

## Коротко о Паскаль

Язык Паскаль был разработан Никлаусом Виртом в 1970 г. как язык со строгой типизацией и интуитивно понятным синтаксисом. В 80-е годы наиболее известной реализацией стал компилятор Turbo Pascal фирмы Borland, в 90-е ему на смену пришла среда программирования Delphi (Object Pascal), которая стала одной из лучших сред для быстрого создания приложений под Windows. Delphi ввела в язык Паскаль ряд удачных объектно-ориентированных расширений. Альтернативная реализация Object Pascal – компилятор Free Pascal.

PascalABC.NET – мощный и современный язык программирования, ориентированный прежде всего на обучение. Прототип PascalABC.NET – учебная система Pascal ABC – появилась в 2002 году. PascalABC.NET – реализация Delphi, в которой отсутствуют специфические языковые средства Delphi, не имеющие особого значения вне среды Delphi.

[Для более подробного ознакомления с системой программирования PascalABC.NET обратитесь сюда.](#)

## Разделы программ на языке Паскаль

```

program ИмяПрограммы;
uses
    ИмяМодуля1, ..;
label
    ИмяМетки1, ..;
const
    ИмяКонстанты = ЗначениеКонстанты;
type
    ИмяТипа = ЗначенияТипа;
var
    ИмяПеременной: Тип;
    {объявления процедур и функций программиста}
begin
    {инструкции основной программы}
end.
    
```

## Объявление переменных

- объявления переменных помещают в раздел, который начинается словом **var**:

```

var
    i: integer; {ИмяПеременной: Тип;}
    
```

- Числовыми типами языка Pascal являются: целые – **integer**, **int64**, **shortint** (-128 ..127), **byte** (0..255), **word** (0..65536), дробные – **real** (вещественный), **single** (с одинарной точностью), **double** (с двойной точностью). Логический тип – **boolean** (булевский), символьный тип – **char**, строковый тип – **string**.

## Присвоение и арифметические операторы

- для присвоения значения переменным используется **оператор присваивания** ("**:=**").

Пример записи: **n := 123** {переменная := выражение};

- к **арифметическим операторам** относятся: **+** (сложение), **-** (вычитание), **\*** (умножение), **/** (деление), **div** (целочисленное деление) и **mod** (остаток от деления).

## Математические функции

- для получения модуля числа используется функция **abs (x)**. Функция **round (x)** округляет число **x** до ближайшего целого, **ceil (x)** – возвращает наименьшее целое, большее или равное **x**, **floor (x)** – наибольшее целое, меньшее или равное **x**, **int (x)** – возвращает целую часть, а **frac (x)** – дробную часть числа **x**;
- функция **random** возвращает случайное вещественное число в диапазоне от **0** до **1**, **random (a)** – целое число в диапазоне от **0** до **a-1**, **random (a, b)** – целое число в диапазоне [**a..b**];
- функция **sqrt (x)** – возвращает квадратный корень числа **x**, **sqr (x)** – квадрат числа, **power (x, y)** – **x** в степени **y**, **logn (y, x)** – логарифм числа **x** по основанию **y**;
- функции **sin (x)**, **cos (x)**, **tan (x)**, **arcsin (x)**, **arccos (x)**, **arctan (x)** используются для получения синуса, косинуса, тангенса, арксинуса, арккосинуса или арктангенса числа **x** соответственно.

## Ввод и вывод

- для вывода на экран монитора сообщений и значений переменных предназначены инструкции **write** и **writeln**:

```

write (a); write ('a'); writeln (b); writeln ('b');
    
```

- при выводе можно задавать формат, например:

```

write (15.5432;x;y);
    
```

где **x** – отводимое к-во мест (символов) для вывода, **y** – число знаков дробной части числа.

- для ввода с клавиатуры во время работы программы исходных данных (значений переменных) предназначены инструкции **read** и **readln**:

```

read (a); readln (a); read (a, b); readln (a, b, c);
    
```

- инструкции **readln** и **writeln** (после ввода/вывода данных) осуществляют переход на следующую строку.

### Задания 1-9:

1. Напишите программу, которая выводит на экран четверостишие:

Унылая пора! Очей очарованье!  
 Приятна мне твоя прощальная краса -  
 Люблю я пышное природы увяданье,  
 В багрец и золото одетые леса.

А.С. Пушкин

[решение тут](#)

2. Напишите программу для вывода значения числа  $\pi$  (константа  $\pi$ ) с тремя цифрами в дробной части. Вид экрана вывода:

3.142

[решение тут](#)

3. Напишите программу для вывода значений дробных переменных  $x1$  и  $x2$ . На экране перед значением переменной должен быть выведен поясняющий текст, представляющий собой имя переменной, за которым следует знак "равно". Рекомендуемый вид экрана во время работы программы представлен ниже (данные, вводимые пользователем, выделены полужирным курсивом):

Введите в одной строке через пробел числа  $x1$  и  $x2$  -> **1.83 3.1548**

$x1 = 1.830$

$x2 = 3.155$

[решение тут](#)

4. Напишите программу вычисления площади прямоугольника. Рекомендуемый вид экрана во время работы программы:

Вычисление площади прямоугольника

Введите исходные данные:

Длина (см) -> **5**

Ширина (см) -> **7.3**

Площадь прямоугольника: 36.50 кв.см.

[решение тут](#)

5. Напишите программу вычисления площади треугольника по длине основания и высоте. Вид экрана во время работы программы:

Вычисление площади треугольника по длине основания и высоте

Введите исходные данные:

Основание (см) -> **4.4**

Высота (см) -> **3.8**

Площадь треугольника: 8.36 кв.см

[решение тут](#)

6. Напишите программу вычисления стоимости покупки, состоящей из нескольких тетрадей и карандашей. Вид экрана во время работы программы:

Вычисление стоимости покупки

Введите исходные данные:

Цена тетради (руб.) -> **2.75**

Количество тетрадей -> **5**

Цена карандаша (руб.) -> **0.85**

Количество карандашей -> **2**

Стоимость покупки: 15.45 руб.

[решение тут](#)

7. Напишите программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Вид экрана:

Преобразование числа в денежный формат

Введите дробное число -> **123.4529**

123.45 руб. - это 123 руб. 45 коп.

[решение тут](#)

8. Напишите программу вычисления длины гипотенузы прямоугольного треугольника, если заданы два катета. Вид экрана:

Вычисление длины гипотенузы прямоугольного треугольника

Введите через пробел длины катетов: **3.14 5.682**

Длина гипотенузы: 6.492

[решение тут](#)

9. Напишите программу вычисления площади треугольника, если известны три его стороны (площадь треугольника по известным трём его сторонам можно определить по формуле Герона:

$$s = \sqrt{p * (p - a) * (p - b) * p - c},$$

где  $p$  – полупериметр треугольника). Вид экрана:

Вычисление площади треугольника

Введите через пробел длины трех сторон: **3.3 4.8 6.1**

Площадь треугольника: 7.877

[решение тут](#)

### Выбор If

- инструкция **if** используется для выбора одного из двух серий инструкций, выполняемых во время работы программы в зависимости от соблюдения заданного условия:

```
if {условие} then
    {оператор1}
else {оператор2};
```

- условие – это выражение логического типа, и может принимать одно из двух значений: **true** (истина) или **false** (ложь). При помощи логических операторов **and** и **or** из простых условий можно строить сложные.

### Задания 10-14:

10. Напишите программу, которая вычисляет частное от деления двух чисел. Программа проверяет правильность введенных пользователем данных и, если они неверные (делитель равен нулю), выдаёт сообщение об ошибке. Вид экрана:

Вычисление частного  
Введите в одной строке делимое и делитель,  
затем нажмите <Enter> -> **12 0**  
Ошибка! Делитель не может равняться нулю.

[решение тут](#)

11. Напишите программу вычисления площади кольца. Программа должна проверять правильность вводимых данных. В случае ошибки должно выводиться сообщение:

Ошибка! Радиус отверстия не может быть больше радиуса кольца.

[решение тут](#)

12. Напишите программу вычисления сопротивления электрической цепи, состоящей из двух сопротивлений. Сопротивления могут быть соединены последовательно или параллельно:

Вычисление сопротивления электрической цепи  
Введите исходные данные:  
Величина первого сопротивления (Ом) -> **15**  
Величина второго сопротивления (Ом) -> **27.3**  
Тип соединения (1 - последовательное, 2 - параллельное) -> **2**  
Сопротивление цепи: 9.68 Ом.

[решение тут](#)

13. Напишите программу вычисления стоимости покупки с учётом скидки. Скидка в 10% предоставляется, если сумма покупки больше 1000 руб. Вид экрана:

Вычисление стоимости покупки с учетом скидки  
Введите сумму покупки и нажмите <Enter> -> **1200**  
Вам предоставляется скидка 10%  
Сумма покупки с учетом скидки: 1080.00 руб.

[решение тут](#)

14. Напишите программу вычисления стоимости покупки с учётом скидки. Скидка в 3% предоставляется в случае, если сумма покупки больше 500 руб., в 5% – если сумма больше 1000 руб. Вид экрана:

Вычисление стоимости покупки с учетом скидки  
Введите сумму покупки и нажмите <Enter> -> **1100**  
Вам предоставляется скидка 5%.  
Сумма покупки с учетом скидки: 1045.00 руб.

[решение тут](#)

### Выбор Case

- инструкция **case** используется для выбора одного из нескольких серий (последовательностей) выполняемых инструкций во время работы программы, в зависимости от значения переключателя:

```
case {переключатель} of
  {список выбора 1}: {оператор1};
  ...
  {список выбора N}: {операторN};
else {оператор0};
end;
```

- в качестве переключателя могут использоваться переменные порядкового, т.е. целого (integer) или символического (char) типов.

### Задания 15-19:

15. Напишите программу, которая запрашивает ввод номера месяца и выводит соответствующее название времени года. При вводе некорректных данных, программа должна вывести сообщение «Должно быть число от 1 до 12». Вид экрана:

Введите номер месяца (число от 1 до 12) и нажмите <Enter> -> **11**  
11-й месяц – это зима.

[решение тут](#)

16. Напишите программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота» или «Воскресенье». Вид экрана:

Введите номер недели (число от 1 до 7) и нажмите <Enter> -> **3**  
Рабочий день

[решение тут](#)

17. Напишите программу, которая после введенного с клавиатуры числа (в диапазоне от 1 до 20), обозначающего денежную единицу, дописывает слово «рубль» в правильной форме. Например, «12 рублей», «3 рубля» и т.д. Вид экрана:

Введите количество рублей (от 1 до 20) и нажмите <Enter> -> **12**  
12 рублей

[решение тут](#)

18. Напишите программу, которая после введенного с клавиатуры числа (в диапазоне от 1 до 99), обозначающего денежную единицу, дописывает слово «копейка» в правильной форме. Например, «5 копеек», «41 копейка» и т.д. Вид экрана:

Введите количество копеек и нажмите <Enter> -> **44**  
44 копейки

[решение тут](#)

19. Напишите программу, вычисляющую стоимость междугороднего телефонного разговора. Исходными данными для программы являются код города и продол-

жительность разговора. Цена одной минуты зависит от расстояния до города, в котором находится абонент:

Город	код	Цена минуты, руб.
Владивосток	423	10,0
Москва	095	5,0
Мурманск	815	8,0
Ростов	863	2,5
Самара	846	5,5

Вид экрана:

Вычисление стоимости разговора по телефону  
 Введите исходные данные: Код города → **423**  
 Длительность (целое количество минут) → **3**  
 Город: Владивосток. Цена минуты: 10.00 руб.  
 Стоимость разговора: 30.00 руб.

[решение тут](#)

### Циклы For

- инструкция **for** используется для организации циклов с фиксированным, определяемым во время разработки программы, числом повторений;
- количество повторений цикла определяется начальным и конечным значениями переменной-счетчика [счетчик может быть целого (*integer*, *byte*) или символьного (*char*) типов]:  

```
for i:={начальное_значение} to {конечное_значение} do
    {оператор}
```
- оператор **break** используется для выхода из цикла.

### Задания 20-35:

20. Напишите программу, которая 10 раз выводит на экран имя и фамилию. Вид экрана:

```
Иван Петров.
Иван Петров.
Иван Петров.
...
Иван Петров.
```

[решение тут](#)

21. Напишите программу, которая выводит таблицу квадратов первых десяти целых положительных чисел. Вид экрана:

Таблица квадратов

```
-----
Число  Квадрат
-----
1      1
2      4
3      9
...
9      81
10     100
-----
```

[решение тут](#)

22. Напишите программу, которая выводит таблицу квадратов первых пяти целых положительных нечётных чисел. Вид экрана:

```
-----
Таблица квадратов нечетных чисел
-----
Число  Квадрат
-----
1      1
3      9
5      25
7      49
9      81
```

[решение тут](#)

23. Напишите программу, которая выводит таблицу степеней двойки (от нулевой до десятой). Вид экрана:

```
-----
Таблица степеней двойки
-----
0      1
1      2
2      4
...
9      512
10     1024
```

[решение тут](#)

24. Напишите программу, которая вычисляет сумму первых **n** положительных целых чисел. Количество суммируемых чисел вводится во время работы программы. Вид экрана:

```
Вычисление суммы положительных чисел
Введите количество суммируемых чисел → 20
Сумма первых 20 положительных целых чисел равна 210.
```

[решение тут](#)

25. Напишите программу, для вычисления суммы первых **n** целых положительных четных чисел. Вид экрана:

Вычисление суммы положительных четных чисел  
 Введите количество суммируемых чисел -> **12**  
 Сумма первых 12 положительных четных чисел равна 156.

[решение тут](#)

26. Напишите программу, вычисления суммы **n** (вводится с клавиатуры) членов ряда 1, 3, 5, 7, ... . Вид экрана:

Вычисление суммы положительных нечетных чисел  
 Введите количество суммируемых чисел -> **15**  
 Сумма первых 15 нечетных чисел равна 225.

[решение тут](#)

27. Напишите программу, вычисления суммы **n** (вводится с клавиатуры) членов ряда 1, 1/2, 1/3, 1/4, ... . Вид экрана:

Вычисление суммы членов ряда 1, 1/2, 1/3, 1/4, ...  
 Введите количество суммируемых чисел -> **5**  
 Сумма первых 5 членов ряда 1, 1/2, 1/3, 1/4, ... равна 2.2833333333333333.

[решение тут](#)

28. Напишите программу, которая вычисляет факториал числа **n**, введенного с клавиатуры (факториал – произведение целых чисел от **1** до **n**). Вид экрана:

Вычисление факториала числа  
 Введите число -> **8**  
 Факториал числа 8 равен 40320.

[решение тут](#)

29. Напишите программу, которая выводит таблицу значений функции  $y = -2,4x^2 + 5x - 3$  в диапазоне от -2 до +2 с шагом 0,5. Вид экрана:

Таблица значений функции  $y = -2,4x^2 + 5x - 3$

```
-----
-2.00 -1.50 -1.00 -0.50  0.00  0.50  1.00  1.50  2.00
-22.60 -15.90 -10.40 -6.10 -3.00 -1.10 -0.40 -0.90 -2.60
-----
```

[решение тут](#)

30. Напишите программу, которая вычисляет среднее арифметическое введенных с клавиатуры 5 дробных чисел. Вид экрана:

Обработка последовательности дробных чисел  
 После ввода каждого числа нажимайте <Enter>  
 Ввод очередного числа -> **1.2**  
 Ввод очередного числа -> **5.4**  
 Ввод очередного числа -> **7.16**  
 Ввод очередного числа -> **3**  
 Ввод очередного числа -> **15.9**  
 Введено чисел: 5 Сумма: 32.66 Среднее арифметическое: 6.53

[решение тут](#)

31. Напишите программу, для вычисления среднего арифметического введенных с клавиатуры дробных чисел. Количество чисел вводится с клавиатуры. Вид экрана:

Обработка последовательности дробных чисел  
 Введите количество чисел -> **4**  
 После ввода каждого числа нажимайте <Enter>  
 Ввод очередного числа -> **8.1**  
 Ввод очередного числа -> **10**  
 Ввод очередного числа -> **6.7**  
 Ввод очередного числа -> **4.2**  
 Введено чисел: 4 Сумма: 29.00 Среднее арифметическое: 7.25

[решение тут](#)

32. Напишите программу, которая после ввода с клавиатуры каждого числа последовательности пяти дробных чисел, выводит среднее арифметическое введенных чисел. Вид экрана:

Обработка последовательности дробных чисел  
 После ввода каждого числа нажимайте <Enter>  
 Ввод очередного числа -> **12.3**  
 Введено чисел: 1 Сумма: 12.30 Среднее арифметическое: 12.30  
 Ввод очередного числа -> **15**  
 Введено чисел: 2 Сумма: 27.30 Среднее арифметическое: 13.65  
 Ввод очередного числа -> **22.17**  
 ...

[решение тут](#)

33. Напишите программу, которая вычисляет среднее арифметическое последовательности дробных чисел (количество чисел и числа вводятся с клавиатуры). Программа также должна вывести максимальное и минимальное числа последовательности. Вид экрана:

Обработка последовательности дробных чисел  
 Введите количество чисел -> **3**  
 После ввода каждого числа нажимайте <Enter>  
 Ввод очередного числа -> **1.24**  
 Ввод очередного числа -> **6.62**  
 Ввод очередного числа -> **4.9**  
 Введено чисел: 3. Среднее арифметическое: 4.25  
 Максимальное число: 6.62. Минимальное число: 1.24

[решение тут](#)

34. Напишите программу проверки знания таблицы умножения чисел от 1 до 10 на случайное число **m** (от 2 до 9). Программа должна вывести 10 примеров и выставить оценку: за 10 и 9 правильных ответов – "отлично", за 8 и 7 – "хорошо", за 6 и 5 – "удовлетворительно", за 4 и менее – "плохо".

Проверка знания таблицы умножения на 6  
Введите правильный ответ и нажмите <Enter>

1 x 6 = **6**  
2 x 6 = **12**  
3 x 6 = **16**

...

У Вас 8 правильных ответов. Ваша оценка - хорошо.

[решение тут](#)

35. Напишите программу проверки умения складывать и вычитать числа в пределах 100. Программа должна вывести 10 случайных примеров, причём в примерах уменьшаемое не должно быть меньше вычитаемого, т.е. не допускается предлагать испытуемому примеры с отрицательным результатом. Вид экрана:

Проверка умения складывать и вычитать числа  
Введите правильный ответ и нажмите <Enter>

4 + 15 = **19**  
99 - 80 = **19**  
78 - 19 = **61**  
65 + 15 = **70**

...

32 - 19 = **13**

У Вас 7 правильных ответов. Ваша оценка - хорошо.

[решение тут](#)

### Циклы Repeat

- цикл **repeat** – это цикл с постусловием, инструкции которого выполняются хотя бы один раз, и до тех пор, пока стоящее после слова **until** условие *завершения цикла* ложно (т.е. в теле цикла **repeat** должны быть инструкции, которые влияют на условие завершения):

```
repeat
  read(a);
  s:= s+a;
until s > 100;
```

### Задания 36-40:

36. Напишите программу, вычисляющую сумму и среднее арифметическое последовательности положительных чисел, которые вводятся с клавиатуры. Вид экрана:

Вычисление среднего арифметического последовательности положительных чисел  
Числа вводите после стрелки.

Для завершения ввода введите ноль.

-> **45**  
-> **23**  
-> **15**  
-> **0**

Введено чисел: 3

Сумма чисел: 83

Среднее арифметическое: 27.67

[решение тут](#)

37. Напишите программу, которая определяет максимальное число из введенной с клавиатуры последовательности положительных чисел (длина последовательности не ограничена):

Определение максимального числа последовательности положительных чисел.

Числа вводите после стрелки.

Для завершения ввода введите ноль.

-> **56**  
-> **75**  
-> **43**  
-> **0**

Максимальное число: 75

[решение тут](#)

38. Напишите программу, которая проверяет, является ли целое число, введенное пользователем, простым. Вид экрана:

Введите целое число и нажмите <Enter> -> **45**

45 - не простое число

[решение тут](#)

39. Напишите программу для определения минимального количества (**n**) целых положительных чисел, сумма которых не менее числа **m** (**m** вводится с клавиатуры). Вид экрана:

Определение минимального количества целых положительных чисел,

сумма которых не менее (введите число) -> **100**

Сумма первых 14 целых положительных чисел не менее 100 и составляет 105.

[решение тут](#)

40. Напишите программу, которая "задумывает" число в диапазоне от 1 до 10 и предлагает пользователю угадать число за 5 попыток. Вид экрана:



Игра "Угадай число".  
 Компьютер "задумал" число от 1 до 10.  
 Угадайте его за 5 попыток.  
 Введите число и нажмите <Enter>  
 -> 5  
 -> 3  
 Поздравляем! Вы выиграли за 2 попытки!

[решение тут](#)

### Циклы While

- выполнение инструкций цикла **while** продолжается до тех пор, пока условие, записанное после слова **while**, истинно;
- цикл **while** – это цикл с предусловием, т.е. инструкции тела цикла могут быть вообще не выполнены;

```
while n < 100 do
begin
    writeln(n);
    n:= n + 1;
end;
```

### Задания 41-44:

41. Напишите программу для определения минимального количества (**n**) членов ряда 1, 3, 5, 7, ..., сумма которых не менее числа **m** (**m** вводится с клавиатуры). Вид экрана:

Определение минимального количества членов ряда 1, 3, 5, 7, ..., сумма которых не менее (введите число) -> 101  
 Сумма первых 11 членов ряда 1, 3, 5, 7, ... не менее 101 и составляет 121.

[решение тут](#)

42. Напишите программу для определения минимального количества (**n**) членов ряда 1, 1/2, 1/3, 1/4, ..., сумма которых составляет не менее 5. Вид экрана:

Определение минимального количества чисел  
 Сумма первых 83 членов ряда 1, 1/2, 1/3, 1/4, ... не менее 5 и составляет 5.00206827268017.

[решение тут](#)

43. Напишите программу, которая вычисляет  $\pi$  с заданной пользователем точностью. Для этого воспользуйтесь тем, что значение суммы ряда  $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$  при с достаточно большом количестве членов приближается к  $\pi/4$ . Вид экрана:

Вычисление приближенного значения  $\pi$   
 Задайте точность вычисления  $\pi$  -> 0.001  
 Значение числа  $\pi$  с точностью 0.001 равно 3.14259165433954.  
 Просуммировано 1001 членов ряда.

[решение тут](#)

44. Напишите программу, которая вычисляет наибольший общий делитель двух целых чисел. Вид экрана:

Определение наибольшего общего делителя для двух целых чисел  
 Введите в одной строке два числа и нажмите <Enter> -> 55 40  
 Наибольший общий делитель чисел 55 и 40 равен 5.

[решение тут](#)

### Символы и строки

- каждому символу соответствует число – код символа. Символ по его коду **n** (в кодировке Windows) можно получить функцией **chr(n)**, а код символа **sym** – функцией **ord(sym)**;
- получить доступ к отдельному символу строки можно, указав его номер в квадратных скобках после имени переменной (строки). Функция **pos(subs, s)** возвращает позицию подстроки **subs** в строке **s**, а **length(s)** – длину строки **s**;
- оператор **+** выполняет конкатенацию (слияние) строк, а операция **s += s2** добавляет в конец строки **s** строку **s2**;
- для преобразования (целого или вещественного) числа в строковое представление можно использовать функции **inttostr(i)**, **floattostr(n)**, а для обратных преобразований – **strtoint(s)**, **strtofloat(s)**;
- функции **leftstr(s, n)**, **rightstr(s, n)**, **copy(s, i, n)** возвращают **n** символов строки **s** слева, справа или с позиции **i** соответственно;
- процедура **insert(subs, s, i)** вставляет подстроку **subs** в строку **s** с позиции **i**, а **delete(s, i, n)** удаляет из строки **s** **n** символов с позиции **i**;
- удаление начальных и конечных пробелов (строки **s**) производится функциями **trim(s)**, **trimleft(s)**, **trimright(s)**;
- процедура **sleep(ms)** делает паузу на **ms** миллисекунд.

### Задания 45-54:

45. Напишите программу, которая выдает код символа, введенного пользователем. Программа должна завершать работу после ввода точки. Вид экрана:

Введите символ и нажмите <Enter>  
 (Для завершения работы программы введите точку).  
 -> А  
 Символ: А, код: 192  
 -> б  
 Символ: б, код: 225  
 -> .

[решение тут](#)

46. Напишите программу, которая выводит на экран символы, имеющие коды от 128 до 255 (таблица должна состоять из восьми колонок и шестнадцати строк). Фрагмент вывода на экран:

...  
 ё - 184 № - 185 е - 186 » - 187 j - 188 S - 189 s - 190 i - 191  
 A - 192 B - 193 B - 194 Г - 195 Д - 196 Е - 197 Ж - 198 З - 199  
 ...

[решение тут](#)

47. Напишите программу, которая выводит на экран сообщение в «телеграфном» стиле, т.е. буквы сообщения должны появляться по одной, причём с некоторой задержкой.

[решение тут](#)

48. Напишите программу, которая во введённой с клавиатуры строке преобразует строчные буквы русского алфавита в прописные. Вид экрана:

Введите текст на русском языке длиной не более 80 символов:

-> **изучив Паскаль, можно приступить к изучению Си**

Строка, преобразованная к верхнему регистру:

ИЗУЧИВ ПАСКАЛЬ, МОЖНО ПРИСТУПИТЬ К ИЗУЧЕНИЮ СИ.

[решение тут](#)

49. Напишите программу, которая удаляет начальные пробелы из введённой с клавиатуры строки (не используя функции **trim**, **trimleft**, **trimright**). Вид экрана:

Удаление начальных пробелов строки

Введите строку:

**Программа удаления начальных пробелов**

Строка без начальных пробелов:

Программа удаления начальных пробелов

[решение тут](#)

50. Напишите программу, которая проверяет, является ли введённая с клавиатуры строка двоичным числом. Вид экрана:

Введите число и нажмите <Enter>

-> **1011201211220**

Введённая строка не является двоичным числом.

[решение тут](#)

51. Напишите программу, которая проверяет, является ли введённая с клавиатуры строка шестнадцатеричным числом. Вид экрана:

Введите число и нажмите <Enter>

-> **10983ACF5**

Введённая строка является шестнадцатеричным числом.

[решение тут](#)

52. Напишите программу, которая преобразует введённое с клавиатуры десятичное двоичное число в десятичное. Вид экрана:

Введите двоичное число (не более 30 разрядов - знаки после 30 разряда будут отброшены!)

-> **10111111010000011001**

Двоичному числу 10111111010000011001 соответствует десятичное 783385.

[решение тут](#)

53. Напишите программу, которая преобразует введённое пользователем десятичное число в число в указанной системе счисления (от 2-х до 10-ти). Вид экрана:

Введите целое число -> **25**

Введите основание системы счисления -> **3**

Число, полученное после преобразования: 221

[решение тут](#)

54. Напишите программу, которая преобразует введённое пользователем десятичное число в шестнадцатеричное. Вид экрана:

Введите десятичное число для преобразования в шестнадцатеричную систему счисления -> **201**

Десятичному числу 201

соответствует шестнадцатеричное число C9

[решение тут](#)

### Массивы

- массив представляет собой совокупность индексированных (номерованных) элементов одного типа;
- массивы могут быть одномерными и многомерными;
- массив объявляется в разделе объявления переменных, в массиве могут использоваться также именованные константы.

примеры объявлений одномерных массивов:

**const**

dn: **array**[1..7] **of string**[11]=('Понедельник',... , 'Воскресенье');

**var**

t: **array**[1..7] **of real**;

sym: **array**['A'..'Z'] **of integer**;

пример объявления трехмерного массива:

**var**

n: **array**[1..10, 1..8, 3..7] **of byte**;

- доступ к элементу массива осуществляется путём указания индекса(ов) [номера(ов)] элемента  
*например:* dn[5] - будет 'Пятница');
- индексами элементов массива могут быть переменные порядковых (целого или символьного) типов.

### Задания 55-68:

55. Напишите программу, которая выводит количество ненулевых элементов вводимого массива из 5 целых чисел. На экране должна появляться подсказка с номером элемента:



Введите элементы массива (5 целых чисел): a[1] -> **12**  
a[2] -> **28**  
a[3] -> **3**  
a[4] -> **0**  
a[5] -> **55**  
В массиве 4 ненулевых элемента.

[решение тут](#)

56. Напишите программу, которая выводит минимальный элемент введенного в одной строке с клавиатуры массива 5 целых чисел. Вид экрана:

Поиск минимального элемента массива из 5 целых чисел.  
Введите в одной строке элементы массива: -> **23 0 45 -5 12**  
Минимальный элемент массива: -5.

[решение тут](#)

57. Напишите программу, которая выводит максимальный элемент введенного в одной строке с клавиатуры массива 5 целых чисел. Вид экрана:

Поиск максимального элемента массива.  
Введите в одной строке элементы массива (5 целых чисел)  
-> **-1 2 8 5 4**  
Максимальный элемент массива: 8.

[решение тут](#)

58. Напишите программу, которая вычисляет сумму, количество и среднее арифметическое ненулевых элементов введенного с клавиатуры массива 10 целых чисел. Вид экрана:

Введите в одной строке элементы массива (10 целых чисел):  
-> **2 11 3 6 0 9 5 10 7 7**  
Сумма элементов массива: 60  
Количество ненулевых элементов: 9  
Среднее арифметическое ненулевых элементов: 6.67

[решение тут](#)

59. Напишите программу, которая вычисляет среднюю (за неделю) температуру воздуха. Исходные данные вводятся во время работы программы. Вид экрана:

Введите температуру воздуха по дням недели:  
Понедельник -> **12**  
Вторник -> **9**  
...  
Воскресенье -> **14**  
Средняя температура за неделю: XX.X град.

[решение тут](#)

60. Напишите программу, которая проверяет, находится ли в массиве введенное с клавиатуры число. Массив должен вводиться во время работы программы. Вид экрана:

Поиск в массиве.  
Введите 10 целых чисел в одной строке через пробел  
-> **3 5 4 8 2 0 9 11 8 7**  
Введите образец для поиска (целое число) -> **8**  
Совпадение образца с элементом номер 4.

[решение тут](#)

61. Напишите программу, которая проверяет, представляют ли элементы введенного с клавиатуры массива возрастающую последовательность. Вид экрана:

Проверка упорядоченности массива по возрастанию.  
Введите в одной строке элементы массива (10 целых чисел):  
-> **2 3 4 5 6 8 10 11 9 12**  
Массив не упорядочен по возрастанию.

[решение тут](#)

62. Напишите программу, которая вычисляет, сколько раз введенное с клавиатуры число встречается в массиве. Вид экрана:

Введите массив - 10 целых чисел в одной строке.  
-> **5 9 1 3 2 5 55 14 3 51**  
Введите образец для поиска -> **5**  
Число 5 встречается в массиве 2 раз(а).

[решение тут](#)

63. Напишите программу, которая проверяет, есть ли во введенном с клавиатуры массиве элементы с одинаковым значением. Вид экрана:

Введите в одной строке элементы массива (10 целых чисел):  
-> **7 15 21 1 3 4 3 2 1 1**  
В массиве обнаружено 4 вариант(а,ов) совпадений.

[решение тут](#)

64. Напишите программу, которая сортирует введенный с клавиатуры одномерный массив по возрастанию. Вид экрана:

Сортировка массива по возрастанию.  
Введите в одной строке 10 действительных чисел:  
-> **1.2 4 5.6 7.7 9 17.4 3.6 99 51.5 23.8**  
Отсортированный массив: 1.2 3.6 4 5.6 7.7 9 17.4 23.8 51.5 99

[решение тут](#)

65. Напишите программу, которая сортирует по убыванию введенный с клавиатуры одномерный массив. Вид экрана:

Сортировка массива по убыванию методом "пузырька".  
Введите в одной строке 10 действительных чисел:  
-> **1 2.5 4 5.6 7.7 9 17.4 3.6 99 51.5**  
Отсортированный массив: 99 51.5 17.4 9 7.7 5.6 4 3.6 2.5 1

[решение тут](#)

66. Напишите программу, которая определяет средний рост в классе, и количество учеников, чей рост превышает средний. Вид экрана:

Анализ роста учеников

Введите рост в см (для завершения введите 0):

-> 165

-> 172

-> 168

...

-> 164

-> 180

-> 0

Средний рост: 167.8 см.

у 8 учеников рост превышает средний.

[решение тут](#)

67. Напишите программу, которая вводит с клавиатуры (по строкам) двумерный массив и вычисляет сумму его элементов по строкам. Вид экрана:

Введите массив (строк – 3, в каждой строке 5 целых чисел):

1-я строка массива -> 5 9 6 4 7

2-я строка массива -> 1 3 5 2 6

...

Сумма элементов 2-й строки массива: 17

Сумма элементов 3-й строки массива: 32

[решение тут](#)

68. Напишите программу, которая вводит с клавиатуры двумерный массив, вычисляет сумму его элементов по столбцам и выводит данные в виде таблицы.

Введите массив (строк – 3, в каждой строке 5 целых чисел):

1-я строка массива -> 2 5 3 8 9

2-я строка массива -> 5 4 6 2 9

3-я строка массива -> 8 4 6 7 3

1-я строка	2	5	3	8	9
2-я строка	5	4	6	2	9
3-я строка	8	4	6	7	3

Сумма	15	13	15	17	21
-------	----	----	----	----	----

[решение тут](#)

### Функции

- функция – подпрограмма пользователя, результатом которой является только одно значение;
- для передачи данных в функцию надо использовать только параметры (переменные, объявленные вне функции, применять не рекомендуется);  
тип каждого фактического параметра (константы или переменной) в инструкции вызова функции должен соответствовать типу формального параметра, указанного при объявлении функции;

оператор **exit** предназначен для досрочного завершения процедуры или функции;

- вызов **exit** в разделе операторов основной программы приводит к её немедленному завершению;
- если в инструкции объявления функции перед именем формального параметра стоит слово **var**, то формальным параметром может быть только переменная (можно использовать для возврата результата в программу), без слова **var** – в качестве формального параметра может использоваться константа или переменная;
- среди выполняемых инструкций функции обязательно должна быть инструкция присваивания значения функции. Пример записи функции:

```
function AreaRectangle(a,b: real): real; {площадь прямоугольника}
begin
    AreaRectangle:= a*b;
end;
```

### Задания 69-75:

69. Напишите функцию, которая вычисляет объём цилиндра. Параметрами функции должны быть диаметр и высота цилиндра. Вид экрана при использовании функции:

Вычисление объёма цилиндра

Введите в одной строке диаметр и высоту цилиндра -> 7 11

Объём цилиндра составляет: 423.329610071225

[решение тут](#)

70. Напишите функцию, которая возвращает большее из двух чисел, полученных в качестве аргумента. Вид экрана:

Введите два числа -> 87.2 87

Максимальное из двух чисел: 87.2

[решение тут](#)

71. Напишите функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде одного из знаков: >, < или =. Вид экрана:

Введите два числа -> 55 44

Результат сравнения: >

[решение тут](#)

72. Напишите функцию, которая вычисляет значение  $a^b$ . Числа  $a$  и  $b$  – действительные. Вид экрана:

Введите число и показатель степени: 15.3 3.5

15.300 в степени 3.500 = 14009.415

[решение тут](#)

73. Напишите функцию, которая возвращает преобразованную к верхнему регистру строку, полученную в качестве аргумента. Вид экрана:



### Файлы

- для вывода данных в файл или чтения из файла, нужно объявить файловую переменную типа **text** (например, **f: text**). Связь переменной с файлом осуществляется инструкцией **assign**;
- для доступности файл надо открыть (для чтения с помощью инструкции **reset**, записи – **rewrite**, добавления – **append**). Запись в файл выполняется инструкцией **write (writeln)**, чтение – **read (readln)**, в качестве первого параметра инструкций нужно указать файловую переменную. По завершении работы с файлом его нужно закрыть инструкцией **close**;
- функция **fileexists(f)** определяет существует ли файл **f**, а **eof(f)** и **eoln(f)** – достигнут ли конец файла или строки.

Пример:

```
assign(f, 'a:\text.txt');
rewrite(f);
writeln(f, n);
close(f);
```

### Задания 82-88:

82. Напишите программу, которая в текущем каталоге создает файл «num.txt» и записывает в него 5 введенных с клавиатуры целых чисел. Вид экрана:

```
Запись данных в текстовый файл
Введите пять целых чисел.
После ввода каждого числа нажимайте <Enter>
-> 11
-> 15
-> 16
-> 8
-> 22
```

Введенные числа записаны в файл num.txt

[решение тут](#)

83. Напишите программу, которая дописывает в файл «num.txt» 5 введенных с клавиатуры целых чисел. Вид экрана:

```
Добавление в файл "num.txt"
Введите пять целых чисел для добавления.
После ввода каждого числа нажимайте <Enter>
-> 24
-> 17
-> 30
-> 8
-> 3
```

Введенные числа добавлены в файл "num.txt"

[решение тут](#)

84. Напишите программу, которая выводит на экран содержимое файла «num.txt».

Вид экрана:

```
Содержимое файла "num.txt"
11
15
16
...
8
3
```

[решение тут](#)

85. Напишите программу, которая вычисляет среднее арифметическое чисел, находящихся в файле «num.txt». Вид экрана:

```
Вычисление среднего арифметического чисел, находящихся в файле "num.txt"
Прочитано чисел: 10
Сумма чисел: 154
Среднее арифметическое: 15.40
```

[решение тут](#)

86. Напишите программу, которая дописывает в файл «phone.txt» фамилию, имя и номер телефона одного абонента. Если файла нет, то программа должна создать его. Данные отдельного абонента должны находиться в одной строке. Вид экрана:

```
Добавление абонента в телефонный справочник
Фамилия -> Сидоров
Имя -> Вася
Телефон -> 234-84-37
Информация по абоненту записана в справочник.
```

[решение тут](#)

87. Напишите программу, используя которую можно за один сеанс работы добавить информацию в файл «phone.txt» о нескольких абонентах. Если файла нет, то программа должна создать его. Данные отдельного абонента должны находиться в одной строке. Вид экрана:

Добавление в телефонный справочник  
 Фамилия (<Enter> без ввода данных завершает работу) ->

**Иванов**  
 Имя -> **Вася**  
 Телефон -> **5-12-45**  
 Информация добавлена.  
 ...  
 Фамилия -> **Иванов**  
 Имя -> **Петя**  
 Телефон -> **7-43-76**  
 Информация добавлена.  
 Фамилия ->  
 Работа программы завершена. Добавлено 3 записей.

[решение тут](#)

88. Напишите программу, которая позволяет найти нужные сведения в телефонном справочнике (файл "phone.txt"). Программа должна запрашивать фамилию человека и выводить его телефон. Если в справочнике есть одинаковые фамилии, то программа должна вывести список всех людей, имеющих эти фамилии. Вид экрана:

Поиск в телефонном справочнике  
 Введите фамилию для поиска -> **Петров**  
 Сведений о Петров нет.  
 Вид экрана при другом запросе:  
 Поиск в телефонном справочнике  
 Введите фамилию для поиска -> **Иванов**  
 Иванов Вася 5-12-45  
 Иванов Петя 7-43-76  
 По запросу Иванов найдено 2 записей.

[решение тут](#)

### Рекурсия

- рекурсивной называется процедура или функция, которая может вызывать сама себя;
- для завершения процесса рекурсии в алгоритме рекурсивной функции (процедуры) обязательно должна быть веточка, обеспечивающая непосредственное завершение функции (процедуры).

Пример:

```
function sum(n:integer):real; {рекурсивная функция вычисления}
begin {суммы n членов ряда 1, 1/2, 1/3,..., 1/n}
  if n=1 then sum:=1 {условие, завершения функции}
  else sum:=sum(n-1)+1/n;
end;
```

### Задания 89-93:

89. Напишите рекурсивную функцию вычисления суммы первых **n** натуральных чисел. Вид экрана вывода при использовании функции:  
 Вычисление суммы первых n натуральных чисел  
 Введите количество чисел: **11**  
 Сумма первых 11 натуральных чисел - 66  
[решение тут](#)
90. Напишите рекурсивную функцию вычисления факториала. Вид экрана вывода:  
 Введите число: **7**  
 Факториал числа 7 - 5040  
[решение тут](#)
91. Напишите рекурсивную функцию вычисления суммы всех нечётных чисел от **1** до **n** (включительно). Вид экрана вывода:  
 Вычисление суммы нечётных чисел от 1 до n  
 Введите число n: **12**  
 Сумма нечётных чисел от 1 до 12 - 36  
[решение тут](#)
92. Напишите рекурсивную функцию вычисления суммы всех чисел от **1** до **n**, кратных **k**. Вид экрана вывода:  
 Вычисление суммы чисел от 1 до n, кратных k  
 Введите через пробел числа n и k: **14 3**  
 Сумма чисел от 1 до 14, кратных 3 - 30  
[решение тут](#)
93. Напишите программу, которая вычисляет сопротивление электрической цепи,

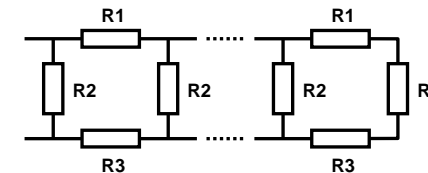


схема которой приведена ниже. Порядок цепи (количество сопротивлений R2) и величины сопротивлений вводятся с клавиатуры.

Вид экрана вывода:

Вычисление сопротивления электрической цепи  
 Введите через пробел значения r1, r2, r3: **1.4 4.8 1.0**  
 Введите порядок цепи: **3**  
 Сопротивление цепи: 1.80

[решение тут](#)

### Множества

- **множество** представляет собой набор однотипных элементов, каждый из которых может входить во множество не более одного раза;





вид экрана для другого слова:

Введите слово: **berlin**

Введённое слово не является именем собственным.

или:

Введите слово: **BERLIN**

Введённое слово не является именем собственным.

[решение тут](#)

102. Составьте функцию, которая определяет, может ли являться введённое слово, состоящее из букв русского алфавита, именем собственным (для имен собственных первая буква должна быть заглавной, но не может быть буквой 'Ы', 'Ь' или 'Ъ', а последующие буквы должны быть строчными). Вид экрана:

Введите слово: **Амур**

Слово "Амур" может быть именем собственным.

вид экрана для другого слова:

Введите слово: **енисей**

Введённое слово не является именем собственным.

или:

Введите слово: **Ыртыш**

Введённое слово не является именем собственным.

[решение тут](#)

### Графика

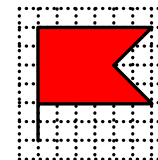
- в графическом режиме экран представляет собой точки, которые могут быть окрашены в разные цвета (для инициализации графического режима используется модуль **Graph (uses GraphABC)**; координаты точек возрастают слева направо и сверху вниз;
- для вывода на экран графики используются примитивы: линия – `Line(x1, y1, x2, y2)`, прямоугольник – `Rectangle(x1, y1, x2, y2)`, окружность – `Circle(x, y, r)`, эллипс – `Ellipse(x1, y1, x2, y2)`, скругленный прямоугольник – `RoundRect(x1, y1, x2, y2, w, h)`, дуга окружности – `Arc(x, y, r, a1, a2)`, сектор окружности – `Pie(x, y, r, a1, a2)`;
- для заливки областей используется процедура `FloodFill(x, y, color)`, прямоугольника – `FillRect(x1, y1, x2, y2)`; для установки стиля заливки – `SetBrushStyle(bs)`, текущего цвета – `SetBrushColor(col)`, цвет можно задать функцией `RGB(R, G, B)` или константами;
- текст выводится процедурой `TextOut(x, y, string)`.

Пример вывода окружности:

```
uses GraphABC;
begin
  Circle(75, 60, 25);
end.
```

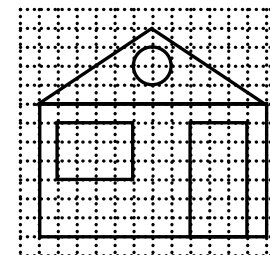
### Задания 103-116:

103. Напишите программу, которая рисует на экране флажок красного цвета:



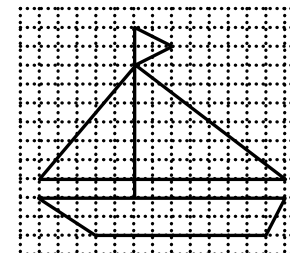
[решение тут](#)

104. Напишите программу, которая вычерчивает на экране домик:



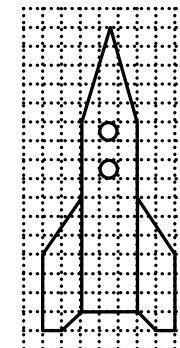
[решение тут](#)

105. Напишите программу, которая вычерчивает на экране кораблик:



[решение тут](#)

106. Напишите программу, которая вычерчивает на экране ракету:

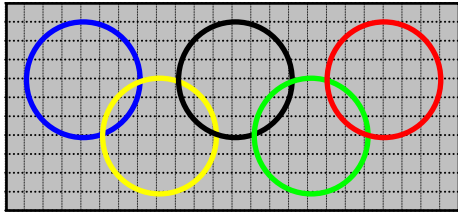


[решение тут](#)

107. Напишите программу, выводящую на экран Государственный (трехцветный) флаг России.

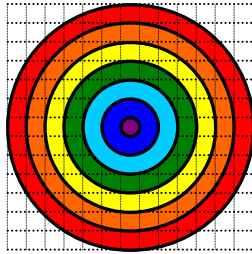
[решение тут](#)

108. Напишите программу, которая рисует флаг Олимпийских игр:



[решение тут](#)

109. Напишите программу, которая выводит на экран рисунок, изображенный ниже:



[решение тут](#)

110. Напишите программу, которая рисует окружность, движущуюся по экрану.

[решение тут](#)

111. Напишите программу, которая выводит на экран движущийся кораблик (см. задание 105).

[решение тут](#)

112. Напишите программу, которая выводит на экран улетающую ракету (см. задание 106).

[решение тут](#)

113. Напишите программу, которая рисует смайлика, приведенного ниже:



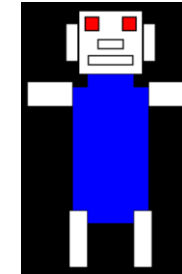
[решение тут](#)

114. Напишите программу, которая рисует снеговика на черном фоне:



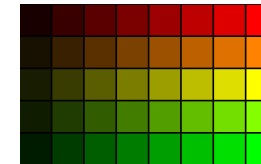
[решение тут](#)

115. Напишите программу, которая рисует робота на черном фоне:



[решение тут](#)

116. Напишите программу, которая выводит нижеприведенную палитру:



[решение тут](#)

### Записи

- для обработки комбинации данных разных типов в Pascal применяется комбинированный тип данных – **запись**. *Запись – это структурированный тип данных, состоящий из фиксированного числа компонентов одного или нескольких типов, называемых **полями записи**.*

- при работе с записями сначала объявляется **запись**:

```

type
    ИмяТипа = record
        ИмяПоля1: ТипПоля1;
        ИмяПоля2: ТипПоля2;
        ...
        ИмяПоляN: ТипПоляN;
    end;
    
```

затем объявляются **переменные** соответствующих типов, входящих в **запись**.

- обращение к значению поля осуществляется с помощью *имени переменной* и *имени поля*, разделенными точкой:  
ПеременнаяТипаЗапись .ИмяПоляN
- при присвоении переменной типа запись значения другой переменной такого же типа, всем полям второй переменной будут присвоены значения первой;
- при обращении к полям записи можно использовать оператор **with**. Один раз указав переменную типа запись в операторе **with**, можно работать с именами полей как с обычными переменными:

```
with ПеременнаяТипаЗапись do
begin
    ПеременнаяТипаЗапись.ИмяПоля1:= ЗначениеПоля1;
    ...
    ПеременнаяТипаЗапись.ИмяПоляN:= ЗначениеПоляN;
end;
```

### Задания 117-120:

117. Напишите программу с использованием записи "автомобили" ('avto') с полями "марка" ('marka') "номер" ('number') "фамилия\_инициалы" ('fio'), в которой полям переменной *a* присваиваются нужные значения, затем переменной *b* присваивается значение переменной *a* и выводится на экран. Вид экрана:

```
Марка и номер автомобиля: LADA, д567ом161rus
Фамилия и инициалы владельца автомобиля: Иванов П.Д.
```

[решение тут](#)

118. Напишите программу с использованием записи "автомобили" (см. пред. задан.), в которой значения полей элементов массива *a* вводятся с клавиатуры. После ввода данных перечень автомобилей выводится на экран. Вид экрана:

```
Введите данные по 1-му автомобилю:
Марка автомобиля: Волга
Номер автомобиля: o124моb1rus
Фамилия и инициалы владельца автомобиля: Яковлев И.Г.
...
Введите данные по 5-му автомобилю:
Марка автомобиля: Лада
Номер автомобиля: p555oc161rus
Фамилия и инициалы владельца автомобиля: Калинин В.С.
1-й автомобиль: o124моb1rus, Волга, Яковлев И.Г.
...
5-й автомобиль: p555oc161rus, Лада, Калинин В.С.
```

[решение тут](#)

119. Напишите программу с использованием записи "школьники" ('schoolboy') с полями "фамилия" ('surname') "имя" ('name') "школа" ('school') "класс" ('klass'), в которой значения полей элементов массива вводятся с клавиатуры.

После ввода данных, список выводится на экран. Вид экрана:

```
Введите фамилию 1-го ученика (цы): Васютин
Введите имя 1-го ученика (цы): Илья
Введите через пробел № школы, класс 1-го ученика (цы): 23 8
Введите фамилию 2-го ученика (цы): Михайлова
Введите имя 2-го ученика (цы): Олеся
Введите через пробел № школы, класс 2-го ученика (цы): 15 9
...
Список учащихся:
1. Васютин Илья 23 8
2. Михайлова Олеся 15 9
...
```

[решение тут](#)

120. Напишите программу с использованием записи "школьники" (см. пред. задан.), которая, после ввода данных с клавиатуры, выводит список учащихся в порядке возрастания номеров школ:

```
Введите фамилию 1-го ученика (цы): Васютин
Введите имя 1-го ученика (цы): Илья
Введите через пробел № школы, класс 1-го ученика (цы): 23 8
...
Введите фамилию 15-го ученика (цы): Иванов
Введите имя 15-го ученика (цы): Сергей
Введите через пробел № школы, класс 15-го ученика (цы): 1 11
...
Список учащихся:
1. Иванов Сергей 1 11
...
20. Васютин Илья 23 8
...
```

[решение тут](#)

### Динамические массивы

- Динамический массив объявляется следующим образом:
  - a: **array of** тип элементов; {одномерный массив}
  - b: **array [,] of** тип элементов; {двумерный массив}
 и т.д.
- динамический массив\* нуждается в инициализации (выделении памяти под элементы), для этого используются два способа:

– использованием операции **new**:

```
a:= new integer[5];
b:= new real[4,3];
```

при этом можно задавать значения элементов массива:

```
a := new integer[3] (1, 2, 3);
b := new real[4, 3] ((1, 2, 3), (4, 5, 6), (7, 8, 9), (0, 1, 2));
```

– использованием стандартной процедуры **SetLength**:

```
SetLength(a, 10);
SetLength(b, 5, 3);
```

при использовании процедуры **SetLength**, старое содержимое массива сохраняется.

\* *примечание:* в PascalABC.NET динамические массивы нумеруются с нуля.

- при описании инициализацию динамического массива можно проводить в сокращенной форме:

```
b: array [,] of real := ((1, 2, 3), (4, 5, 6), (7, 8, 9), (0, 1, 2));
```

- длина массива (количество элементов в нем) возвращается стандартной функцией **Length** или свойством **Length**:

```
d := Length(a);
d := a.Length;
```

- динамические массивы одного типа можно присваивать друг другу, при этом обе переменные-ссылки будут указывать на одну память:

```
a1 := a2;
```

- чтобы одному динамическому массиву присвоить копию другого массива, следует воспользоваться стандартной функцией **Copy**:

```
a1 := Copy(a2);
```

### Задания 121-126:

121. Напишите программу с использованием динамического массива, которая выводит на экран определенное количество (вводится с клавиатуры) случайных целых чисел. Вид экрана:

```
Введите число элементов массива: 24
Динамический массив из 24 случайных целых чисел:
7 4 8 4 6 1 9 9 7 8 9 2 2 7 6 3 6 5 8 2 4 2 8 1
```

[решение тут](#)

122. Напишите программу с использованием динамических массивов **a** и **b**, где массив **a** присваивается массиву **b**, затем некоторым элементам массива **b** задаются новые значения (Программа должна вывести на экран массив **a** в первоначальном и измененном виде). Вид экрана:

```
Введите число элементов массива "a": 18
Первоначальный массив "a":
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Введите значения 1-го и 4-го элементов массива "b": 11 44
Измененный массив "a":
11 1 2 44 4 5 6 7 8 9 10 11 12 13 14 15 16 17
```

[решение тут](#)

123. Напишите программу с повторной инициализацией динамического массива стандартной процедурой **SetLength**. (Программа должна вывести на экран заполненный массив после первой и повторной инициализации). Вид экрана:

```
SetLength при повторном вызове сохраняет значения элементов:
Первоначальный массив:
1 2 3 4 5 6 7 8 9 10 11 12
Массив после повторной инициализации:
1 2 3 4 5 6 17 18 19 20 21 22 23 24 25 26 27 28
```

[решение тут](#)

124. Напишите программу с повторными инициализациями динамического массива стандартной процедурой **SetLength**. (Программа должна вывести на экран заполненный массив после каждой инициализации). Вид экрана:

```
Первоначальный массив:
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Массив после повторной инициализации:
20 19 18 17 16 15 14 13 12 11
Массив после третьей инициализации:
20 19 18 17 16 15 14 13 12 11 0 0 0 0 0 0 0 0 0 0
```

[решение тут](#)

125. Напишите программу инициализацией динамического массива операцией **new** повторно. (Программа должна вывести на экран заполненный массив в первоначальном виде и после повторной инициализации). Вид экрана:

```
Первоначальный массив:
16 15 14 13 12 11 10 9
Массив после повторной инициализации:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

[решение тут](#)

126. Напишите программу с использованием динамических массивов **a** и **b**, где массиву **b** присваивается копия массива **a**, затем некоторым элементам массивов **a** и **b** задаются новые значения (Программа должна вывести на экран в первоначальном виде массив **a**, и в измененном виде массивы **a** и **b**). Вид экрана:

```
Введите число элементов массива "a": 15
Первоначальный массив "a":
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Введите значения 14-го элемента массива "a": 41
Введите значения 1-го элемента массива "b": 11
Измененный массив "a":
1 2 3 4 5 6 7 8 9 10 11 12 13 41 15
Измененный массив "b":
11 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

[решение тут](#)

127. Напишите программу с процедурой создания динамического массива, в которой число элементов массива вводится с клавиатуры. Вид экрана:

Введите число элементов массива: **22**

Массив со случайными целыми числами:

0 5 1 9 2 8 3 2 2 4 3 3 6 2 8 1 4 7 6 3 9 6

[решение тут](#)

***Литература:***

1. Абрамян М.Э. Электронный задачник по программированию. ЮФУ, 1998-2020.
2. Культин Н.Б. Turbo Pascal в задачах и примерах. СПб.: «БХВ-Петербург», 2006.
3. Рапаков Г.Т., Ржеуцкая С.Ю. Программирование на языке Pascal. СПб.: «БХВ-Петербург», 2004.
4. Эллиот Б., Кофман. Turbo Pascal, 5-е издание. Пер. с англ. – М.: Издательский дом «Вильямс», 2003.