритма на языке Pascal ABC может быть описана следующим образом:

```

if <Условие>
then <Оператор_1>
else <Оператор_2>;
    
```

Из блок-схемы алгоритма видно, что при выполнении заданного условия (ветвь «Да») выполняется *Оператор_1*, если нет (ветвь «Нет»), – *Оператор_2*. Операторы – это любые арифметические или иные выражения.

Имеются две формы ветвления – полная и неполная, когда отсутствует одна из ветвей.



Фрагмент блок-схемы разветвляющегося алгоритма
(неполная форма)

Форма записи алгоритма на языке Pascal ABC в этом случае имеет вид:

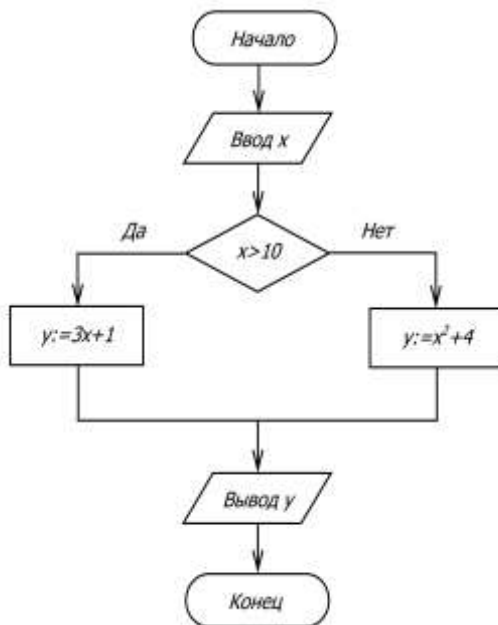
```

if <Условие>
then <Оператор_1>;
    
```

Задача 2.1. Составить блок-схему алгоритма и написать программу расчёта функции Y :

$$y = \begin{cases} 3x+1, & \text{при } x > 10 \\ x^2 + 4 & \text{- в остальных случаях} \end{cases}$$

Блок-схема алгоритма расчёта функции Y:



Программа расчёта функции Y:

```

program Branch_1;
var
    x, y: integer;
begin
    write ('Введите значение x, x=');
    readln (x);
    if x > 10
    then y := 3 * x + 1
    else y := x * x + 4;
    writeln ('При x=', x, ' значение функции y=', y);
    writeln ('Программа завершила работу');
end.
    
```



Рассчитайте значения функции Y по заданным значениям x .
Результаты расчётов запишите в таблицу.

№№ п/п	Значение x	Значение функции Y
1.	8	
2.	17	
3.	25	
4.	-200	
5.	9,5	

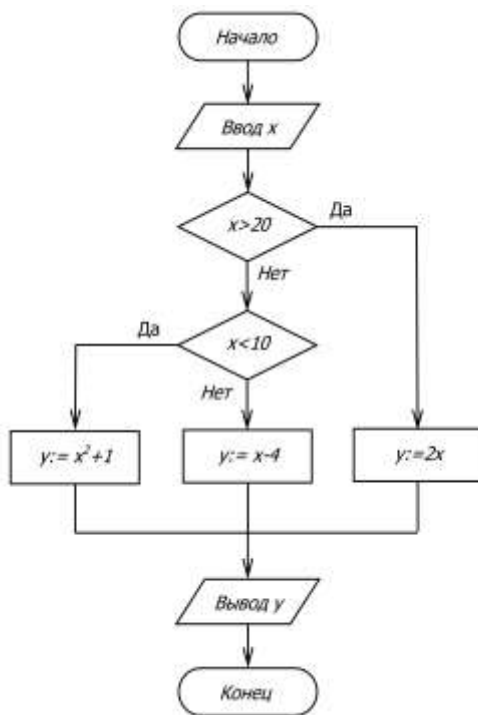
Дополнительные задания

1. Какие изменения нужно внести в программу для работы с вещественными числами?
2. Какие изменения нужно внести в программу для увеличения точности расчётов до 0,001? Уменьшения точности расчётов до 0,1?

Задача 2. Составить блок-схему алгоритма и написать программу расчёта функции Y :

$$y = \begin{cases} 2x, & \text{при } x > 20 \\ x^2 + 1, & \text{при } x < 10 \\ x - 4 & \text{– в остальных случаях} \end{cases}$$

-схема алгоритма расчёта функции Y :



Программа расчёта функции Y :

```

program Branch_2;
var
    x, y: integer;
begin
    write ('Введите значение x, x=');
    readln (x);

```



```
if x>20
  then y:=2*x
  else
    if x<10
      then y:=x*x+1
      else y:=x-4;
writeln ('При x=', x, ' значение функции y=', y);
end.
```

Рассчитайте значения функции Y по заданным значениям x .
Результаты расчётов запишите в таблицу.

№№ п/п	Значение x	Значение функции Y
1.	35	
2.	15	
3.	7	
4.	-300	
5.	-5,2	

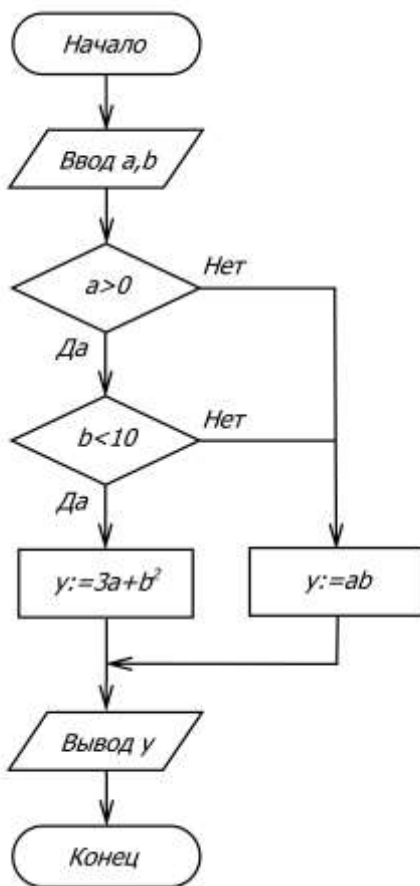
Дополнительные задания:

1. Объясните результаты расчётов, полученные в п. 5. Какие изменения необходимо внести в программу для получения правильных ответов?
2. Какие изменения необходимо внести в программу для увеличения точности расчётов до 0,01? Уменьшения точности расчётов до 0,1?

Задача 3. Составить блок-схему алгоритма и написать программу расчёта функции Y :

$$y = \begin{cases} 3a + b^2, & \text{при } a > 0 \text{ и } b < 10 \\ ab & \text{- в остальных случаях} \end{cases}$$

Блок-схема алгоритма расчёта функции Y :



Программа расчёта функции Y :

```
program Branch_3;  
var  
    a, b, y: integer;  
begin  
    write ('Введите значение a, a=');  
    readln (a);  
    write ('Введите значение b, b=');  
    readln (b);  
    if (a>0) and (b<10)  
        then y:=3*a+b*b  
        else y:=a*b;  
    writeln ('Значение функции y=' , y);  
end.
```

Напишите на компьютере программу расчёта функции Y . Рассчитайте значения функции Y по заданным значениям a и b . Результаты расчётов запишите в таблицу.

№№ п/п	Значение a	Значение b	Значение функции Y
1.	2	6	
2.	8	30	
3.	-5	20	
4.	10	-200	
5.	-1,9	4,7	

Дополнительное задание:

1. Объясните результаты расчётов, полученные в п. 4 и п. 5. Какие изменения необходимо внести в программу для получения правильных ответов?

Задача 4. Заданы три действительных числа: a , b , c . Составить блок-схему алгоритма решения задачи (самостоятельно) и написать программу выбора максимального числа из трёх заданных.

```

program Maximum_Number;
var
    a, b, c, max: integer;      {max – максимальное число}
begin
    write ('Введите число a, a=');
    readln (a);
    write ('Введите число b, b=');
    readln (b);
    write ('Введите число c, c=');
    readln (c);
    if a>b
    then
        max:=a
    else
        max:=b;
        if c>max
        then max:=c;
    writeln ('ОТВЕТ: ');
    writeln ('При a=', a, ' b=', b, ' c=', c, ' максимальным
        является число ', max);
    writeln ('Программа завершила работу');
end.
    
```

Выполните расчёты по заданным значениям a , b , c . Результаты расчётов запишите в таблицу.

№№ п/п	Число a	Число b	Число c	Максимальное число
1.	5	20	12	
2.	23036	19472	31957	
3.	- 49451	45003	20194	
4.	-10,5	3,68	0,5	



Задача 5. Составить блок-схему алгоритма (самостоятельно) и написать программу расчёта корней квадратного уравнения.

```
program Branch_Programme;
var
  a, b, c : real;      {Коэффициенты уравнения}
  x1, x2 : real;      {Корни уравнения}
  d: real;            {Дискриминант}
begin
  writeln ('Программа решения квадратного уравнения');
  write ('Введите значение коэффициента a, a=');
  readln (a);
  write ('Введите значение коэффициента b, b=');
  readln (b);
  write ('Введите значение коэффициента c, c=');
  readln (c);
  d:= b*b - 4*a*c;    {Вычисление дискриминанта}
  if d >= 0
  then
    begin
      x1:= (- b + sqrt (d)) / (2*a);
      x2:= (- b - sqrt (d)) / (2*a);
      WriteLn; {Ввод пустой строки}
      WriteLn ('Корни уравнения:');
      WriteLn ('Корень x1=', x1);
      WriteLn ('Корень x2=', x2);
    end
  else
    writeln ('Действительных корней нет!');
end.
```

По заданным значениям a , b , c вычислите корни квадратного уравнения. Результаты расчётов запишите в таблицу.

№№ п/п	Коэффициенты уравнения			Корни уравнения	
	<i>a</i>	<i>b</i>	<i>c</i>	<i>x1</i>	<i>x2</i>
1.	5	3	-26		
2.	-2	6	10		
3.	1	4	3		
4.	4	1	7		
5.	0	5	2		

Задача 6. Составить блок-схему алгоритма (самостоятельно) и написать программу расчёта площади треугольника по формуле Герона с проверкой условия существования решения.

```

program Geron_1;
var
    a, b, c, d, s, p: real;
begin
    write ('Введите размер стороны a, a=');
    readln (a);
    write ('Введите размер стороны b, b=');
    readln (b);
    write ('Введите размер стороны c, c=');
    readln (c);
    p:=(a+b+c)/2;           {Полупериметр треугольника}
    d:= p*(p-a)*(p-b)*(p-c);
    if d<0
    then
        begin
            writeln ('Треугольника с такими размерами сторон нет');
            writeln ('Повторите, пожалуйста, ввод ... ');
        end
    else
        begin
            s:=sqrt(d);
            write ('Площадь треугольника s=', s:5:3);
        end;
end.
    
```

По заданным значениям размеров сторон a , b , c треугольника вычислите его площадь. Результаты расчётов запишите в таблицу.

№№ п/п	Размеры сторон треугольника			Площадь треугольника, S
	a	b	c	
1.	10	20	25	
2.	16,5	51,7	49,1	
3.	16500	51700	49100	
4.	12	34	46	
5.	26	40	11	

Дополнительное задание:

- Объясните результаты расчётов, полученные в п. 4 и п. 5.

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задача 1. Составить блок-схему алгоритма и написать программу расчёта функции Y :

$$y = \begin{cases} 2x, & \text{при } x < 5 \\ 2 + x & \text{— в остальных случаях} \end{cases}$$

Задача 2. Составить блок-схему алгоритма и написать программу выбора минимального числа из трёх заданных чисел.

Задача 3. Составить блок-схему алгоритма и написать программу расчёта функции Y :

$$y = \begin{cases} x^2 - 5, & \text{при } x > 10 \\ 5x, & \text{при } x < 10 \\ x + 5, & \text{при } x = 10 \end{cases}$$

Задача 4. Составить блок-схему алгоритма и написать программу определения чётности или нечётности натуральных чисел.

- Задача 5.** Составить блок-схему алгоритма и написать программу, которая позволяет определить возможность построения треугольника из отрезков прямых размерами a, b, c .
- Задача 6.** Заданы три действительных числа a, b, c . Составить блок-схему алгоритма и написать программу проверки выполнения условия $a > b > c$.
Ответ записать в виде: «Условие выполняется» или «Условие не выполняется»
- Задача 7.** Заданы два действительных числа a и b . Составить блок-схему алгоритма и написать программу проверки выполнения условий: $a > b, a < b$ или $a = b$.
Ответ записать в виде: «Число a больше, чем число b », «Число a меньше, чем число b », «Число a равно числу b »
- Задача 8.** Составить блок-схему алгоритма и написать программу расчёта функции $y = \sqrt{x}$ с проверкой условия существования решения.

3. ОПЕРАТОР ВЫБОРА В PASCAL ABC

При составлении программ на языке Pascal ABC встречаются задачи, в которых используют многократно вложенные друг в друга условные операторы *if...then*. При большом количестве операторов (три и более) программы становятся трудночитаемыми. Для компактной записи программ в таких случаях используют оператор выбора *Case...of*. Он имеет следующий вид:

```

case <переменная> of
  <значение_1> : <оператор_1>;
  <значение_2> : <оператор_2>;
  .....;
  <значение_n> : <оператор_n>;
else
  < оператор_n+1>
end;

```

Блок-схему оператора выбора можно представить следующим образом:



Описать работу оператора выбора *case...of* можно следующим образом: текущее значение *<Переменной>* определяет, какой *<Оператор>* будет выполнен. Например,

если <Переменная> принимает значение, которое находится в <Значение_2>, то выполняется <Оператор_2>, а остальные операторы пропускаются.

Таким образом, оператор *Case...of* обеспечивает удобный выбор одного из нескольких возможных направлений вычислительного процесса. Запомните, что переменная может быть только порядкового типа. Например, она может быть целым числом (*shortint, byte, integer, longint*), или иметь нечисловой порядковый тип (*char*) (A, B, C, ... Z; a, b, c, ... z).

В структуре оператора выбора *Case...of* можно использовать альтернативу *else*. Если в списке условий нет заданного значения переменной, то программа будет выполняться по ветви *else*. Часть программы, которая начинается с *else*, можно опустить. Тогда при отсутствии в списке переменных нужного условия, оператор выбора *Case...of* просто завершит свою работу.

Задача 3.1. Составить программу расчёта функции Y с использованием оператора выбора *Case...of*.

$$y = \begin{cases} x+1, & \text{при } -200 \leq x \leq -5 \\ x^2, & \text{при } -4 \leq x \leq 10 \\ 2x, & \text{при } 11 \leq x \leq 100 \\ x-50, & \text{при } 101 \leq x \leq 200 \end{cases}$$

program Listing_1;

var

x, y : integer;

begin

write ('Введите значение x, x=');

readln (x); *{Переменная x должна быть порядкового типа}*

case x of

-200..-5 : write ('При x=', x, ' значение функции y=', x+1);

*-4..10 : write ('При x=', x, ' значение функции y=', x*x);*

*11..100 : write ('При x=', x, ' значение функции y=', 2*x);*

101..200 : write ('При x=', x, ' значение функции y=', x-50);

else

write ('Число x=', x, ' находится за пределами диапазона!');

end; *{Конец выполнения оператора case...of}*

end. *{Конец выполнения программы}*

Рассчитайте значения функции Y при заданных значениях X . Результаты расчётов запишите в таблицу.

№№ п/п	Значение x	Значение функции Y
1.	-10	
2.	5	
3.	100	
4.	75,2	
5.	250	

Дополнительные задания

- Объясните результаты расчётов, полученные в п. 4 и п. 5.
- Внесите изменения в программу, которые позволят вычислять значения функции Y при изменении X на отрезках:
 $X \in [-20; -10]$, $X \in [-9; 0]$, $X \in [1; 10]$, $X \in [11; 20]$.

Задача 3.2. Составить программу определения делимости натуральных чисел на 5.

program Listing_2;

var

N : integer;

{Задаваемое число}

m : shortint;

{Остаток от деления}

begin

write ('Введите число N, N=');

readln (N);

m:= N mod 5;

case m of

0 : writeln ('Число N=', N, ' делится на 5 без остатка');

1..4 : writeln ('Число N=', N, ' делится на 5 с остатком ', m);

end;

{Конец выполнения оператора case...of}

end.



Дополнительные задания

1. Внесите изменения в программу, которые позволят определять делимость натуральных чисел на 3, на 7.
2. Внесите изменения в программу, позволяющие определять делимость отрицательных чисел.

Задача 3.3. Составить программу, которая позволяет последовательно запрашивать и выводить на экран монитора названия дней недели по их порядковым номерам.

```
program Listing_3;
label M1;                               {Определение метки}
var
    day : shortint;
begin
    M1:                                   {Метка M1}
    writeln;
    write ('Введите порядковый номер дня недели от 1 до 7. => ');
    readln (day);
    write (day);                          {Вывод на экран монитора номера дня недели}
    case day of
        1: writeln (' - Понедельник');
        2: writeln (' - Вторник');
        3: writeln (' - Среда');
        4: writeln (' - Четверг');
        5: writeln (' - Пятница');
        6: writeln (' - Суббота');
        7: writeln (' - Воскресенье');
        0:                               {Для выхода из программы введите цифру "0"}
            begin
                halt;
            end
        else
            writeln ('Такого дня недели не существует! Повторите ввод!');
    end;                                   {Конец выполнения оператора case...of}
    goto M1                                {Переход на метку M1}
end.
```



Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора результаты выполнения программы в виде текста: «Первый день недели – понедельник», «Второй день недели – вторник» и т.д.
2. Внесите изменения в программу, которые позволят выводить на экран монитора название месяца (Январь и т.д.) по его порядковому номеру.
3. Внесите изменения в программу, которые позволят выводить на экран монитора названия времён года по порядковому номеру месяца, например, «4 – Весна».
4. Внесите изменения в программу, которые позволят выводить на экран монитора цифры от 0 до 10 и писать их названия на русском и английском языках, например, «1 – один, one»; «2 – два, two» и т.д.

Задача 3.4. Составить программу, которая дописывает на чеке после суммы покупки слово «рубль» в нужном падеже.

```
program Listing_4;  
var  
    n, r : integer;  
begin  
    writeln ("Введите сумму покупки, n=");  
    readln (n);  
    if n>100  
    then n:=n mod 100;  
        if (n>= 11) and (n<=14)  
        then  
            writeln (' рублей')  
        else  
            begin  
                r:=n mod 10;
```



```
case r of
  0, 5..9 : writeln (n, ' рублей');
           1 : writeln (n, ' рубль');
           2..4 : writeln (n, ' рубля');
end;      {Конец выполнения оператора case...of}
end;
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят дописывать на чеке после суммы покупки слово «доллар» в нужном падеже.
2. Внесите изменения в программу, которые позволят выводить на экран монитора результаты работы программы в следующем виде: «Сумма вашей покупки составляет 51 рубль».

Задача 3.5. Составить программу, которая по введённому с клавиатуры символу определяет, что введено: цифра, прописная или строчная буквы латинского или русского алфавитов (кириллица).

```
program Listing_5;
var
  N : char; {Вводимый символ}
begin
  write ('Введите символ N=> ');
  readln (N);
  case N of
    '0'..'9' : writeln (N, ' - это цифра');
    'A'..'Z' : writeln (N, ' - это прописная буква латинского алфавита');
    'a'..'z' : writeln (N, ' - это строчная буква латинского алфавита');
    'А'..'Я' : writeln (N, ' - это прописная буква русского алфавита');
    'а'..'я' : writeln (N, ' - это строчная буква русского алфавита');
```



```
end;           {Конец выполнения оператора case...of}  
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не цифра».
2. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не буква».
3. Внесите изменения в программу, которые позволят выводить на экран монитора надпись: «Это не буква и не цифра. Это другой символ» при вводе неизвестных символов.
4. Внесите изменения в программу, которые позволят дополнительно выводить на экран монитора названия введённых символов:
 - - это точка, , - это запятая, ' - это апостроф,
: - это двоеточие, ; - это точка с запятой, / - это слеш,

Задача 3.6. Составить программу, которая по введённым с клавиатуры латинским буквам «Y» или «y» и «N» или «n» выводит на экран монитора фразы: «Этот символ означает Да» или «Этот символ означает Нет».

```
program Listing_6;  
var  
    ch : char ;  
begin  
    write ('Введите символы «Y», «y» или «N», «n» =>');  
    readln (ch);  
    case ch of  
        'Y', 'y' : writeln ('Этот символ означает <Да>')  
        'N', 'n' : writeln ('Этот символ означает <Нет>');  
    end  
    {Конец выполнения оператора case...of}  
end.
```



Задача 3.7. Составить программу, которая по введённым с клавиатуры цифрам пишет их названия на русском языке, например, «0 – это цифра ноль».

```
program Listing_7;  
var  
    x : shortint;  
begin  
    write ('Введите любую цифру от 0 до 9. => ');  
    readln (x);  
    case x of  
        0: writeln (x, ' – это цифра ноль');  
        1: writeln (x, ' – это цифра один');  
        2: writeln (x, ' – это цифра два');  
        3: writeln (x, ' – это цифра три');  
        4: writeln (x, ' – это цифра четыре');  
        5: writeln (x, ' – это цифра пять');  
        6: writeln (x, ' – это цифра шесть');  
        7: writeln (x, ' – это цифра семь');  
        8: writeln (x, ' – это цифра восемь');  
        9: writeln (x, ' – это цифра девять');  
    else  
        write ('Это не цифра! Это число');  
    end;  
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые позволят выводить на экран монитора названия чисел от 20 до 29 на русском языке, например, «24 – это число двадцать четыре».
2. Внесите изменения в программу, которые позволят выводить на экран монитора названия чисел от 145 до 154 на русском языке, например, «148 – это число сто сорок восемь».



Задача 3.8. Составить программу, которая выводит на экран монитора результаты сложения, вычитания и умножения двух целых чисел a и b по указанному программистом знаку арифметических действий («+», «-» или «*»).

```
program Listing_8;  
var  
    a, b, x : integer;  
    simbol : char;  
begin  
    write ('Введите число a, a=');  
    readln (a);  
    write ('Введите число b, b=');  
    readln (b);  
    write ('Введите знак арифметических действий => ');  
    readln (simbol);  
    case simbol of  
        '+' : x:=a+b;  
        '-' : x:=a-b;  
        '*' : x:=a*b;  
    end;  
    writeln ('Ответ: ', a, simbol, b, '=', x);  
end.
```

№№ п/п	Значение a	Значение b	$a+b=$	$a-b=$	$a*b=$
1.	5	2			
2.	19	47			
3.	538	961			
4.	25,6	69,5			

Дополнительные задания

1. Для нескольких значений a и b выполните действия сложения, вычитания и умножения. Результаты расчётов запишите в таблицу. Используйте калькулятор для проверки ответов.

2. Объясните результаты расчётов в п.3 и п.4. Какие изменения нужно внести в программу для получения правильных ответов?
3. Какие изменения нужно внести в программу для выполнения деления чисел a и b (a/b).

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

- Задача 1. Составить программу, которая по заданному номеру месяца (1 – январь, 2 – февраль, 3 – март и т.д.) выводит название соответствующего времени года («Зима», «Весна» и т.д.).
- Задача 2. Составить программу, которая по заданному номеру месяца (1 – январь, 2 – февраль, 3 – март и т.д.) выводит количество дней в этом месяце.
- Задача 3. Составить программу, которая выводит словесное описание оценки, полученной студентом:
2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отлично».
- Задача 4. Дано целое число в диапазоне от 1 до 99, которое соответствует возрасту человека в годах. Составить программу, которая по заданному возрасту выводит его описание на русском языке в нужном падеже, например, «Мне 41 год», «Мне 23 года», «Мне 18 лет».
- Задача 5. Составить программу, которая после введённой с клавиатуры денежной единицы от 1 до 99, дописывает слово «копейка» в нужном падеже, например, 1 копейка, 43 копейки, 65 копеек и т. д.

4. ЦИКЛИЧЕСКИЕ ПРОГРАММЫ В PASCAL ABC

Перед выполнением практических работ необходимо повторить тему «Циклические алгоритмы», а также изучить теоретический материал, используя конспект занятия и рекомендуемую литературу. При самостоятельном изучении материала указанной темы следует обратить внимание на то, что в своей практической деятельности человек часто решает задачи, в которых требуется многократно повторять одни и те же действия. Для составления алгоритмов решения таких задач используют команды повторения. Повторяющиеся действия в программировании называют циклами. В языке Pascal ABC есть три оператора, которые предназначены для выполнения циклических алгоритмов:

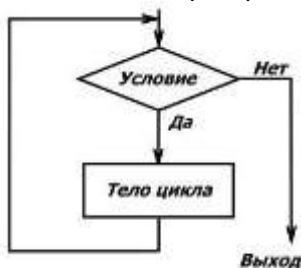
1.	оператор цикла с предусловием (<i>while</i>);
2.	оператор цикла с постусловием (<i>repeat</i>);
3.	оператор цикла с параметром (<i>for</i>).

Оператор цикла с предусловием

Структура оператора цикла с предусловием (предварительной проверкой условия) имеет вид:

```
while <условие> do  
begin  
    <тело цикла >;  
end;
```

Блок-схема оператора цикла с предусловием имеет вид:



Оператор цикла *while* организует выполнение одного оператора неизвестное число раз. При выполнении оператора цикла с предусловием сначала проверяется условие. Если логическое выражение является истинным (*True*), то выполняются тело цикла. После этого снова проверяется условие.

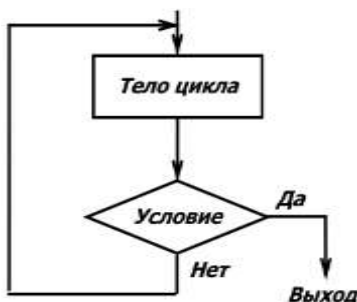
Этот процесс продолжается до тех пор, пока в результате проверки условия оно не окажется ложным (*False*).

Оператор цикла с постусловием

Структура оператора цикла с постусловием (с последующей проверкой условия) имеет вид:

```
repeat
begin
    < тело цикла >;
end;
until <условие>;
```

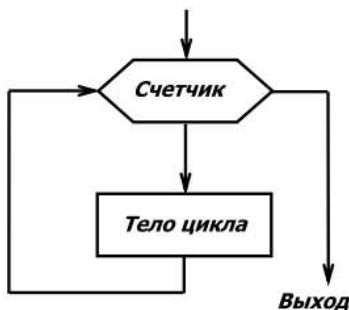
При выполнении оператора цикла с постусловием сначала выполняется один или группа операторов, а затем производится проверка условия. Выход из цикла осуществляется при истинности (*True*) некоторого логического выражения.



Слова *repeat* и *until* являются операторными скобками, поэтому перед словом *until* ставить точку с запятой необязательно.

Если между *begin* и *end* находится только одна инструкция, то слова *begin* и *end* можно не писать.

Оператор цикла с параметром



```
for i:=n1 to n2 do
    < Тело цикла >;
```

где *i* - параметр цикла;

n1 - начальное значение параметра цикла;

n2 - конечное значение параметра цикла.

Оператор цикла организует выполнение одного оператора известное число раз.

Существует второй вариант оператора цикла с параметром:

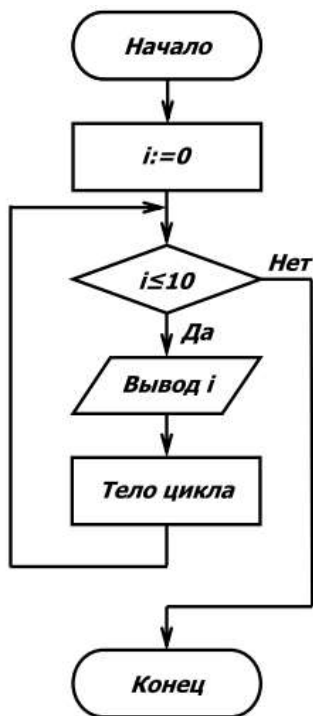
```
for i:=n2 downto n1 do
    < тело цикла >;
```

В этом случае оператор цикла начинает выполнять инструкции со значения $n2$ и заканчивает значением $n1$, т.е. в обратном порядке.

ПРАКТИЧЕСКАЯ РАБОТА

Задача 4.1. Составить блок-схему алгоритма и написать программу вывода на экран монитора последовательного ряда целых чисел от 0 до 10 с использованием оператора цикла с предусловием *while*.

4.1.1. Блок-схема алгоритма решения задачи:



4.1.2. Программа вывода на экран монитора чисел от 0 до 10

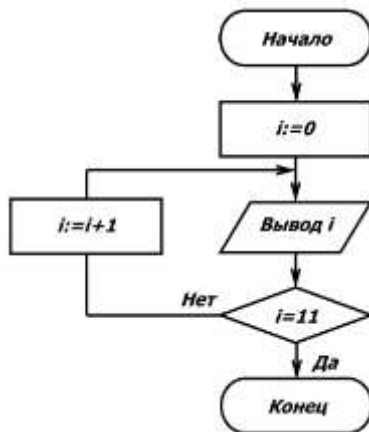
```
program CycleWhile;
var
    i: integer;
begin
    WriteLn ('Последовательный ряд чисел:');
    i:= 0;
    while i <=10 do
    begin
        Write (i, ' ');
        i:=i+1;
    end;
end.
```

4.1.3. Дополнительные задания к Задаче 4.1.

1. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от 10000 до 10010?
2. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от 50000 до 50010?
3. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от -50000 до -50010?
4. Какие изменения нужно внести в программу, чтобы вывести на экран монитора последовательный ряд чисел от 0 до 1 с шагом 0,05?
5. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда чисел от 10 до 100 с шагом 10?

Задача 4.2. Составить блок-схему алгоритма и написать программу вывода на экран монитора последовательного ряда целых чисел от 0 до 10 с использованием оператора цикла с постусловием *repeat*.

4.2.1. Блок-схема алгоритма решения задачи.



4.2.2. Программа вывода на экран монитора чисел от 0 до 10.

```

program CycleRepeat;
var
    i: integer;
begin
    writeln ('Последовательный ряд чисел:');
    i:= 0;
    repeat
        write (i, ' ');
        i:= i+1;
    until i=11;
end.
    
```

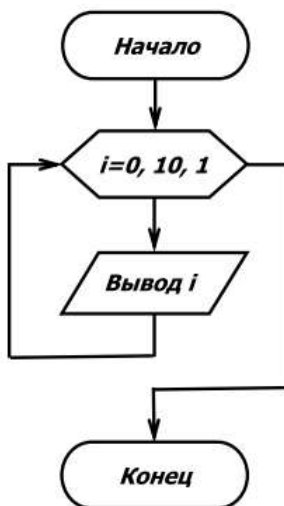
4.2.3. Дополнительные задания к Задаче 4.2.

1. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от 20 до 50 с шагом 5?
2. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда чисел от 2 до 4 с шагом 0,2?
3. Какие изменения нужно внести в программу, чтобы вывести на экран монитора двадцать символов " * "?

4. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от -100000 до -100010?

Задача 4.3. Составить блок-схему алгоритма и написать программу вывода на экран монитора последовательного ряда целых чисел от 0 до 10 с использованием оператора цикла с параметром *for*.

4.3.1. Блок-схема алгоритма решения задачи.



4.3.2. Программа вывода на экран монитора чисел от 0 до 10.

```

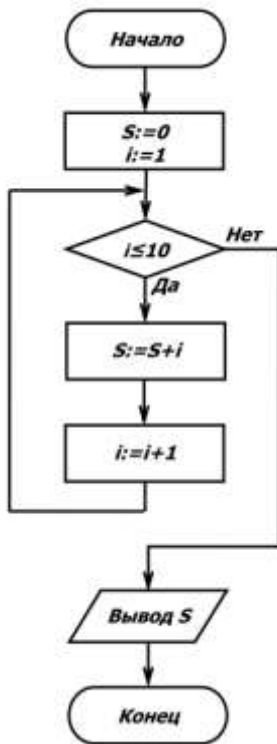
program CycleFor;
var
    i: integer;
begin
    writeln ('Последовательный ряд чисел:');
    for i:= 0 to 10 do
        write (i, ' ');
    end.
  
```

3.3. Дополнительные задания к Задаче 4.3.

1. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от 100 до 110?
2. Какие изменения нужно внести в программу для вывода на экран монитора последовательного ряда целых чисел от -2 до $+2$ с шагом $0,5$?
3. Какие изменения нужно внести в программу, чтобы вывести на экран монитора последовательный ряд чисел от 100000 до 100010?
4. Какие изменения нужно внести в программу для вывода на экран монитора двадцати символов « * » ?
5. Какие изменения нужно внести в программу, чтобы вывести на экран монитора пять предложений «Я учусь в ДГТУ» в одной строке?
6. Какие изменения нужно внести в программу, чтобы вывести на экран монитора предложение «Я учусь на факультете Международный» в десяти строках?

Задача 4.4. Составить блок-схему алгоритма и написать программу вывода на экран монитора суммы чисел от 1 до 10 с использованием оператора цикла с предусловием *while*.

4.4.1. Блок-схема алгоритма решения задачи:



4.4.2. Программа расчёта суммы чисел от 1 до 10.

```
program CycleSumWhile;
```

```
var
```

```
    i, s: integer;
```

```
begin
```

```
    s:= 0;
```

```
    i:= 1;
```

```
    while i <= 10 do
```

```
        begin
```



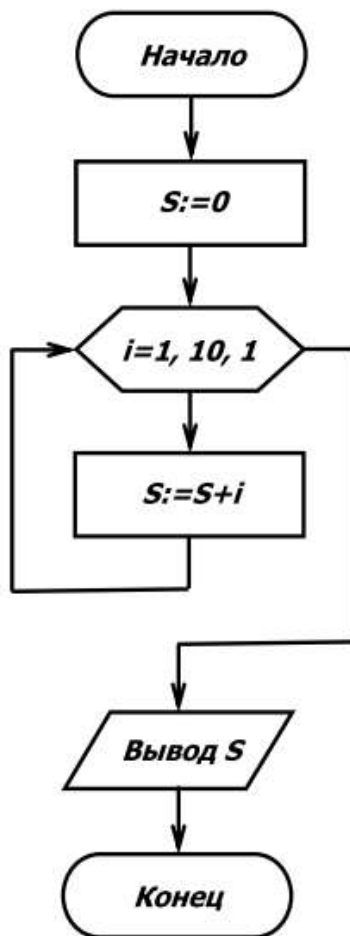
```
s:=s+i;  
i:=i+1;  
end;  
write ('Сумма натурального ряда чисел s=');  
writeln (s);  
end.
```

4.4.3. Дополнительные задания к Задаче 4.4.

1. Какие изменения нужно внести в программу для вывода на экран монитора суммы целых чисел от 10 до 20?
2. Какие изменения нужно внести в программу для вывода на экран монитора суммы целых чисел от -10 до 20?
3. Какие изменения нужно внести в программу для вывода на экран монитора суммы чисел от 1 до 2 с шагом 0,1?
4. Какие изменения нужно внести в программу для вывода на экран монитора суммы целых чисел от -10 до +10? Объясните полученный ответ.

Задача 4.5. Составить блок-схему алгоритма и написать программу вывода на экран монитора суммы натурального ряда чисел от 1 до 10 с использованием оператора цикла с параметром *for*.

4.5.1. Блок-схема алгоритма решения задачи.



4.5.2. Программа расчёта суммы чисел от 1 до 10.

```
program CycleSumFor;  
var  
    s, i: integer;  
begin  
    s:= 0;  
    for i:=1 to 10 do  
        s:= s + i;  
    write ('Сумма натурального ряда чисел S=' , S);  
end.
```

4.5.3. Дополнительные задания к Задаче 4.5.

1. Какие изменения нужно внести в программу для вывода на экран монитора суммы последовательного ряда чисел от 1 до 45?
2. Какие изменения нужно внести в программу для вывода на экран монитора суммы последовательного ряда чисел от 50 до 80?
3. Какие изменения нужно внести в программу для вывода на экран монитора ответа в виде: «Сумма натурального ряда чисел от 1 до 10 составляет $S=55$ »?
4. Какие изменения нужно внести в программу, чтобы компьютер запрашивал начальное и конечное значения последовательного ряда чисел?
5. Какие изменения нужно внести в программу для вывода на экран монитора указания в виде: «Программа завершила работу. Нажмите клавишу Enter»?

Задача 4.6. Составить программу вывода на экран монитора прописных букв латинского алфавита от *A* до *Z*.

4.6.1. Программа вывода на экран букв латинского алфавита.

```
program CycleLetter;  
var  
    letter: char;  
begin  
    for letter:= 'A' to 'Z' do  
        write (letter);  
end.
```

4.6.2. Дополнительные задания к Задаче 4.6.

1. Какие изменения нужно внести в программу, чтобы вывести на экран монитора прописные буквы латинского алфавита от *A* до *S*?
2. Какие изменения нужно внести в программу, чтобы вывести на экран монитора строчные буквы русского алфавита от *А* до *Я*?
3. Какие изменения нужно внести в программу, чтобы вывести на экран монитора строчные буквы латинского алфавита от *z* до *a*?
4. Какие изменения нужно внести в программу, чтобы вывести на экран монитора строчные буквы русского алфавита от *я* до *а*?
5. Какие изменения нужно внести в программу, чтобы вывести на экран монитора прописные буквы русского алфавита от *Я* до *Ж*?

Задача 4.7. Составить программу расчёта и вывода на экран монитора простых чисел до заданного целого числа.

Примечание: Простым называется число, которое делится только на себя и на 1. Например, число 17 можно разделить только на «17» и на «1». Примеры простых чисел: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41 и т.д.

4.7.1. Программа вывода на экран монитора простых чисел.

```
program Prime_number;  
var  
    i, j, lim, Number: integer;  
begin  
    writeln ('Программа расчёта простых чисел');  
    write ('Введите любое целое положительное число =>');  
    read (Number);  
    writeln ('Все простые числа до ', Number, ':');  
    for i:=1 to Number do begin  
        j:=2;  
        lim:=round (sqrt (i));  
        while (i mod j<>0) and (j<=lim) do inc(j);  
        If j>lim  
            then write (i, ' ');  
    end;  
end.
```

4.7.2. Дополнительные задания к Задаче 4.7.

1. Выведите на экран монитора все простые числа от 1 до 50.
2. Какие изменения нужно внести в программу, чтобы вывести на экран монитора все простые числа от 1 до 100 ?

Задача 4.8. Составить программу вывода на экран монитора всех делителей натурального числа.

4.8.1. Программа вывода на экран делителей целых чисел.

```
program Division_of_natural_numbers;  
var  
    a, n, c, d: integer;  
begin  
    writeln ('Расчёт делителей числа');  
    write ('Введите любое      число a, a=');
```



```
readln (a);  
write ('Число ', a, ' делится без остатка на: ');  
n:=1;  
  while (n <= sqrt (a)) do  
  begin  
    c:=a mod n;  
    d:=a div n;  
    if c = 0 then begin  
      write (n, ' ');  
      if n <> d then write (d, ' ');  
    end;  
    inc (n);  
  end;  
end.
```

4.8.2. Дополнительные задания к Задаче 4.8.
Расчитайте делители натуральных чисел.

№№ п/п	Натуральное число	Делители числа
1.	28	
2.	145	
3.	450	
4.	1246	
5.	23162	

Задача 4.9. Составить программу вывода на экран монитора совершенных чисел.

Примечание: Совершенным называется число, которое равно сумме всех своих делителей. Например, 6, 28, 496, 8128 и т.д.

Совершенное число 6 имеет делители: 1, 2, 3; их сумма равна 6. Совершенное число 28 имеет делители: 1, 2, 4, 7, 14; их сумма равна 28.



Задача 4.10. Найдите все совершенные числа до 1000000.

```
program Interesting_Number;  
var n, i, j, s, lim, c, d, x : longint;  
begin  
  writeln (' Программа расчёта совершенных чисел');  
  write ('Введите любое целое число =>');  
  read (x);  
  write ('Это совершенные числа до ', x, ': ');  
  for i:=1 to x do begin  
    s:=1; lim:=round (sqrt (i));  
    for j:=2 to lim do begin  
      c:=i mod j;  
      d:=i div j;  
      if c = 0 then begin  
        inc (s, j);  
        if (j<>d) then inc(s, d);  
      end;  
    end;  
    if s=i  
      then write (i, ' ');  
  end;  
end.
```

Задача 4.11. Составить программу перевода двоичных чисел в десятичную систему счисления.

4.11.1. Программа перевода двоичных чисел в десятичную систему счисления.

```
program Translation_of_2_to_10;  
var  
  x: string;  
  function BIN_DEC (x: string): longint;  
  const digits: array [0..1] of char = ('0', '1');  
  var  
    res, ves: longint;  
    i, j: byte;
```



```
begin
    res:=0; ves:=1;
    for i:=length(x) downto 1 do begin
        j:=0;
        while (digits[j]<>x[i]) do inc(j);
        res:= res + ves * j;
        ves:=ves*2;
    end;
    BIN_DEC:=res;
end;
begin
    writeln ('Программа перевода двоичных чисел
        в десятичные');
    write ('Введите двоичное число =>');
    read (x);
    writeln ('Двоичное число ', x, ' это ', BIN_DEC (x),
        ' в десятичной системе счисления');
    writeln ('Программа завершила работу');
end.
```

10.2. Дополнительное задание к Задаче 9. Переведите двоичные числа в десятичную систему счисления.

№№ п/п	Двоичное число	Десятичное число
1.	11011 ₂	
2.	101101 ₂	
3.	1101101 ₂	
4.	11110101 ₂	
5.	110101111 ₂	



Задача 4.12. Составить программу перевода десятичных чисел в двоичную систему счисления.

11.1. Программа перевода десятичных чисел в двоичную систему счисления.

```
program Translation_of_10_to_2;
var
    a: longint;
    function DEC_BIN (x: longint): string;
    const digits: array[0..1] of char = ('0', '1');
var
    res: string;
    d: 0..1;
begin
    res:= ' ';
    while (x<>0) do begin
    d:=x mod 2;
    res:=digits[d]+res;
    x:=x div 2;
end;
    DEC_BIN:=res;
end;
begin
    writeln ('Программа перевода двоичных чисел
        в десятичные');
    write ('Введите десятичное число =>');
    readln (a);
    writeln ('Десятичное число ', a, ' это ', DEC_BIN (a),
        ' в двоичной системе счисления');
end.
```

4.12. Дополнительное задание к Задаче 4.12. Переведите десятичные числа в двоичную систему счисления.

№№ п/п	Десятичное число	Двоичное число
1.	9	
2.	84	
3.	121	
4.	256	
5.	1947	

Задача 4.13. Составить программу определения количества разных букв в слове.

4.13.1. Программа определения количества разных букв в слове.

```

program Number_of_Letters;
var s: string;
    r: real;
    i, j, n: integer;
begin
    r:=0;
    write ('Напишите любое русское слово => ');
    readln (s);
    for i:=1 to length(s) do begin
        n:=0;
        for j:=1 to length(s) do begin
            If s[i]=s[j] then inc(n);
        end;
        r:=r+1/n;
    end;
    writeln ('Количество разных букв в этом слове - ', r:1:0);
end.
    
```

4.13.2. Дополнительное задание к Задаче 13.

Посчитайте количество разных букв в словах.

№№ п/п	Слово	Количество разных букв в этом слове
1.	Университет	
2.	Факультет	
3.	Информатика	
4.	Аудитория	
5.	Химия	

5. МАССИВЫ В PASCAL ABC

Массив – это последовательный ряд однотипных элементов, которые имеют общее имя, а каждый элемент этой последовательности определяется порядковым номером, например, $a_1, a_2, a_3, \dots, a_n$. Элементы массива нумеруют по порядку. Порядковый номер элемента массива называется индексом этого элемента.

Например, результаты измерения температуры воздуха в течение месяца можно рассматривать как совокупность вещественных чисел, объединённых в один объект – массив измерений температуры.

При описании массива используются зарезервированные слова *array* (массив) и *of* (из). В квадратных скобках указывают число элементов, которые входят в массив и тип этих элементов. В общем виде описание массива задаётся следующим образом:

```
a: array [1..10] of integer;  
b: array [1..20] of real;  
c: array [0..30] of char;
```

Из этого описания массивов видно, что массив *a* состоит из 10 (десяти) целых элементов (*integer*), массив *b* состоит из 20 вещественных элементов (*real*), а массив *c* – из 30 символьных элементов (*char*), например, букв.

Использование языка программирования Pascal ABC при работе с массивами позволяет:

- обеспечить ввод и вывод элементов массива;
- определять сумму элементов массива;
- определять среднее арифметическое значение элементов массива;
- определять максимальное и минимальное значения элементов массива;
- сортировать элементы массива по возрастанию или по убыванию и др.

При выполнении лабораторно-практической работы следует проанализировать работу программы, написать текст программы на компьютере в среде программирования языка Pascal ABC, проверить правильность написания программы (клавиша F9), запустить программу на выполнение (F9). Затем следует выполнить дополнительные задания, которые приведены в конце каждой



задачи. Результаты расчётов запишите в соответствующие таблицы.

Задача 1. Составить программу ввода с клавиатуры и вывода на экран монитора пяти элементов массива.

```
program Massiv_01;  
const n = 5;           {n – количество элементов массива}  
var  
    a: array [1..n] of integer;  
    i: integer;  
begin  
    writeln ('Введите ', n, ' элементов массива с клавиатуры:');  
    for i:=1 to n do  
        begin  
            write ('Введите значение элемента a[', i, '] = ');  
            readln (a[i]);  
        end;  
    writeln ('Вывод элементов массива на экран монитора ');  
    for i:=1 to n do  
        write (a[i]:5);  
end.
```

Дополнительные задания

1. Какие изменения нужно внести в текст программы для ввода и вывода на экран монитора десяти элементов массива?
2. Какие изменения нужно внести в текст программы для вывода на экран монитора названия темы занятия: «Массивы в Pascal ABC»?
3. Какие изменения нужно внести в текст программы для вывода на экран монитора текста: «Программа завершила работу!»?

Задача 2. Составить программу расчёта суммы элементов массива из 74 пяти чисел.



```
program Massiv_02;  
const n = 5;           {n – количество элементов массива}  
var  
    i: integer;  
    a: array [1..n] of integer;  
    Sum: integer;     {Sum - сумма элементов массива}  
begin  
    writeln ('Расчёт суммы элементов массива');  
    writeln ('Введите ', n, ' элементов массива:');  
    for i:=1 to n do  
        begin  
            write ('Введите значение элемента a[', i, '] =');  
            readln (a[i]);  
        end;  
    Sum:=0;  
    for i:=1 to n do  
        Sum:= Sum + a[i];  
    writeln ('Сумма элементов массива равна ', Sum);  
end.
```

Дополнительные задания

Напишите программу на компьютере и выполните расчёты.
Результаты расчётов запишите в таблицу.

№№ п/п	a ₁	a ₂	a ₃	a ₄	a ₅	Сумма элементов массива
1.	4	1	6	2	5	
2.	56	90	39	21	15	
3.	1000	3000	5000	9000	70000	
4.	5,2	4,5	0,4	1,9	4,7	



Задача 3. Составить программу расчёта среднего арифметического значения элементов массива из 5 чисел.

```
program Massiv_03;  
const n=5;  
var  
    a: array [1..n] of integer;  
    i: integer;  
    Sred: real; {Среднее арифметическое значение элементов массива}  
begin  
    writeln ('Расчёт среднего арифметического значения  
            элементов массива');  
    writeln ('Введите ', n, ' элементов массива:');  
    for i:=1 to n do  
        begin  
            write ('Введите значение элемента a[' , i , ']=');  
            readln (a[i]);  
        end;  
    Sred:=0;  
    for i:=1 to n do  
        Sred:= (Sred + a[i]);           {Сумма элементов массива}  
    Sred:= Sred / n;  
    writeln ('Среднее арифметическое значение элементов  
            массива равно ', Sred);  
end.
```

Дополнительные задания

Напишите программу на компьютере. Вычислите среднее арифметическое по заданным значениям пяти элементов массива. Результаты расчётов запишите в таблицу.

№№ п/п	a ₁	a ₂	a ₃	a ₄	a ₅	Среднее арифметическое
1.	4	5	6	2	1	
2.	56	90	39	21	15	
3.	1000	30000	50	300	7000	
4.	5,2	4,5	0,4	1,9	4,7	

Если количество элементов массива заранее не задано, то для расчёта суммы его элементов можно воспользоваться программой, приведённой в Задаче 4. Следует помнить, что



количество элементов массива может быть только целым числом.

Задача 4. Составить программу расчёта суммы элементов массива из n чисел.

```
program Massiv_04;  
var  
    i: integer;  
    n: integer;  
    a: array [1..20] of integer; {20 – количество элементов массива}  
    Sum: integer;  
begin  
    write ('Расчёт суммы элементов массива из n чисел, n= ');  
    readln (n);  
    writeln ('Введите ', n, ' элементов массива:');  
    for i:=1 to n do  
        begin  
            write ('Введите значение элемента a[', i, '] =');  
            readln (a[i]);  
        end;  
    Sum:=0;  
    for i:=1 to n do  
        Sum:= Sum + a[i];  
    writeln ('Сумма элементов массива равна ', Sum);  
end.
```

Дополнительные задания

Выполните расчёты для 5 (пяти), 10 (десяти) и 20 (двадцати) элементов массива. Для расчётов возьмите любые целые и вещественные значения элементов массива. Проверьте правильность выполнения расчётов на калькуляторе.



Задача 5. Составить программу заполнения массива пятью случайными числами из диапазона от 0 до 100. Вывести элементы этого массива на экран монитора.

```
program Massiv_05;  
var  
    a: array[1..100] of integer;  
    i: integer;  
begin  
    Writeln ('Вывод 5 элементов массива на экран монитора:');  
    randomize;  
    for i:=1 to 5 do      {Выбор 5 случайных чисел из 100}  
        begin  
            a[i]:=random (100);  
            Writeln ('Значение массива a[', i, ']= ', a[i]);  
        end;  
end.
```

Дополнительные задания

1. Составить программу заполнения массива 10 (десятью) случайными числами из диапазона от 0 до 100.
2. Составить программу заполнения массива десятью случайными числами из диапазона от 0 до 1000.
3. Составить программу заполнения массива десятью случайными числами из диапазона от 0 до 1000.

Задача 6. Составить программу сортировки элементов массива по возрастанию.

```
program Massiv_06;  
const n=5;      {n – количество элементов массива}  
var  
    a: array [1..n] of integer;  
    i, k, w: integer;  
begin
```



```
writeln ('Сортировка элементов массива по возрастанию');  
writeln ('Введите ', n, ' элементов массива:');  
  for i:=1 to n do  
    begin  
      write ('Введите значение элемента a[' , i, ']= ');  
      readln (a[i]);           {Ввод массива}  
    end;  
  for k:=n-1 downto 1 do  
    for i:=1 to k do  
      if a[i] > a[i+1] then  
        begin  
          w:=a[i];  
          a[i]:=a[i+1];  
          a[i+1]:=w;  
        end;  
writeln ('Сортировка элементов массива по возрастанию:');  
  for i:=1 to n do  
write (a[i]:4);  {4 – количество пустых позиций между числами}  
end.
```

Дополнительные задания

1. Внесите изменения в программу, которые обеспечивают сортировку по возрастанию 7 (семи) чисел.
2. Внесите изменения в программу, которые обеспечивают вывод на экран монитора результатов сортировки семи чисел по вертикали.
3. Внесите изменения в программу, которые обеспечивают сортировку по убыванию пяти чисел.

Задача 7. Составить программу автоматического заполнения массива 10 (десятью) случайными числами и найти максимальный элемент этого массива.

```
program Massiv_07;  
const n=10;  
var  
  a: array [1..n] of integer;  
  i: integer;  
  max: integer;
```



begin

```
writeln ('Автоматическое заполнение массива случайными  
числами');  
writeln ('Нахождение максимального элемента массива');  
randomize;  
for i:=1 to n do  
begin  
    a[i]:= random(100);    {Выбор n чисел из 100}  
writeln ('Значение ', i, ' элемента массива равно ', a[i]);  
end;  
max:=a[1];  
for i:=1 to n do  
begin  
    if a[i] > max then max:=a[i];  
end;  
writeln ('Максимальный элемент массива равен ', max);  
end.
```

Дополнительные задания

1. Какие изменения нужно внести в программу для автоматического заполнения массива 5, 10 и 20 случайными числами из 100? Чему равен максимальный элемент массива?
2. Какие изменения нужно внести в программу для автоматического заполнения массива 4, 8 и 15 случайными числами из 500? Чему равен максимальный элемент массива?

ЛИТЕРАТУРА И ИНТЕРНЕТ-ИСТОЧНИКИ

1. Моренко Б.Н. Информатика. Алгоритмы и алгоритмизация: учеб.-метод. пособие для иностранных слушателей дополнительных общеобразовательных программ, инженерно-технической и технологической направленности обучения [Электрон. ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/96a0a634-bb48-42f8-a74d-eb70601695f1/3832/4903/3732.pdf>
Рег. номер 3732 от 12.05.2017 г.
2. Моренко Б.Н. Информатика. Линейные программы в Turbo Pascal: метод. указания к лаб.-практ. работам для иностранных студентов предвузовской подготовки, обучающихся по образовательным программам инженерно-технической, технологической и естественнонаучной направленности. [Электрон. ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/2924/2727.pdf>
Рег. номер 2727 от 01.02.2016 г.
3. Методические указания к интегрированной практической работе по информатике и математике для иностранных студентов предвузовской подготовки «Решение квадратных уравнений с использованием языка Turbo Pascal» [Электронный ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/1603/1384.pdf>
Рег. номер 1384 от 30.09.2013.
4. Моренко Б.Н. Расчет электрического сопротивления цепи: практикум по дисциплине «Информатика» дополнительные общеобразовательные программы для иностранных граждан и лиц без гражданства [Электронный ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/2335/2176.pdf>
Рег. номер 2176 от 06.02.2015.

5. Моренко Б.Н. Информатика. Программы ветвления в Turbo Pascal: практикум по дисциплине «Информатика» для слушателей дополнительных общеобразовательных программ, обеспечивающих подготовку иностранных граждан и лиц без гражданства к освоению профессиональных образовательных программ на русском языке [Электрон. ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/3426/4903/3302.pdf>
Рег. номер 3302 от 09.01.2017 г.
6. Моренко Б.н. Использование ветвлений в Turbo Pascal при решении экономических задач: метод. указания [Электрон. ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова, О.М. Воскерчьян.
– Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/1465/1171.pdf>
Рег. номер 1171 от 01.04.2013.
7. Моренко Б.Н. Подготовка к итоговой аттестации по информатике : учеб.-метод. пособие / Б.Н. Моренко, Л.Д. Бабакова, О.В. Николенко; ДГТУ. – Ростов н/Д. : ИЦ ДГТУ, 2019. - 52 с.
– Режим доступа: <https://ntb.donstu.ru/content/podgotovka-k-itogovoy-attestacii-po-informatike.pdf>