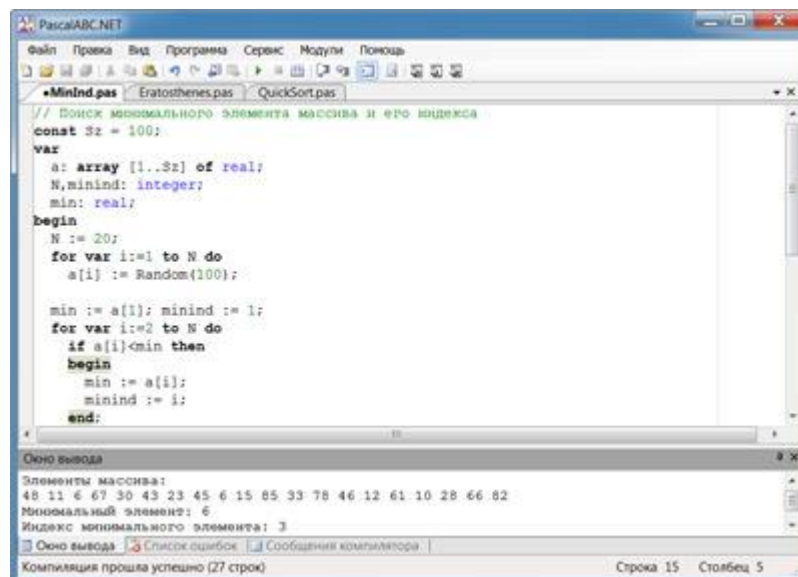


Уинская СОШ

PascalABC.NET

ПРИМЕРЫ ЗАДАЧ



```
// Поиск минимального элемента массива и его индекса
const Sz = 100;
var
  a: array [1..Sz] of real;
  N, minind: integer;
  min: real;
begin
  N := 20;
  for var i:=1 to N do
    a[i] := Random(100);

  min := a[1]; minind := 1;
  for var i:=2 to N do
    if a[i]<min then
      begin
        min := a[i];
        minind := i;
      end;
end;
```

Окно вывода

Элементы массива:
48 11 6 67 30 43 23 45 6 15 85 33 78 46 12 61 10 28 66 82
Минимальный элемент: 6
Индекс минимального элемента: 3

Компиляция прошла успешно (27 строк) Строчка 15 Столбец 5

С сайта
<http://pascalabc.net/>

Оглавление

PascalABC.NET	1
Начало. Тексты программ	6
HelloWorld	6
Вычисление площади круга	6
Вычисление A^8	6
Вычисление расстояния между двумя точками на прямой	6
Вычисление гипотенузы по катетам	7
Бросание кубиков	7
Обмен значений двух переменных с использованием третьей.....	7
Выделение цифр из двузначного числа	8
Начало. Возможные ошибки	8
Неописанная переменная	8
Отсутствующая ;	8
Несовместимость типов при присваивании	8
Отсутствие закрывающего апострофа литеральной строки	9
Ошибки расстановки запятых и апострофов при выводе строк и выражений	9
Ошибка ввода	9
Ошибка неинициализированной переменной	9
Деление на 0	10
Корень из отрицательного числа	10
Условный оператор и оператор выбора. Тексты программ.....	11
Минимум из двух значений	11
Определение четности числа	11
Состоит ли двузначное число из одинаковых цифр	11
Упорядочение двух значений по возрастанию	11
Проверка числа на двузначность	12
Наименование сезона по номеру месяца	12
Определение четверти, в которой находится точка	12
Нахождение корней квадратного уравнения	12
Словесное наименование сезона по номеру месяца	13
Определение того, является ли символ цифрой или буквой	13
Условный оператор и оператор выбора. Возможные ошибки	14
Неверная расстановка begin end.....	14

Отсутствие begin end	14
Неверная расстановка скобок в сложном логическом выражении	14
Циклы. Тексты программ	15
Вывод последовательностей 1 2 3 4 5 и 5 4 3 2 1	15
Вывод последовательности 1 3 5 7 9	15
Сумма и произведение введенных чисел	16
Вычисление $n!$	17
Вычисление A^n	17
Вывод цифр числа	18
Вывод букв английского алфавита	18
Числа Фибоначчи	18
Минимум из введенных	19
Алгоритм Евклида поиска НОД	20
Вывод таблицы умножения	20
Определение простоты числа	21
Циклы. Возможные ошибки	21
Неверный тип переменной цикла	21
Изменение переменной цикла в теле цикла	21
Использование одинаковой переменной цикла в заголовках вложенных циклов	22
Неинициализированная переменная	22
Защипывание	22
Переполнение	22
Сравнение вещественных чисел на равенство	23
Процедуры и функции. Тексты программ	24
Вычисление площади и периметра прямоугольника	24
Табулирование функции	24
Функция вычисления a^n	24
Функция вычисления $n!$	25
Функция, преобразующая цифру к ее словесному наименованию	25
Функция проверки числа на четность	26
Функция определения простоты числа	26
Функция вычисления n -того числа Фибоначчи	26
Функция определения НОД числа	27
Процедура вывода таблицы умножения	27
Процедуры и функции. Возможные ошибки	27

Невозвращение значения функции по одной из ветвей алгоритма	27
Описание переменной цикла, используемого в процедуре, как глобальной	28
Передача выходных параметров по значению	28
Массивы. Тексты программ.....	29
Ввод и вывод	29
Заполнение случайными числами	29
Заполнение арифметической прогрессией.....	29
Заполнение степенями двойки.....	30
Заполнение числами Фибоначчи.....	30
Инвертирование массива	30
Максимальный элемент в массиве и его индекс.....	31
Минимальный четный элемент и его индекс.....	31
Запись четных элементов массива в новый массив	32
Слияние отсортированных массивов в отсортированный	32
Сдвиг элементов влево.....	35
Сдвиг элементов вправо.....	35
Удаление элемента.....	35
Вставка элемента	36
Подсчет количества элементов, удовлетворяющих условию.....	36
Есть ли элемент, удовлетворяющий условию	37
Сортировка пузырьком.....	38
Сортировка выбором	38
Массивы. Возможные ошибки	39
Попытка вывести массив не поэлементно, а целиком.....	39
Использование записи $a[i]$ вне цикла	39
Выход за границы диапазона изменения индекса	39
Затирание элементов при сдвиге	39
Матрицы. Тексты программ	41
Общая часть	41
Заполнение матрицы случайными числами и вывод.....	41
Перемена местами двух строк	41
Поиск минимумов в строках.....	42
Поиск максимумов в столбцах	42
Поиск сумм в строках.....	43
Поиск произведений в столбцах.....	43

Наличие нуля в матрице.....	44
Сумма чисел на главной диагонали	44
Сумма чисел на побочной диагонали	45
Заполнение нулями ниже главной диагонали.....	45
Заполнение нулями выше побочной диагонали.....	45
Матрицы. Возможные ошибки.....	46
Записи. Тексты программ	47
Ввод и вывод данных о персонах.....	47
Поиск в массиве персон по фамилии.....	47
Подсчет количества персон с данным возрастом.....	48
Выбор из массива точек точки, наиболее близкой к началу координат	48
Сортировка массива персон по возрасту	49
Строки и символы. Тексты программ	50
Содержание	50
Определение кода по символу и символа по коду	50
Преобразование цифрового символа в соответствующее однозначное число и обратно.....	50
Преобразование числа в строку и обратно.....	50
Формирование строки из n одинаковых символов.....	51
Замена всех маленьких английских букв в строке соответствующими большими.....	51
Сумма записанных в строке чисел.....	51
Количество вхождений подстроки в строку.....	52
Строки и символы. Возможные ошибки.....	52
Ошибка при преобразовании строки в число	52
Выход за границы строки.....	52

Начало. Тексты программ

HelloWorld

```
begin
  writeln('Привет, мир!');
end.
```

Это - самая первая программа на любом языке программирования.

Вычисление площади круга

```
const Pi = 3.1415;

var
  r: real; // радиус круга
  S: real; // площадь круга

begin
  write('Введите радиус круга: ');
  readln(r);
  S := Pi*r*r;
  writeln('Площадь круга равна ',S);
end.
```

Здесь иллюстрируется ввод, затем вычисления и вывод результата. Входные и выходные переменные описаны на разных строчках, описание сопровождается комментарием

Вычисление A^8

Код на PascalABC.NET:

```
var a: real;
begin
  write('Введите a: ');
  readln(a);
  var a2,a4,a8: real; // вспомогательные переменные
  a2 := a * a;
  a4 := a2 * a2;
  a8 := a4 * a4;
  writeln(a, ' в степени 8 = ',a8);
end.
```

Здесь иллюстрируется использование промежуточных переменных и вычислений: без промежуточных вычислений мы бы потратили 7 умножений, а с промежуточными - всего 3.

Вычисление расстояния между двумя точками на прямой

Код на Pascal:

```
var
  a,b: real; // координаты точек
  r: real; // расстояние между точками на прямой

begin
  write('Введите координату точки a: ');
  readln(a);
```

```

write('Введите координату точки b: ');
readln(b);
r := abs(a-b);
writeln('Расстояние между точками = ',r);
end.

```

Здесь иллюстрируется использование стандартной функции `abs`

Вычисление гипотенузы по катетам

```

var
  a,b: real; // катеты
  c: real; // гипотенуза
begin
  write('Введите катеты прямоугольного треугольника: ');
  readln(a,b);
  c := sqrt(a*a+b*b);
  writeln('Гипотенуза = ',c);
end.

```

Здесь иллюстрируется использование стандартной функции `sqrt`

Бросание кубиков

```

var
  r1,r2: integer; // значения на верхних гранях кубиков
begin
  r1 := Random(6)+1;
  r2 := Random(6)+1;
  writeln('Очки, выпавшие на кубиках: ',r1,' ',r2);
  writeln('Сумма очков равна ',r1+r2);
end.

```

Здесь иллюстрируется использование стандартной функции `Random`

Обмен значений двух переменных с использованием третьей

```

var
  x,y: real;
  v: real; // вспомогательная переменная
begin
  write('Введите x,y: ');
  readln(x,y);
  v := x;
  x := y;
  y := v;
  writeln('Новые значения x,y: ',x,' ',y);
end.

```

Это - классическая задача о перемене местами значений двух переменных с использованием третьей переменной. Она часто возникает в других задачах. Интересно, что `x` и `y` являются как входными, так и выходными переменными

Код на PascalABC.NET:

```

var x,y: real;

```

```

begin
  write('Введите x,y: ');
  readln(x,y);
  var v: real; // вспомогательная переменная
  v := x;
  x := y;
  y := v;
  writeln('Новые значения x,y: ',x,' ',y);
end.

```

Данное решение использует внутриблочное описание переменной. В результате вспомогательная переменная описывается первый раз лишь в тот момент, в который она действительно понадобилась

Выделение цифр из двузначного числа

```

var
  x: integer; // двузначное число
  c1,c2: integer; // первая и вторая цифры двузначного числа
begin
  write('Введите двузначное число: ');
  readln(x);
  c1 := x div 10;
  c2 := x mod 10;
  writeln('Первая и вторая цифры двузначного числа: ',c1,' ',c2);
end.

```

Начало. Возможные ошибки

Неописанная переменная

```

begin
  S := 1; // Неизвестное имя S
end.

```

Все используемые переменные должны быть предварительно описаны в разделе описаний. В PascalABC.NET допускаются также описания переменных внутри блока.

Отсутствующая ;

```

var S: integer // Ожидалась ;
begin
  S := 1;
end.
var S: integer;
begin
  S := 1 // Ожидалась ;
  S := S + 1
end.

```

Очень частая ошибка у начинающих. Курсор, как правило, позиционируется в начале следующей строки

Несовместимость типов при присваивании


```
var S: integer;
begin
  S := 1.3; // Нельзя преобразовать тип real к integer
end.
```

Безболезненно можно присваивать только данные одного типа. Если данные - разных типов, то в редких случаях можно преобразовать данные одного типа в данные другого. Например, целое можно преобразовать в вещественное, а символ - в строку. Обратные преобразования не допускаются.

Отсутствие закрывающего апострофа литеральной строки

```
var x, y: integer;
begin
  x := 2; y := 3;
  writeln('Результат сложения равен=x+y); // Не хватает закрывающего
апострофа
end.
```

Закрывать апостроф надо на той же строке, где расположен открывающий апостроф

Ошибки расстановки запятых и апострофов при выводе строк и выражений

```
var x, y: integer;
begin
  x := 2; y := 3;
  writeln(x, '+, 'y, '=' ,x+y); // Неверная расстановка запятых и апострофов
end.
```

Слишком много запятых и апострофов рядом, потому начинающие часто путаются :) Надо уяснить правила:

- запятые разделяют разные элементы вывода
- все, что находится в апострофах, будет выведено на экран без изменений

Ошибка ввода

```
var x: integer;
begin
  read(x); // введите блаблабла и посмотрите, что получится
end.
```

Это - ошибка во время выполнения. Программа пытается преобразовать введенную строку в число, не может это сделать и завершается с ошибкой

Ошибка неинициализированной переменной

```
var x, r: integer;
begin
  // Забыли инициализировать или ввести x
  r := x*x;
  writeln('Квадрат числа ', x, ' = ', r);
end.
```

Перед использованием любую переменную надо ввести или присвоить ей начальное значение. Это действие называется инициализацией переменной.

Деление на 0

```
var x, c: integer;  
begin  
  x := 0;  
  c := 666 div x; // Здесь происходит деление на 0  
end.
```

Если во время выполнения программа выполнит деление на 0, то она завершится с ошибкой.

Корень из отрицательного числа

```
begin  
  writeln(sqrt(-1)); // Корень из отрицательного числа  
end.
```

Условный оператор и оператор выбора. Тексты программ

Минимум из двух значений

```
var
  x,y: integer;
  min: integer;
begin
  write('Введите x и y: ');
  readln(x,y);
  if x<y then
    min := x
  else min := y;
  writeln('Минимум = ',min);
end.
```

Определение четности числа

```
var x: integer;
begin
  write('Введите x: ');
  readln(x);
  if x mod 2 = 0 then
    writeln('Это четное число')
  else writeln('Это нечетное число');
end.
```

Состоит ли двузначное число из одинаковых цифр

Код на PascalABC.NET:

```
var x: integer;
begin
  write('Введите двузначное число: ');
  readln(x);
  var c1 := x div 10;
  var c2 := x mod 10;
  if c1=c2 then
    writeln('Цифры числа совпадают')
  else writeln('Цифры числа не совпадают');
end.
```

Упорядочение двух значений по возрастанию

```
var
  x,y: integer;
  v: integer;
begin
  write('Введите x,y: ');
  readln(x,y);
  if x>y then
    begin
      v := x;
      x := y;
      y := v
    end
end.
```

```

end;
writeln('Результат упорядочения по возрастанию: ',x,' ',y);
end.

```

Проверка числа на двузначность

```

var x: integer;
begin
  write('Введите x: ');
  readln(x);
  if (x>=10) and (x<=100) then
    writeln('Двузначное число')
  else writeln('Не двузначное число')
end.

```

Наименование сезона по номеру месяца

```

var
  Month: integer;
  Season: string;
begin
  write('Введите номер месяца: ');
  readln(Month);
  if (Month=1) or (Month=2) or (Month=12) then
    Season := 'Зима'
  else if (Month=3) or (Month=4) or (Month=5) then
    Season := 'Весна'
  else if (Month=6) or (Month=7) or (Month=8) then
    Season := 'Лето'
  else Season := 'Осень';
  writeln('Это ',Season)
end.

```

Определение четверти, в которой находится точка

```

var
  x,y: integer; // Координаты точки
  Quater: integer; // Номер четверти
begin
  write('Введите координаты точки: ');
  readln(x,y);
  if x>0 then
    if y>0 then
      Quater := 1
    else Quater := 4
  else
    if y>0 then
      Quater := 2
    else Quater := 3;
  writeln('Номер четверти = ',Quater);
end.

```

Нахождение корней квадратного уравнения

```

var
  a,b,c: real;
  x1,x2,D: real;

```

```

begin
  writeln('Введите коэффициенты a,b,c квадратного уравнения a*x*x+b*x+c=0:
');
  readln(a,b,c);
  D := b*b - 4*a*c;
  if D<0 then
    writeln('Корней нет')
  else if D=0 then
    begin
      x1 := -b/2/a;
      writeln('Корни совпадут: x1=x2=',x1);
    end
  else
    begin
      x1 := (-b-sqrt(D))/2/a;
      x2 := (-b+sqrt(D))/2/a;
      writeln('Корни: x1=',x1:0:3,' x2=',x2:0:3);
    end;
end;
end.

```

Словесное наименование сезона по номеру месяца

```

var
  Month: integer;
  Season: string;
begin
  write('Введите номер месяца: ');
  readln(Month);
  case Month of
    1,2,12: Season := 'Зима';
    3..5:   Season := 'Весна';
    6..8:   Season := 'Лето';
    9..11:  Season := 'Осень';
    else    Season := 'Вы ввели неверный номер месяца';
  end;
  writeln('Это ',Season)
end.

```

Определение того, является ли символ цифрой или буквой

```

var
  Symbol: char;
begin
  write('Введите символ: ');
  readln(Symbol);
  case Symbol of
    'a'..'z': writeln('Это маленькая английская буква');
    'A'..'Z': writeln('Это большая английская буква');
    '0'..'9': writeln('Это цифра');
  end;
end;
end.

```

Условный оператор и оператор выбора. Возможные ошибки

Неверная расстановка begin end

```
var
  x,y: integer;
  v: integer;
begin
  write('Введите x,y: ');
  readln(x,y);
  begin // ошибка: операторные скобки должны использоваться по ветке then
оператора if
    if x>y then
      v := x;
      x := y;
      y := v
    end;
  writeln('Результат упорядочения по возрастанию: ',x,' ',y);
end.
```

Операторные скобки должны находиться внутри другого оператора (например, оператора if), а не охватывать его.

Отсутствие begin end

```
var
  x,y: integer;
  v: integer;
begin
  write('Введите x,y: ');
  readln(x,y);
  if x>y then
    v := x; // ошибка: операторные скобки отсутствуют, поэтому по ветке then
выполняется только этот оператор
    x := y; // эти операторы выполняются ПОСЛЕ оператора if в любом случае
    y := v //
  writeln('Результат упорядочения по возрастанию: ',x,' ',y);
end.
```

Операторные скобки begin-end служат для объединения нескольких операторов в один. В данном случае ими следует окружить 3 оператора по ветке then. Если этого не сделать, то по ветке then будет выполняться только один оператор - v := x; последующие операторы будут выполняться после оператора if всегда, что неверно.

Неверная расстановка скобок в сложном логическом выражении

```
var x: integer;
begin
  readln(x);
  if x>1 and x<10 then // ошибка: отсутствуют скобки
    write(x);
end.
```

Циклы. Тексты программ

Вывод последовательностей 1 2 3 4 5 и 5 4 3 2 1

```

var i: integer;
begin
  // С помощью for
  for i := 1 to 5 do
    write(i, ' ');
  writeln;

  for i := 5 downto 1 do
    write(i, ' ');
  writeln;
  writeln;

  // С помощью while
  i := 1;
  while i<=5 do
  begin
    write(i, ' ');
    i := i + 1;
  end;
  writeln;

  i := 5;
  while i>=1 do
  begin
    write(i, ' ');
    i := i - 1;
  end;
  writeln;
  writeln;

  // С помощью repeat
  i := 1;
  repeat
    write(i, ' ');
    i := i + 1;
  until i>5;
  writeln;

  i := 5;
  repeat
    write(i, ' ');
    i := i - 1;
  until i<1;
  writeln;
end.

```

Вывод последовательности 1 3 5 7 9

```

var i, x: integer;

```

```

begin
// С помощью for и промежуточной переменной
x := 1;
for i := 1 to 5 do
begin
write(x, ' ');
x := x + 2;
end;
writeln;

// С помощью for без промежуточной переменной
for i := 1 to 5 do
write(2*i-1, ' ');
writeln;

// С помощью while
x := 1;
while x<10 do
begin
write(x, ' ');
x := x + 2;
end;
writeln;

// С помощью repeat
x := 1;
repeat
write(x, ' ');
x := x + 2;
until x>=10;
end.

```

Сумма и произведение введенных чисел

Код на Pascal

```

var
i: integer;
s,p: real;
x: real;
begin
writeln('Введите 10 чисел: ');
s := 0;
p := 1;
for i := 1 to 10 do
begin
read(x);
s := s + x;
p := p * x;
end;
writeln('Сумма введенных чисел = ',s);
writeln('Произведение введенных чисел = ',p);

```



```
end.
```

Код на PascalABC.NET

```
var
  s,p: real;
begin
  writeln('Введите 10 чисел: ');
  s := 0;
  p := 1;
  for var i := 1 to 10 do
  begin
    var x: integer;
    read(x);
    s += x;
    p *= x;
  end;
  writeln('Сумма введенных чисел = ',s);
  writeln('Произведение введенных чисел = ',p);
end.
```

Вычисление n!

Код на Pascal

```
var
  n, fact: integer;
  i: integer;
begin
  write('Введите n (n<=13): ');
  readln(n);
  fact := 1;
  for i := 2 to n do
    fact := fact * i;
  writeln(n, '! = ', fact);
end.
```

Код на PascalABC.NET

```
var n: integer;
begin
  write('Введите n (n<=13): ');
  readln(n);
  var fact := 1;
  for var i := 2 to n do
    fact *= i;
  writeln(n, '! = ', fact);
end.
```

Вычисление A^n

Код на Pascal

```
var
  n,i: integer;
  a,p: real;
```

```

begin
  write('Введите a,n: ');
  readln(a,n);
  p := 1;
  for i := 1 to n do
    p := p * a;
  writeln(a, ' в степени ',n, ' = ',p);
end.

```

Код на PascalABC.NET

```

var
  n: integer;
  a: real;
begin
  write('Введите a,n: ');
  readln(a,n);
  var p := 1.0;
  for var i := 1 to n do
    p *= a;
  writelnFormat('{0} в степени {1} = {2}',a,n,p);
end.

```

Вывод цифр числа

```

var x: integer;

begin
  write('Введите x: ');
  readln(x);
  write('Цифры числа x в обратном порядке: ');
  while x <> 0 do
    begin
      write(x mod 10, ' ');
      x := x div 10;
    end;
end.

```

Вывод букв английского алфавита

```

var c: char;

begin
  for c := 'a' to 'z' do
    write(c, ' ');
  writeln;
  c := 'A';
  while c <= 'Z' do
    begin
      write(c, ' ');
      c := succ(c);
    end;
end.

```

Числа Фибоначчи

Код на Pascal

```

const n = 25;

var
  a,b,c: integer;
  i: integer;

begin
  a := 1;
  b := 1;
  write(a, ' ', b, ' ');
  for i := 3 to n do
    begin
      c := a + b;
      write(c, ' ');
      a := b;
      b := c;
    end;
end.

```

Код на PascalABC.NET

```

const n = 25;

begin
  var a := 1;
  var b := 1;
  write(a, ' ', b, ' ');
  for var i := 3 to n do
    begin
      var c := a + b;
      write(c, ' ');
      a := b;
      b := c;
    end;
end.

```

Минимум из введенных**Код на Pascal**

```

const n = 10;

var
  min: integer;
  x: integer;
  i: integer;

begin
  writeln('Введите ', n, ' значений: ');
  read(x);
  min := x;

```

```

for i := 2 to n do
begin
  read(x);
  if x<min then
    min := x;
  end;
  writeln('Минимальное значение = ',min);
end.

```

Код на PascalABC.NET

```

const n = 10;

var min: integer;

begin
  writelnFormat('Введите {0} значений: ',n);
  min := integer.MaxValue;
  for var i := 1 to n do
  begin
    var x: integer;
    read(x);
    if x<min then
      min := x;
    end;
    writeln('Минимальное значение = ',min);
  end.

```

Алгоритм Евклида поиска НОД

```

var a,b,c: integer;

begin
  writeln('Введите a,b: ');
  read(a,b);
  while b<>0 do
  begin
    c := a mod b;
    a := b;
    b := c;
  end;
  writeln('Наибольший Общий Делитель = ',a);
end.

```

Вывод таблицы умножения

Код на PascalABC.NET

```

const n = 9;

begin
  for var i:=1 to n do
  begin
    for var j:=1 to n do
      write(i*j:4);
    end;
  end;
end.

```

```
writeln;
end;
end.
```

Определение простоты числа

```
var
  N: integer;
  IsSimple: boolean;

begin
  writeln('Введите число: ');
  readln(N);

  IsSimple := True;
  for var i:=2 to round(sqrt(N)) do // если число составное, то один из его
    сомножителей <= sqrt(N)
    if N mod i = 0 then
    begin
      IsSimple := False;
      break;
    end;

  if IsSimple then
    writeln('Число ',N, ' простое')
  else writeln('Число ',N, ' составное');
end.
```

Циклы. Возможные ошибки

Неверный тип переменной цикла

```
var x: real;
begin
  for x := 1 to 10 do // переменная - параметр цикла не может иметь
    вещественный тип
    write(x);
end.
```

Изменение переменной цикла в теле цикла

Пример 1.

```
var i: integer;
begin
  for i := 1 to 10 do
  begin
    write(i);
    i := i + 2; // Нельзя изменять значение переменной цикла внутри цикла
  end;
end.
```

Пример 2.

```
var i: integer;
```

```

begin
  for i := 1 to 10 do
    begin
      read(i); // Нельзя изменять значение переменной цикла внутри цикла
      write(i);
    end;
  end.

```

Использование одинаковой переменной цикла в заголовках вложенных циклов

```

var i, j: integer;
begin
  for i := 1 to 10 do
    for i := 1 to 10 do // ошибка: во вложенном цикле нельзя использовать
      параметр цикла с тем же именем
    begin
      write(i);
    end;
  end.

```

Неинициализированная переменная

```

var
  i: integer;
  p: real;
begin
  // Забыли инициализировать p := 1
  for i := 1 to 10 do
    p := p * 2;
    write(p);
  end.

```

Защипливание

```

var x: integer;
begin
  x := 1;
  while x>0 do // ошибка: условие все время остается истинным
    begin
      writeln(x);
      x := x + 1;
    end;
  end.

```

Переполнение

```

var
  n: integer;
  x: real;
begin
  n := integer.MaxValue; // максимальное целое
  writeln(n);
  n := n + 1; // следующее за максимальным целым
  writeln(n);
  writeln;

```

```
x := real.MaxValue; // максимальное вещественное
writeln(x);
x := x * 2; // больше, чем максимальное вещественное
writeln(x); // в ряде компиляторов возникает ошибка переполнения.
             // Здесь x получает значение "бесконечность"
```

end.

Сравнение вещественных чисел на равенство

begin

```
if 0.3*3 = 0.9 then // Значения не равны ввиду ошибки округления при работе
с вещественными числами
```

```
  writeln('Равны')
```

```
else writeln('Не равны');
```

end.

Процедуры и функции. Тексты программ

Вычисление площади и периметра прямоугольника

```

procedure RectSP(a,b: real; var S,P:real);
begin
  S := a * b;
  P := a + b;
end;

var
  x,y: real; // Стороны прямоугольника
  S: real; // Площадь прямоугольника
  P: real; // Периметр прямоугольника

begin
  write('Введите стороны прямоугольника: ');
  readln(x,y);
  RectSP(x,y,S,P);
  writeln('Площадь равна ',S);
  writeln('Периметр равен ',P);
end.

```

Табулирование функции

```

function f(x: real): real;
begin
  Result := x*x - 2*x + 3;
end;

var
  a,b: real; // [a,b] - отрезок, на котором табулируется функция f(x)
  N: integer; // Количество частей, на которые разбивается [a,b]

begin
  write('Введите a,b: ');
  readln(a,b);
  write('Введите N: ');
  readln(N);

  var h := (b-a)/N;
  var x := a;
  writeln('x      f(x)');
  for var i:=0 to N do
  begin
    writelnFormat('{0,3:f1} {1,6:f2} ',x,f(x));
    x += h;
  end;
end.

```

Функция вычисления a^n

```

function Power(a: real; n: integer): real;
begin
  Result := 1;

```



```

    for var i:=1 to n do
        Result *= a;
    end;

var
    a: real;
    n: integer;

begin
    write('Введите a: ');
    readln(a);
    write('Введите n: ');
    readln(n);
    writelnFormat('{0} в степени {1} = {2}', a, n, Power(a, n));
end.

```

Функция вычисления n!

```

function Factorial(n: integer): integer;
begin
    Result := 1;
    for var i:=2 to n do
        Result *= i;
    end;

var n: integer;

begin
    write('Введите n: ');
    readln(n);
    writelnFormat('{0}! = {1}', n, Factorial(n));
end.

```

Функция, преобразующая цифру к ее словесному наименованию

```

function DigitName(n: integer): string;
begin
    case n of
        0: Result := 'ноль';
        1: Result := 'один';
        2: Result := 'два';
        3: Result := 'три';
        4: Result := 'четыре';
        5: Result := 'пять';
        6: Result := 'шесть';
        7: Result := 'семь';
        8: Result := 'восемь';
        9: Result := 'девять';
    end;
end;

begin
    writeln(DigitName(3));
    writeln(DigitName(5));
end.

```

```
writeln(DigitName(7));
end.
```

Функция проверки числа на четность

```
function Even(x: integer): boolean;
begin
  Result := x mod 2 = 0;
end;
```

```
begin
  writeln(Even(3));
  writeln(Even(4));
  writeln(Even(7));
end.
```

Функция определения простоты числа

```
function IsSimple(N: integer): boolean;
begin
  Result := True;
  for var i:=2 to round(sqrt(N)) do
    if N mod i = 0 then
      begin
        Result := False;
        exit;
      end;
  end;
end;
```

```
begin
  writeln(IsSimple(13));
  writeln(IsSimple(4));
  writeln(IsSimple(7));
end.
```

Функция вычисления n-того числа Фибоначчи

```
function Fib(N: integer): integer;
begin
  if (N=1) or (n=2) then
    Result := 1;
  else
    begin
      var a := 1;
      var b := 1;
      for var i:=3 to N do
        begin
          Result := a + b;
          a := b;
          b := Result;
        end;
    end;
end;
```

```
begin
```

```
writeln(Fib(5));
writeln(Fib(6));
writeln(Fib(7));
```

end.

Функция определения НОД числа

```
function NOD(a,b: integer): integer;
```

```
begin
```

```
  while b<>0 do
```

```
  begin
```

```
    var c := a mod b;
```

```
    a := b;
```

```
    b := c;
```

```
  end;
```

```
  Result := a;
```

```
end;
```

```
var x,y: integer;
```

```
begin
```

```
  x := 15; y := 12;
```

```
  writelnFormat('НОД чисел {0} и {1} равен {2}',x,y,NOD(x,y));
```

```
  x := 144; y := 64;
```

```
  writelnFormat('НОД чисел {0} и {1} равен {2}',x,y,NOD(x,y));
```

```
  x := 99; y := 54;
```

```
  writelnFormat('НОД чисел {0} и {1} равен {2}',x,y,NOD(x,y));
```

end.

Процедура вывода таблицы умножения

```
procedure Pifagor(n: integer);
```

```
begin
```

```
  for var i:=1 to n do
```

```
  begin
```

```
    for var j:=1 to n do
```

```
      write(i*j:4);
```

```
    writeln;
```

```
  end;
```

```
end;
```

```
begin
```

```
  Pifagor(15);
```

end.

Процедуры и функции. Возможные ошибки

Невозвращение значения функции по одной из ветвей алгоритма

```
function Bad(x: integer): integer;
```

```
begin
```

```
  if x>0 then
```

```
    Result := 777
```

```
  // При x<=0 значение функции непредсказуемо
```

```
end;
```

```
begin
  writeln(Bad(-2));
end.
```

Описание переменной цикла, используемого в процедуре, как глобальной

```
var i: integer;

procedure Bad;
begin
  for i := 1 to 10 do // Переменная-параметр цикла for должна описываться
    подпрограмме, где используется цикл
    write(i);
end;

begin
  for i := 1 to 10 do
    Bad;
end.
```

Передача выходных параметров по значению

```
procedure BadInc(i: integer);
begin
  i := i + 1;
end;

var i: integer;

begin
  i := 5;
  writeln(i);
  BadInc(i); // меняется копия переменной i, а не сама переменная i.
  Необходимо использовать var-параметр
  writeln(i);
end.
```

Массивы. Тексты программ

Ввод и вывод

```

const Sz = 100; // Размер массива

var
  a: array [1..Sz] of integer;
  N: integer; // Количество элементов в массиве
  i: integer;

begin
  write('Введите количество элементов в массиве: ');
  readln(N);
  write('Введите элементы массива: ');
  for i:=1 to N do
    read(a[i]);
  write('Вывод элементов массива: ');
  for i:=1 to N do
    write(a[i], ' ');
end.

```

Заполнение случайными числами

```

const Sz = 100; // Размер массива

var
  a: array [1..Sz] of integer;
  N: integer; // Количество элементов в массиве
  i: integer;

begin
  N := 20;
  for i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива: ');
  for i:=1 to N do
    write(a[i], ' ');
end.

```

Заполнение арифметической прогрессией

```

const
  Sz = 100;
  a0 = 5; // Первый элемент арифметической прогрессии
  d = 3; // Разность арифметической прогрессии

var
  a: array [1..Sz] of integer;
  N: integer; // Количество элементов в массиве

begin
  N := 20;
  a[1] := a0;
  for var i:=2 to N do

```

```

    a[i] := a[i-1] + d;
writeln('Арифметическая прогрессия: ');
for var i:=1 to N do
    write(a[i], ' ');

```

end.

Заполнение степенями двойки

```

const
    Sz = 100;
var
    a: array [1..Sz] of integer;
    N: integer;
begin
    N := 20;
    a[1] := 2;
    for var i:=2 to N do
        a[i] := a[i-1] * 2;
writeln('Степени двойки: ');
    for var i:=1 to N do
        writeln(i:3,a[i]:9);

```

end.

Заполнение числами Фибоначчи

```

const
    Sz = 100;
var
    a: array [1..Sz] of integer;
    N: integer;
begin
    N := 20;
    a[1] := 1;
    a[2] := 1;
    for var i:=3 to N do
        a[i] := a[i-2] + a[i-1];
writeln('Числа Фибоначчи: ');
    for var i:=1 to N do
        write(a[i], ' ');

```

end.

Инвертирование массива

```

const
    Sz = 100;
var
    a: array [1..Sz] of integer;
    N: integer;
begin
    N := 20;
    for var i:=1 to N do
        a[i] := Random(100);
writeln('Элементы массива: ');
    for var i:=1 to N do
        write(a[i], ' ');

```

```
writeln;

for var i:=1 to N div 2 do
  Swap(a[i],a[N-i+1]);

writeln('После инвертирования: ');
for var i:=1 to N do
  write(a[i], ' ');
end.
```

Максимальный элемент в массиве и его индекс

```
const
  Sz = 100;
var
  a: array [1..Sz] of real;
  N: integer;
  min: real;
  minind: integer;
begin
  N := 20;
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  min := a[1];
  minind := 1;
  for var i:=2 to N do
    if a[i]<min then
      begin
        min := a[i];
        minind := i;
      end;

  writeln('Минимальный элемент: ',min);
  writeln('Индекс минимального элемента: ',minind);
end.
```

Минимальный четный элемент и его индекс

```
const
  Sz = 100;
var
  a: array [1..Sz] of integer;
  N: integer;
  min: integer;
  minind: integer;
begin
  N := 20;
  for var i:=1 to N do
    a[i] := Random(100);
```

```

writeln('Элементы массива: ');
for var i:=1 to N do
  write(a[i], ' ');
writeln;

min := integer.MaxValue;
for var i:=1 to N do
  if (a[i]<min) and (a[i] mod 2 = 0) then
    begin
      min := a[i];
      minind := i;
    end;

if min = integer.MaxValue then
  writeln('Четных элементов нет')
else
  begin
    writeln('Минимальный четный элемент: ',min);
    writeln('Индекс минимального четного элемента: ',minind);
  end;
end.

```

Запись четных элементов массива в новый массив

```

const
  Sz = 100;
var
  a,b: array [1..Sz] of integer;
  aN: integer; // Количество элементов в массиве a
  bN: integer; // Количество элементов в массиве b
begin
  aN := 20;
  for var i:=1 to aN do
    a[i] := Random(100);
  writeln('Элементы массива: ');
  for var i:=1 to aN do
    write(a[i], ' ');
  writeln;

  bN := 0;
  for var i:=1 to aN do
    if a[i] mod 2 = 0 then
      begin
        bN += 1;
        b[bN] := a[i];
      end;

  writeln('Четные элементы массива: ');
  for var i:=1 to bN do
    write(b[i], ' ');
  writeln;
end.

```

Слияние отсортированных массивов в отсортированный

Способ 1.

```

const
  aN = 10;      // Количество элементов в массиве a
  bN = 6;      // Количество элементов в массиве b
  cN = aN + bN; // Количество элементов в массиве c

var
  a: array [1..aN] of integer := (1,5,12,15,47,89,98,112,171,180);
  b: array [1..bN] of integer := (13,44,58,71,73,111);
  c: array [1..cN] of integer;
  ai,bi,ci: integer;
begin
  writeln('Элементы массива a: ');
  for var i:=1 to aN do
    write(a[i], ' ');
  writeln;
  writeln('Элементы массива b: ');
  for var i:=1 to bN do
    write(b[i], ' ');
  writeln;

  ci := 1;
  ai := 1;
  bi := 1;

  while (ai<=aN) and (bi<=bN) do
  begin
    if a[ai]<b[bi] then
    begin
      c[ci] := a[ai];
      ai += 1;
    end
    else
    begin
      c[ci] := b[bi];
      bi += 1;
    end;
    ci += 1;
  end;
  while ai<=aN do
  begin
    c[ci] := a[ai];
    ai += 1;
    ci += 1;
  end;
  while bi<=bN do
  begin
    c[ci] := b[bi];
    bi += 1;
    ci += 1;
  end;

```

```

end;

writeln('Результат слияния: ');
for var i:=1 to cN do
  write(c[i], ' ');
end.

```

Способ 2. С барьерным элементом

```

const
  aN = 10;           // Количество элементов в массиве a
  bN = 6;           // Количество элементов в массиве b
  cN = aN + bN;     // Количество элементов в массиве c

var
  a: array [1..aN+1] of integer := (1,5,12,15,47,89,98,112,171,180,0); //
  // последний элемент - барьерный
  b: array [1..bN+1] of integer := (13,44,58,71,73,111,0);
  c: array [1..cN] of integer;
  ai,bi,ci: integer;

begin
  writeln('Элементы массива a: ');
  for var i:=1 to aN do
    write(a[i], ' ');
  writeln;
  writeln('Элементы массива b: ');
  for var i:=1 to bN do
    write(b[i], ' ');
  writeln;

  a[aN+1] := integer.MaxValue; // барьерный элемент - самый большой
  b[bN+1] := integer.MaxValue;

  ci := 1;
  ai := 1;
  bi := 1;

  for ci:=1 to cN do
    if a[ai]<b[bi] then
      begin
        c[ci] := a[ai];
        ai += 1;
      end
    else
      begin
        c[ci] := b[bi];
        bi += 1;
      end
    end;

  writeln('Результат слияния: ');
  for var i:=1 to cN do
    write(c[i], ' ');

```

```
end.
```

Сдвиг элементов влево

```
const N = 10;

var a: array [1..N] of integer;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  for var i:=1 to N-1 do
    a[i] := a[i+1];
  a[N] := 0;

  writeln('После сдвига влево: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;
end.
```

Сдвиг элементов вправо

```
const N = 10;

var a: array [1..N] of integer;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  for var i:=N downto 2 do
    a[i] := a[i-1];
  a[1] := 0;

  writeln('После сдвига влево: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;
end.
```

Удаление элемента

```
const N = 10;
```

```
var
```

```

a: array [1..N] of integer;
K: integer;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  K := Random(1,N);

  for var i:=K to N-1 do
    a[i] := a[i+1];

  writeln('После удаления элемента с индексом ',K, ':');
  for var i:=1 to N-1 do
    write(a[i], ' ');
  writeln;
end.

```

Вставка элемента

```

const
  N = 10;
  Elem = 666;

var
  a: array [1..N+1] of integer;
  K: integer;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  K := Random(1,N);

  for var i:=N downto K do
    a[i+1] := a[i];
  a[K] := 666;

  writeln('После вставки элемента ',Elem, ' в позицию ',K, ':');
  for var i:=1 to N+1 do
    write(a[i], ' ');
  writeln;
end.

```

Подсчет количества элементов, удовлетворяющих условию

```

const N = 20;

var
  a: array [1..N] of integer;
  K, Count: integer;

begin
  for var i:=1 to N do
    a[i] := Random(10);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  K := Random(10);

  Count := 0;
  for var i:=1 to N do
    if a[i] = K then
      Count += 1;

  writeln('Количество элементов, равных ', K, ': ', Count);
end.

```

Есть ли элемент, удовлетворяющий условию

```

const N = 10;

var
  a: array [1..N] of integer;
  K: integer;
  IsFound: boolean;

begin
  for var i:=1 to N do
    a[i] := Random(15);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  K := Random(15);

  IsFound := False;
  for var i:=1 to N do
    if a[i] = K then
      begin
        IsFound := True;
        break
      end;

  if IsFound then

```

```

    writeln('Элемент ',K,' найден')
  else writeln('Элемент ',K,' не найден')
end.

```

Сортировка пузырьком

```

const N = 10;

var
  a: array [1..N] of integer;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  for var i:=n downto 2 do
    for var j:=1 to i-1 do
      if a[j+1]<a[j] then
        Swap(a[j+1],a[j]);

  writeln('После сортировки пузырьком: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;
end.

```

Сортировка выбором

```

const N = 10;

var
  a: array [1..N] of integer;
  K: integer;
  IsFound: boolean;

begin
  for var i:=1 to N do
    a[i] := Random(100);
  writeln('Элементы массива a: ');
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;

  for var i:=1 to N-1 do
    begin
      var min := a[i];
      var ind := i;
      for var j:=i+1 to N do
        if a[j]<min then
          begin

```

```

    min := a[j];
    ind := j;
  end;
  a[ind] := a[i];
  a[i] := min;
end;

writeln('После сортировки выбором: ');
for var i:=1 to N do
  write(a[i], ' ');
writeln;
end.

```

Массивы. Возможные ошибки

Попытка вывести массив не поэлементно, а целиком

```

const Sz = 100;

var
  a: array [1..Sz] of integer;
  i: integer;

begin
  read(a); // неверно: массив необходимо вводить поэлементно
end.

```

Использование записи a[i] вне цикла

```

const Sz = 10;

var
  a: array [1..Sz] of integer;
  i: integer;

begin
  var N := Sz;
  for i:=1 to N do
    a[i] := Random(100);
  a[i] := 666; // Ошибка: переменной i нельзя пользоваться вне цикла for
end.

```

Выход за границы диапазона изменения индекса

```

const Sz = 10;

var a: array [1..Sz] of integer;

begin
  var N := Sz;
  for var i:=1 to N+1 do
    a[i] := Random(100); // При i=N+1 происходит ошибка: индекс вне границ массива
  end.

```

Затирание элементов при сдвиге

```
const Sz = 100;

var a: array [1..Sz] of integer;

begin
  var N := 10;
  for var i:=1 to N do
    a[i] := Random(100);
  for var i:=1 to N do
    write(a[i], ' ');
  writeln;
  for var i:=1 to N-1 do // ошибка: алгоритм сдвига вправо надо начинать с
конца
    a[i+1] := a[i];
  for var i:=1 to N do
    write(a[i], ' ');
end.
```


Матрицы. Тексты программ

Общая часть

Во всех задачах будем считать, что матрица описана следующим образом:

```
const
  SzM = 10;
  SzN = 10;

type Matrix = array [1..SzM,1..SzN] of integer;
```

Кроме того, во всех задачах будем использовать следующие процедуры для заполнения и вывода:

```
procedure FillMatrixByRandom(var a: Matrix; m,n: integer); // Заполнение
случайными
begin
  for var i:=1 to M do
    for var j:=1 to N do
      a[i,j] := Random(10);
end;

procedure PrintMatrix(const a: Matrix; m,n: integer); // Вывод матрицы
begin
  for var i:=1 to M do
    begin
      for var j:=1 to N do
        write(a[i,j]:4);
        writeln;
      end;
    end;
end;
```

Данные участки кода следует писать в начале каждой программы, приводимой в этом пункте.

Заполнение матрицы случайными числами и вывод

```
var a: Matrix;

begin
  var m := 4;
  var n := 5;
  FillMatrixByRandom(a,m,n);

  writeln('Элементы матрицы: ');
  PrintMatrix(a,m,n);
end.
```

Перемена местами двух строк

```
var a: Matrix;

begin
  var m := 5;
  var n := 9;
  FillMatrixByRandom(a,m,n);
```

```
writeln('Исходная матрица: ');
PrintMatrix(a,m,n);

var k1 := 2; // поменять местами строки с номерами k1 и k2
var k2 := 4;
for var j:=1 to n do
  Swap(a[k1,j],a[k2,j]);

writeln('Преобразованная матрица: ');
PrintMatrix(a,m,n);
end.
```

Поиск минимумов в строках

```
var
  a: Matrix;
  mins: array [1..SzN] of integer;

begin
  var m := 5;
  var n := 9;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  for var i:=1 to m do
    begin
      var min := a[i,1];
      for var j:=2 to n do
        if a[i,j]<min then
          min := a[i,j];
      mins[i] := min;
    end;

  writeln('Минимумы в строках: ');
  for var i:=1 to m do
    write(mins[i]:4);
  end.
```

Поиск максимумов в столбцах

```
var
  a: Matrix;
  maxs: array [1..SzN] of integer;

begin
  var m := 5;
  var n := 9;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);
```

```

for var j:=1 to n do
begin
  var max := a[1,j];
  for var i:=2 to m do
    if a[i,j]>max then
      max := a[i,j];
  maxs[j] := max;
end;

writeln('Максимумы в столбцах: ');
for var j:=1 to n do
  write(maxs[j]:4);
end.

```

Поиск сумм в строках

```

var
  a: Matrix;
  sums: array [1..SzM] of integer;

begin
  var m := 4;
  var n := 5;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  for var i:=1 to m do
  begin
    var sum := 0;
    for var j:=1 to n do
      sum += a[i,j];
    sums[i] := sum;
  end;

  writeln('Суммы в строках: ');
  for var i:=1 to m do
    write(sums[i]:4);
  end.

```

Поиск произведений в столбцах

```

var
  a: Matrix;
  products: array [1..SzN] of integer;

begin
  var m := 3;
  var n := 4;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

```

```

for var j:=1 to n do
begin
  var p := 1;
  for var i:=1 to m do
    p *= a[i,j];
  products[j] := p;
end;

writeln('Произведения в столбцах: ');
for var j:=1 to n do
  writeln(products[j]:4);
end.

```

Наличие нуля в матрице

```

var
  a: Matrix;
  HasZero: boolean;

label 1;

begin
  var m := 3;
  var n := 4;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  HasZero := False;
  for var i:=1 to m do
  for var j:=1 to n do
    if a[i,j]=0 then
      begin
        HasZero := True;
        goto 1;
      end;
  1:
  if HasZero then
    writeln('В матрице есть нули')
  else writeln('В матрице нулей нет')
end.

```

Сумма чисел на главной диагонали

```

var
  a: Matrix;
  sum: integer;

begin
  var m := 5;
  var n := m;
  FillMatrixByRandom(a,m,n);

```

```
writeln('Исходная матрица: ');
PrintMatrix(a,m,n);

sum := 0;
for var i:=1 to m do
  sum += a[i,i];
writeln('Сумма элементов главной диагонали: ',sum);
end.
```

Сумма чисел на побочной диагонали

```
var
  a: Matrix;
  sum: integer;

begin
  var m := 5;
  var n := m;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  sum := 0;
  for var i:=1 to m do
    sum += a[i,m-i+1];
  writeln('Сумма элементов побочной диагонали: ',sum);
end.
```

Заполнение нулями ниже главной диагонали

```
var
  a: Matrix;
  sum: integer;

begin
  var m := 5;
  var n := m;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  for var i:=2 to m do
    for var j:=m-i+2 to m do
      a[i,j] := 0;

  writeln('Преобразованная матрица: ');
  PrintMatrix(a,m,n);
end.
```

Заполнение нулями выше побочной диагонали

```
var
```

```

a: Matrix;
sum: integer;

begin
  var m := 5;
  var n := m;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  for var i:=1 to m-1 do
    for var j:=i+1 to m do
      a[i,j] := 0;

  writeln('Преобразованная матрица: ');
  PrintMatrix(a,m,n);
end.

```

Матрицы. Возможные ошибки

```

var
  a: Matrix;
  sums: array [1..SzM] of integer;

begin
  var m := 4;
  var n := 5;
  FillMatrixByRandom(a,m,n);

  writeln('Исходная матрица: ');
  PrintMatrix(a,m,n);

  var sum := 0;
  for var i:=1 to m do
    sum += a[1,i];      // здесь перепутаны строка и столбец: переменная i
                        // является
                        // номером столбца и меняется от 1 до m, что
                        // неправильно
end.

```

Записи. Тексты программ

Ввод и вывод данных о персонах

```

type
  Person = record
    name: string;
    age: integer;
  end;

var p1,p2: Person;

begin
  readln(p1.name,p1.age);
  writeln(p1.name, ' ',p1.age);
  p2.name := 'Иванов';
  p2.age := 20;
  p1 := p2;
  writeln(p1.name, ' ',p1.age);
end.

```

Поиск в массиве персон по фамилии

```

type
  Person = record
    name: string;
    age: integer;
  end;

var a: array[1..5] of Person;

begin
  a[1].name := 'Попова'; a[1].age := 20;
  a[2].name := 'Петрова'; a[2].age := 18;
  a[3].name := 'Крылов'; a[3].age := 23;
  a[4].name := 'Иванов'; a[4].age := 25;
  a[5].name := 'Сидоров'; a[5].age := 17;

  var fio: string;
  write('Введите фамилию: ');
  readln(fio);

  var ind := 0;
  for var i:=1 to 5 do
    if a[i].name=fio then
      begin
        ind := i;
        break
      end;

  if ind>0 then
    writeln('Найдено: ',a[ind].name, ' ',a[ind].age)
  else writeln('Человека с такой фамилией нет в списке');
end.

```

end.

Подсчет количества персон с данным возрастом

type

```
Person = record
  name: string;
  age: integer;
end;
```

var

```
a: array [1..5] of Person;
count: integer;
```

begin

```
a[1].name := 'Попова'; a[1].age := 20;
a[2].name := 'Петрова'; a[2].age := 20;
a[3].name := 'Крылов'; a[3].age := 23;
a[4].name := 'Иванов'; a[4].age := 20;
a[5].name := 'Сидоров'; a[5].age := 17;
```

```
var age := 20;
```

```
count := 0;
```

```
for var i:=1 to 5 do
  if a[i].age=age then
    count += 1;
```

```
writeln('Количество человек с возрастом ',age,' равно ',count);
```

end.

Выбор из массива точек точки, наиболее близкой к началу координат

type

```
Point = record
  x,y: integer;
end;
```

var

```
a: array [1..5] of Point;
min: real;
indmin: integer;
```

begin

```
a[1].x := 5; a[1].y := 3;
a[2].x := 4; a[2].y := 4;
a[3].x := 1; a[3].y := 8;
a[4].x := -3; a[4].y := 2;
a[5].x := -2; a[5].y := 4;
```

```
min := MaxDouble;
```

```
for var i:=1 to 5 do
```

```
begin
```

```
var p := sqr(a[i].x)+sqr(a[i].y);
```

```
if p<min then
```



```

begin
    min := p;
    indmin := i;
end;
end;

writeln('Точка, ближайшая к началу координат: ', a[indmin].x, '
', a[indmin].y);
writeln('Ее расстояние до начала координат: ', sqrt(min):5:3);
end.

```

Сортировка массива персон по возрасту

```

type
    Person = record
        name: string;
        age: integer;
    end;

const n=5;

var
    a: array [1..n] of Person;

begin
    a[1].name := 'Попова'; a[1].age := 22;
    a[2].name := 'Петрова'; a[2].age := 15;
    a[3].name := 'Крылов'; a[3].age := 23;
    a[4].name := 'Иванов'; a[4].age := 21;
    a[5].name := 'Сидоров'; a[5].age := 17;

    for var i:=n downto 2 do
        for var j:=1 to i-1 do
            if a[j+1].age < a[j].age then
                Swap(a[j+1], a[j]);
        end;
    end;

    writeln('Массив персон, отсортированный по возрастанию возраста:');
    for var i:=1 to n do
        writeln(a[i].name:8, a[i].age:4);
    end;
end.

```

Строки и символы. Тексты программ

Содержание

[\[убрать\]](#)

- 1 Определение кода по символу и символа по коду
- 2 Преобразование цифрового символа в соответствующее однозначное число и обратно
- 3 Преобразование числа в строку и обратно
- 4 Формирование строки из n одинаковых символов
- 5 Замена всех маленьких английских букв в строке соответствующими большими
- 6 Сумма записанных в строке чисел
- 7 Количество вхождений подстроки в строку
- 8 Ссылки

Определение кода по символу и символа по коду

```
var
  c: char;
  code: integer;
begin
  c := 'z';
  code := Ord(c);
  writeln('Код символа ', c, ' равен ', code);
  code := 35;
  c := Chr(code);
  writeln('Символ с кодом ', code, ' - это ', c);
end.
```

Преобразование цифрового символа в соответствующее однозначное число и обратно

```
var
  c: char;
  digit: integer;
begin
  c := '5';
  digit := Ord(c) - Ord('0');
  writeln('Цифра, соответствующая символу ', c, ' - это ', digit);
  digit := 7;
  c := Chr(digit + Ord('0'));
  writeln('Символ цифры ', digit, ' - это ', c);
end.
```

Преобразование числа в строку и обратно

```
var
  s: string;
  num: integer;
begin
  s := '356';
  num := StrToInt(s);
  writeln('Число, соответствующее строке ', s, ' - это ', num);
```

```

num := 563;
s := IntToStr(num);
writeln('Строка, соответствующая числу ', num, ' - это ', s, '');
end.

```

Формирование строки из n одинаковых символов

```

var
  c: char;
  s: string;
  n: integer;
begin
  n := 10;
  c := 'z';
  s := '';
  for var i:=1 to n do
    s += c;
  writeln(s);
end.

```

Замена всех маленьких английских букв в строке соответствующими большими

```

var s: string := 'Hello, World!';
begin
  var dist := Ord('a')-Ord('A');
  for var i:=1 to s.Length do
    if s[i] in ['a'..'z'] then
      s[i] := Chr(Ord(s[i])-dist);
  writeln(s);
end.

```

Сумма записанных в строке чисел

```

var
  s: string := '1 2 4 8 16 32 64 128 256 512';
  sum: integer;
  num,p: integer;
  numstr: string;
begin
  sum := 0;
  while s.Length>0 do
    begin
      while s[1]=' ' do // Удаление лидирующих пробелов
        Delete(s,1,1);
      p := Pos(' ',s); // Поиск следующего пробела
      if p=0 then // Если он не найден, то до конца строки находится
        последнее число
        p := s.Length;
      numstr := Copy(s,1,p);
      Delete(s,1,p);
      num := StrToInt(numstr);
      sum += num;
    end;
  writeln('Сумма чисел строки равна ', sum);
end.

```

```
end.
```

Количество вхождений подстроки в строку

```
var
  s: string := 'this bra is abracadabra';
  subs := 'bra';
  count: integer;
begin
  count := 0;
  var scopy := s;
  while s.Length > 0 do
    begin
      var p := Pos(subs, s); // Поиск подстроки в строке
      if p = 0 then
        break;
      count += 1;
      Delete(s, 1, p + subs.Length);
    end;
    writelnFormat('Количество вхождений '{0}' в '{1}' равно
  {2}', subs, scopy, count);
end.
```

Строки и символы. Возможные ошибки

Ошибка при преобразовании строки в число

```
var
  s: string := 'z123';
  i: integer;
begin
  // Первый способ - обработка исключений
  try
    i := StrToInt(s);
  except
    writeln('Ошибка преобразования');
  end;
  var errcode: integer;
  // Второй способ - процедура Val
  Val(s, i, errcode);
  if errcode > 0 then
    writeln('Ошибка преобразования')
  else writeln('Преобразованное число ', i);
end.
```

Выход за границы строки

```
var
  s: string := 'ABC';
begin
  s[4] := 'z'; // Ошибка времени выполнения: индекс находится вне границ
  строки
end.
```