

Национальный институт образования

# ИНФОРМАТИКА

ФАКУЛЬТАТИВНЫЕ ЗАНЯТИЯ



А.И. Лапо

## ИНФОРМАТИКА В ЗАДАЧАХ

8–9 КЛАССЫ

Пособие для учащихся

ISBN 978-985-471-688-6



9 789854 716886

УДК 004(075.3=161.3=161.1)  
ББК 32.81я721  
Л24

## ПРЕДИСЛОВИЕ

Уважаемые читатели, перед вами пособие для учащихся к факультативу «Информатика в задачах». В этой книге собраны различные задачи для успешного освоения материала курса.

Чтобы найти правильное решение задачи, надо понять её смысл, то есть проанализировать исходную информацию, а затем преобразовать её и получить правильный результат. В пособии приведены разборы решений многих задач. Тестирование готовой программы позволит определить, правильно ли решена задача. Ответы на вопросы исследовательского характера помогут понять, каким образом можно решать похожие задачи.

Для закрепления умений решения задач в пособии приведены индивидуальные задания в четырёх вариантах.

Работая с материалами пособия, вы научитесь выделять этапы решения задачи с помощью компьютера, использовать процедуры и функции для решения задач, ознакомитесь с алгоритмами целочисленной арифметики, сможете работать с массивами, решать геометрические задачи.

В приложении приведены некоторые процедуры и функции языка Pascal, математические формулы, необходимые для решения задач.

Наиболее сложные задания и задачи помечены звездочкой (\*).

Успехов вам на пути к освоению программирования и алгоритмизации!

**Лапо, А. И.**  
Л24 Информатика в задачах : 8—9-е кл. : пособие для учащихся учреждений общ. сред. образования с белорус. и рус. яз. обучения / А. И. Лапо. — 2-е изд. — Минск : Адукацыя і выхаванне, 2014. — 144 с. : ил. — (Информатика. Факультативные занятия)

ISBN 978-985-471-688-6.

Содержатся теоретические вопросы и практические задания в поддержку факультативного курса «Информатика в задачах» для 8—9-го классов.

Теоретические вопросы позволяют проверить уровень усвоения материала. Некоторые практические задания в пособии даются с разбором типовой задачи, другие предназначены для самостоятельного решения. Разбор типовой задачи содержит все этапы решения: анализ входных и выходных данных, подробное описание алгоритма решения, программу, тесты для проверки, дополнительные задания.

Для каждого типа задач предлагаются 4 варианта. Решение задач всех четырёх вариантов основывается чаще всего на одном алгоритме.

УДК 004(075.3=161.3=161.1)  
ББК 32.81я721

ISBN 978-985-471-688-6

© НМУ «Национальный институт образования», 2010  
© Оформление. РУП «Издательство “Адукацыя і выхаванне”», 2010

## ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ С ПОМОЩЬЮ КОМПЬЮТЕРА

### Вопросы для повторения

1. Какие этапы решения задачи вам известны?
2. Что такое алгоритм?
3. Какие виды алгоритмов вам известны?
4. Что такое программа?
5. Из каких разделов состоит программа на языке Pascal?
6. Каким образом можно осуществлять отладку программ в языке Pascal?
7. Что такое тестирование?
8. Что такое оператор?
9. Какие операторы языка Pascal вам известны?
10. Зачем нужен оператор присваивания? Какой вид он имеет?
11. Что может быть записано в правой части оператора присваивания?
12. Что такое переменная?
13. Что такое константа?
14. Какие правила применяются для создания имён переменных?
15. Что такое идентификатор?
16. Почему знак умножения всегда выписывают явно (пишут  $a*t$ , а не  $at$ )?
17. Как описываются переменные в языке Pascal?
18. Какие стандартные числовые типы языка Pascal вам известны?

19. Что вам известно о соответствии типов переменных в языке Pascal?

20. Какие арифметические операции можно выполнять в языке Pascal?

21. Что вам известно о приоритете арифметических действий в языке Pascal?

22. Какие математические функции есть в языке Pascal?

23. Какая команда служит для ввода данных?

24. Какой формат записи имеет команда ввода?

25. Чем команда ввода *read* отличается от команды ввода *readln*?

26. Какая команда служит для вывода данных?

27. Какой формат записи имеет команда вывода?

28. Чем команда вывода *write* отличается от команды вывода *writeln*?

29. Как распечатать текст?

30. Что такое форматный вывод?

31. Как напечатать значение переменной на фиксированном количестве позиций?

32. Как напечатать значение переменной с фиксированным количеством знаков после запятой?

33. Как разместить комментарии в программе?

### Задания для практических работ

**Задание 1.** Даны  $x, y, z$ . Напишите программу для вычисления  $a, b$ .

$$a = \frac{2x + 3y - z}{3 + 2x} \quad b = \frac{\sqrt{x} + 4}{|y - 2|}$$

### Этапы выполнения задания (a)

I. Определение исходных данных: переменные  $x, y, z$ .

II. Определение результатов: переменная  $a$ .

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление значения выражения.
3. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *real*. (Переменные  $x, y, z$  могут быть и целого типа, а переменная  $a$  — только вещественного типа, поскольку в выражении используется операция деления.)

V. Программа:

```
Var x,y,z,a: real;
```

```
Begin
```

```
  Write('введите значение x=');
```

```
  Readln(x);
```

```
  Write('введите значение y=');
```

```
  Readln(y);
```

```
  Write('введите значение z=');
```

```
  Readln(z);
```

```
  a:=(2*x+3*y-z)/(3+2*x);
```

```
  Writeln('a=',a:7:3);
```

```
  Readln;
```

```
End.
```

VI. Тестирование программы.

1. Запустите программу и введите значения:

$x=3$

$y=2$

$z=0$

Проверьте. Результат должен быть следующим:

$a=1.333$

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу:

	x	y	z	a
1	2	-5	1	
2	2.362	4.258	-7.458	
3	-1.2358	-4.554	5.562	

3. Подберите такие значения  $x, y, z$ , чтобы значение  $a = 2.000$ .

4. Что изменится и почему, если в программе заменить команду `Writeln('a=',a:7:3);` на команду `Writeln(a);`?

5. Как распечатать только целую часть результата?

### Этапы выполнения задания (b)

I. Определение исходных данных: переменные \_\_\_\_\_.

II. Определение результатов: переменная \_\_\_\_\_.

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление значения выражения.
3. Вывод результата.

IV. Описание переменных:  
все переменные, определённые для решения задачи, имеют тип \_\_\_\_\_.

V. Программа:

Var x,y,a: real;

Begin

Write('vvedite znachenie \_\_\_');

Readln(\_\_\_);

Write('vvedite znachenie \_\_\_');

Readln(\_\_\_);

b:=\_\_\_\_\_;

writeln(\_\_\_\_\_);

Readln;

End.

VI. Тестирование программы.

1. Запустите программу и введите значения:

$x=9$

$y=-5$

2. Проверьте. Результат должен быть следующим:

$a=1.000$

Проверить правильность вычислений можно на калькуляторе.

3. Заполните таблицу:

	$x$	$y$	$b$
1	16	-5	
2	2.362	4.258	
3	-1.2358	-4.554	

4. Подберите такие значения  $x$ ,  $y$ , чтобы значение  $b = 2.000$ .

5. Можно ли объединить в одну программу вычисление двух выражений? Как это сделать?

### Индивидуальные задания

$$1) a = y + \frac{x}{y^2 + x^2 / (y^3 / 3)};$$

$$b = 1 + (y - x) + \frac{(y - x)^3}{3};$$

$$2) a = \frac{3 + x^2 - y^3}{x - y^2} (x + 3)^2;$$

$$b = (1 + y) \frac{x + y / (x^2 + 4)}{x - 2 + 1 / (x^2 + 4)};$$

$$3) a = \frac{1 + (x + y)^2}{2 + x - 2x / (1 + x^2 y^2)};$$

$$b = \frac{1}{2} + \frac{x - z^3}{xy + z^2} (x - y);$$

$$4) a = \sqrt{x + y} \frac{x - y}{2 + x + y^2};$$

$$b = \frac{\sqrt{x - 1} - \sqrt[4]{x}}{1 + x^2 / 2 + y^3 / 4}.$$

**Задание 2.** Напишите программу для решения задачи, используя операции *div* и *mod*.

Задано двузначное число. Поменяйте местами первую и вторую цифры числа. Например, при вводе числа 25 получаем число 52.

### Этапы выполнения задания

I. Определение исходных данных: переменная  $a$  (исходное число).

II. Определение результатов: переменная  $b$  (преобразованное число).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для преобразования числа необходимо выполнить следующие действия:

а) выделить из исходного числа вторую цифру и сохранить её в переменной  $a1$ . Для выделения цифры нужно найти остаток от деления исходного числа на 10;

б) выделить из исходного числа первую цифру и сохранить её в переменной  $a2$ . Для выделения цифры нужно найти целую часть от деления исходного числа на 10;

в) получить искомое число  $b$ . Для этого нужно умножить на 10 значение переменной  $a1$  и к полученному произведению прибавить значение переменной  $a2$ .

3. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *integer*.

V. Программа:

```
Var a,b,a1,a2: integer;
```

```
Begin
```

```
Write('введите значение a= ');
```

```
Readln(a);
```

```
a1:=a mod 10; {Выделение последней циф-  
ры в двузначном числе}
```

```
a2:=a div 10; {Выделение первой цифры  
в двузначном числе}
```

```
b:=a1*10+a2;
```

```
Write('искомое число = ',b);
```

```
readln;
```

```
End.
```

VI. Тестирование.

1. Запустите программу и введите значение  $a=25$ .

Проверьте. Результат должен быть следующим:  
искомое число = 52

2. Заполните таблицу:

	$a$	$b$
1	15	
2	46	
3	63	
4		
5		

3. Для каких значений переменной  $a$  значение переменной  $b$  будет таким же? Добавьте в таблицу такое значение.

4. Всегда ли в результате выполнения программы получим двузначное число, если исходное число было двузначным? Почему? Приведите пример и добавьте его в таблицу.

5. Каким будет результат, если ввести однозначное число?

6. Попробуйте ввести трёхзначное число (например, 125). Объясните получившийся результат.

### *Индивидуальные задания*

1. Задано двузначное число. Уменьшите количество десятков в числе на один.

2. Задано двузначное число. Получите новое число, в котором количество десятков совпадает с количеством десятков в исходном числе, а количество единиц равно среднему арифметическому цифр исходного числа.

3. Заданы два двузначных числа. Поменяйте в них цифры, стоящие в разряде единиц. Например, для чисел 13 и 28 должны получить 18 и 23.

4. Заданы два двузначных числа. Получите новое число, в котором цифра в каждом разряде получается как среднее арифметическое цифр первого и второго числа, стоящих в соответствующих разрядах. Например, для чисел 28 и 45 должны получить 36.

**Задание 3.** Напишите программу для решения геометрической задачи. Формулы можно посмотреть в приложении.

Задан квадрат с длиной стороны  $a$ . Найдите его площадь и длину диагонали.

### *Этапы выполнения задания*

I. Определение исходных данных: переменная  $a$  (длина стороны).

II. Определение результатов: переменные  $S$  (площадь) и  $d$  (длина диагонали).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Вычисление значений площади по формуле  $S = a^2$  и длины диагонали по формуле  $d = a\sqrt{2}$ .

3. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *real*.

V. Программа:

```
Var a,S,d: real;
```

```
Begin
```

```
  Write('введите значение a=');
```

```
  Readln(a);
```

```
  S:=SQR(a);
```

```
  D:=a*SQR(2);
```

```
  Write('площадь=',S:7:3);
```

```
  Write('длина диагонали=',d:7:3);
```

```
  readln;
```

```
End.
```

VI. Тестирование.

1. Запустите программу и введите значение  $a=2$

Проверьте. Результат должен быть следующим:

```
S=4.000 d=2,828
```

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу:

	$a$	$S$	$d$
1	5		
2	2.35		
3	0.129		

3. Можно ли подобрать такое значение  $a$ , чтобы значение  $d$  содержало нули в дробной части?

4. Что нужно изменить в программе, чтобы значения  $a$  и  $d$  выводились в разных строчках?

5. Какие ещё возможны варианты записи формулы для вычисления площади?

*Совет:* при составлении программы для решения геометрической задачи сначала следует решить задачу математически и только потом писать программу.

### Индивидуальные задания

1. Даны гипотенуза и катет прямоугольного треугольника. Найдите второй катет и радиус вписанной окружности.

2. Треугольник задан координатами своих вершин. Вычислите его площадь и периметр.

3. Вычислите площадь кольца, внутренний радиус которого равен  $R_1$ , а внешний —  $R_2$ .

4. Треугольник задан длинами сторон  $a$ ,  $b$ ,  $c$ . Найдите длины высоты, медианы и биссектрисы этого треугольника, опущенных на сторону  $a$ .

**Задание 4.** Напишите программу для решения задачи. При решении воспользоваться операциями  $div$  и  $mod$ .

1. С начала месяца прошло  $t$  часов. Определите, какое сейчас число.

2. С начала месяца прошло  $t$  часов. Определите, какое сейчас число и сколько часов прошло с начала суток.

3. Идёт  $k$ -я секунда суток. Определите, который сейчас час (в часах, минутах и секундах).

4. С начала года прошло  $t$  секунд. Определите, какой сегодня день (число и месяц) и который сейчас час (в часах, минутах и секундах).

**Задание 5.** Напишите программу для решения задачи.

1. Определите полное количество часов, прошедших от начала суток до того момента (в первой половине суток), когда часовая стрелка повернулась на  $f$  градусов ( $0 \leq f < 360$ ,  $f$  — целое число).

2. Определите полное количество часов и минут, прошедших от начала суток до того момента (в первой половине суток), когда часовая стрелка повернулась на  $f$  градусов ( $0 \leq f < 360$ ,  $f$  — целое число).

3. Определите  $f$  — угол (в градусах) между положением часовой стрелки в начале суток и её положением в  $h$  часов  $t$  минут ( $0 \leq h < 12$ ,  $0 \leq t < 60$ ).

4. Определите  $f$  — угол (в градусах) между положением часовой стрелки в начале суток и её положением в  $h$  часов  $t$  минут ( $0 \leq h < 12$ ,  $0 \leq t < 60$ ). На какой угол повернута минутная стрелка?

**Задание 6.** Напишите программу для решения задачи.

1. Дано действительное число  $a$ . Не пользуясь никакими другими арифметическими действиями, кроме умножения, получите  $a^9$  за четыре операции.

2. Дано действительное число  $a$ . Не пользуясь никакими другими арифметическими действиями, кроме умножения, получите  $a^{15}$  за пять операций.

3. Дано действительное число  $a$ . Не пользуясь никакими другими арифметическими действиями, кроме умножения, получите  $a^2$  и  $a^5$  и  $a^{17}$  за шесть операций.

4. Дано действительное число  $x$ . Не пользуясь никакими другими арифметическими действиями, кроме сложения, вычитания и умножения, вычислите  $2x^4 - 3x^3 + 4x^2 - 5x + 6$ . Разрешается использовать не более четырёх умножений и не более четырёх сложений и вычитаний.

**Задание 7.** Напишите программу для решения задачи.

Нарисуйте на экране прямоугольный треугольник с катетами, параллельными сторонам экрана. Длины катетов и координаты прямого угла вводятся в клавиатуры.

### Этапы выполнения задания

I. Определение исходных данных: переменные  $a$  и  $b$  (катеты), переменные  $x$  и  $y$  (координаты прямого угла).

II. Определение результатов: рисунок прямоугольного треугольника, удовлетворяющий условию задачи.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для того чтобы построить треугольник, нужно выполнить следующие действия:

- провести линии из точки с координатой  $(x, y)$  в точки с координатами  $(x + a, y)$  и  $(x, y + b)$ ;
- соединить линией точки  $(x + a, y)$  и  $(x, y + b)$ .

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *integer*.

V. Программа:

```
Uses graphABC;
```

```
Var a,b,x,y:integer;
```

```
Begin
```

```
  writeln('введите координаты');
```

```
  readln(x,y);
```

```
  writeln('введите длины катетов');
```

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

```
  line(x,y,x+a,y);
```

```
  line(x,y,x,y+b);
```

```
  line(x+a,y,x,y+b);
```

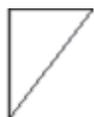
```
End.
```

## VI. Тестирование.

1. Запустите программу и введите значения:  
координаты 100 100

катеты 70 120

Проверьте. Результат должен быть следующим:



2. Поэкспериментируйте с программой, вводя различные значения.

3. Что произойдёт, если ввести отрицательные значения катетов? Объясните результат.

4. Что произойдёт, если ввести отрицательные значения координат? Объясните результат.

5. Закрасьте треугольник.

### *Индивидуальные задания*

1. Нарисуйте на экране квадрат со сторонами, параллельными сторонам экрана. Длина стороны квадрата и координаты верхнего левого угла вводятся в клавиатуры.

2. Нарисуйте на экране два квадрата со сторонами, параллельными сторонам экрана. Один из квадратов лежит в другом так, что квадраты имеют общий центр (точка пересечения диагоналей). Длины сторон квадратов и координаты центра вводятся в клавиатуры.

3. Нарисуйте на экране квадрат со сторонами, параллельными сторонам экрана. Длина стороны

квадрата и координаты верхнего левого угла вводятся в клавиатуры. В квадрат вписать окружность.

4. Нарисуйте на экране равнобедренный треугольник. Длины основания, высоты и координаты вершины треугольника вводятся в клавиатуры. (Треугольник расположите так, чтобы его основание было параллельно одной из сторон экрана.) Проведите среднюю линию треугольника, параллельную основанию.

## ЛОГИЧЕСКИЕ ЗАДАЧИ

### Вопросы для повторения

1. Какую величину называют логической?
2. Какие логические операции вам известны?
3. Что такое составное условие?
4. Каковы правила записи составных условий в оператор языка Pascal?
5. Как в языке Pascal записывается оператор выбора?
6. Для чего предназначен оператор выбора?
7. Что такое полная и сокращённая записи оператора выбора?
8. Какого типа должна быть переменная, значения которой выбирают с помощью оператора *case*?

### Задания для практических работ

**Задание 1.** Напишите программу для решения задачи.

Выдайте на экран значение true или false, в зависимости от того, является ли введённое число  $x$  чётным или нет.

### Этапы выполнения задания

I. Определение исходных данных: переменная  $x$  (введённое число).

II. Определение результатов: переменная  $t$  (логическая переменная).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Вычисление значения логической переменной. Число является чётным, если остаток от деления данного числа на 2 равен нулю. Значение переменной  $t$  будет определяться значением логического выражения  $x \bmod 2 = 0$ .

3. Вывод результата.

IV. Описание переменных:

переменная  $x$  — *integer* (или *real*, для данной задачи это не имеет значения), переменная  $t$  — *boolean*.

V. Программа:

```
Var x: integer;
```

```
t: boolean;
```

```
Begin
```

```
Write('введите значение x=');
```

```
Readln(x);
```

```
t:=x mod 2=0;
```

```
Write('чётность - ',t);
```

```
Readln;
```

```
End.
```

### VI. Тестирование.

1. Запустите программу и введите значение

$x=2$

Проверьте. Результат должен быть следующим: четность — true

2. Заполните таблицу:

	$x$	$t$
1	5	
2	0	
3	-13	
4	-8	

3. В каком случае программа выдаёт значение false?

4. Сформулируйте обратную задачу: чтобы во всех случаях, для которых в исходной задаче получалось true, было бы false и наоборот, во всех случаях, для которых в исходной задаче получалось false, было бы true.

5. Внесите изменения в программу для решения обратной задачи.

6\*. В языке Pascal есть функция odd(x), которая выдаёт значение true, если число нечётное, и false, если число чётное. Измените программу для решения обратной задачи с использованием функции odd.

### Индивидуальные задания

1. Задано число  $x$ . Определите, является ли число  $x$  кратным 3 или нет. Выдайте на экран соответствующее сообщение.

2. Заданы два положительных числа:  $x$  и  $y$ . Определите, верно ли, что  $x$  делится на  $y$ , без остатка или нет. Выдайте на экран соответствующее сообщение.

3. Задано положительное число  $x$ , являющееся возрастом человека в годах. Определите, человек совершеннолетний или нет. Выдайте на экран соответствующее сообщение.

4. Задано неотрицательное число  $x$ . Определите, может ли это число быть школьной отметкой или нет. Выдайте на экран соответствующее сообщение.

**Задание 2.** Напишите программу для решения задачи.

Задано число  $x$ . Если число чётное, то нарисуйте на экране зелёный квадрат, а если нечётное, то красный круг.

#### *Этапы выполнения задания*

I. Определение исходных данных: переменная  $x$  (введённое число).

II. Определение результатов: рисунок, зависящий от значения  $x$ .

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Результат будет зависеть от того, истинным или ложным является значение логического выражения  $x \bmod 2 = 0$ . Для решения воспользуемся условным оператором.

3. Вывод результата.

IV. Описание переменных:  
переменная  $x$  — *integer*.

V. Программа:

```
uses graphABC;  
var x:integer;  
begin  
  writeln('введите значение x=');  
  readln(x);  
  if x mod 2=0 then  
  begin  
    setbrushcolor(clgreen);  
    rectangle(150,150,350,350);  
  end  
  else  
  begin  
    setbrushcolor(clred);  
    circle(400,300,100);  
  end;  
end;
```

End.

VI. Тестирование.

1. Запустите программу и введите значение  $x=2$

Проверьте. Результат должен быть следующим:



2. Заполните таблицу:

	1	2	3
$x$	13	0	-25
рисунок			

3. Найдите в программе составные операторы и объясните, зачем они здесь нужны.

4. Какие минимальные изменения нужно внести в программу, чтобы для чётных чисел рисовался красный круг, а для нечётных — зелёный квадрат?

5. Внесите в программу изменения так, чтобы пользователь мог вводить длину стороны квадрата или радиус круга по запросу: 'введи длину стороны квадрата' или 'введи радиус круга'. Запрос должен появиться после того, как введено число, определяющее тип фигуры.

#### **Индивидуальные задания.**

1. Задано число  $x$ . Если число нечётное, то нарисуйте на экране овал, в котором горизонтальный радиус больше вертикального, а если чётное — овал, в котором вертикальный радиус больше горизонтального. Овал должен быть закрашен горизонтальной штриховкой для введённого нечётного числа и вертикальной для чётного числа.

2. Задано число  $x$ . Нарисуйте на экране светофор для пешеходов. Если число  $x$  чётное, то должен гореть зелёный свет, а если нечётное, то — красный.

3. Задано число  $x$ . Нарисуйте на экране эллипс. Если число  $x$  положительное, то закрасьте только верхнюю половинку эллипса, если число  $x$  неотрицательное, то закрасьте нижнюю половинку эллипса.

4. Задано число  $x$ . Нарисуйте на экране круг, вписанный в квадрат, если число  $x$  положительное или квадрат, вписанный в круг, если число  $x$  неотрицательное.

**Задание 3.** Вычислите значение функции для заданного  $x$ .

$$f(x) = \begin{cases} -2, & \text{если } x < -2 \\ x^3, & \text{если } -2 \leq x \leq 2 \\ x + 2, & \text{если } x > 2 \end{cases}$$

#### **Этапы выполнения задания**

I. Определение исходных данных: переменная  $x$  (значение переменной).

II. Определение результатов: переменная  $f$  (значение функции).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Проверка значения  $x$  и вычисление значения функции, в зависимости от значения аргумента.

3. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *real*.

V. Программа:

```
Var x, f: real;
```

```
Begin
```

```
  Write('введите значение x=');
```

```
  Readln(x);
```

```
  If x<-2 then
```

```
    f:=-2
```

```
  else
```

```
    if x<=2 then
```

```
      f:=x*x*x
```

```
    else
```

```
      f:=x+2;
```

```
  Writeln('f=',f:7:3);
```

```
  readln;
```

```
End.
```

VI. Тестирование.

1. Запустите программу, введите значения и проверьте результат.

$x$	$f$
-5	-2.000
0.2	0.008
3.27	5.27

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу:

	$x$	$f$
1	-3.52	
2	-1.28	
3	23.48	

3. Добавьте в таблицу свои значения  $x$ . Предуспмотрите все возможные случаи вычисления значения функции. Сколько строк нужно добавить в таблицу?

4. Для заданной функции и фрагмента программы, вычисляющей значение этой функции, вставьте пропущенное:

$$f(x) = \begin{cases} x, & \text{если } x < -3 \\ 2x + \dots, & \text{если } \dots \leq x \leq 7 \\ \dots, & \text{если } x > \dots \end{cases}$$

```
      If x<... then
        y:=x
      else
        if x<=7 then
          ...:=2*x+7
        else
          y:=2*x-5;
```

5. Запишите функцию, значение которой вычисляется с помощью предложенного фрагмента программы.

```
If x< -3 then
```

```
  y:=-x
```

```
else
```

```
  if x<0 then
```

```
    y:=0
```

```
  else
```

```
    y:=x*x;
```

### Индивидуальные задания

1.  $f(x) = \begin{cases} x^2, & \text{если } x \geq 2 \\ 4, & \text{в остальных случаях} \end{cases}$

$$2. f(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 2 \\ \frac{1}{x^2 + 4x + 5}, & \text{в остальных случаях} \end{cases}$$

$$3. f(x) = \begin{cases} x - 7, & \text{если } x < -7 \\ \sqrt{49 - x^2}, & \text{если } -7 \leq x \leq 7 \\ -x + 7, & \text{в остальных случаях} \end{cases}$$

$$4. f(x) = \begin{cases} x + 4, & \text{если } x \leq 0 \\ x + 1, & \text{если } 0 < x < 1 \\ 2, & \text{если } 1 \leq x \leq 2 \\ 3x - 4, & \text{если } x > 2 \end{cases}$$

**Задание 4.** Напишите программу для решения задачи.

Проверьте, является ли введённое число трёхзначным, и выведите чётные цифры числа.

#### Этапы выполнения задания

I. Определение исходных данных: переменная  $a$  (трёхзначное число).

II. Определение результатов: переменные  $a1$ ,  $a2$ ,  $a3$  в том случае, если они чётные, или сообщение, что чётных цифр нет.

III. Алгоритм решения задачи.

1. Ввод исходного числа.

2. Проверка. Является ли число трёхзначным? Трёхзначное число больше 99 и меньше 1000 — значит, должны выполняться одновременно два условия:  $(a > 99)$  and  $(a < 1000)$ .

3. Если число трёхзначное, то выделяем цифры числа и проверяем каждую из них на чётность:

а) для выделения первой цифры (переменная  $a1$ ) трёхзначного числа необходимо найти целую часть от деления числа на 100;

б) для выделения второй цифры (переменная  $a2$ ) трёхзначного числа необходимо найти остаток от деления числа на 100 и от него найти целую часть при делении на 10;

в) для определения последней цифры (переменная  $a3$ ) трёхзначного числа необходимо найти остаток от деления числа на 10.

г) для проверки цифры на чётность нужно проверить, равен ли нулю остаток от деления цифры на 2.

4. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *integer*.

V. Программа:

```
Var a,a1,a2,a3: integer;
```

```
Begin
```

```
Write('введите значение a=');
```

```
Readln(a);
```

```
If (a>99) and (a<1000) then {используется составное условие}
```

```

Begin
  a1:=a div 100;
  a2:=a mod 100 div 10;
  a3:=a mod 10;
  if a1 mod 2=0 then
    Writeln('цифра ',a1,' чётная');
  if a2 mod 2=0 then
    Writeln('цифра ',a2,' чётная');
  if a3 mod 2=0 then
    Writeln('цифра ',a3, ' чётная');
  If (a1 mod 2=1) and (a2 mod 2=1) and
    (a3 mod 2=1) then
    Writeln('в числе нет чётных цифр');
End
Else
  Writeln('число не трёхзначное');
readln;
End.

```

#### VI. Тестирование.

1. Проверьте работу программы для следующих значений:

Ввод <i>a</i>	Вывод
246	Цифра 2 чётная; цифра 4 чётная; цифра 6 чётная
103	Цифра 0 чётная
537	В числе нет чётных цифр
25	Число не является трёхзначным

2. Заполните таблицу:

	<i>a</i>	Вывод
1	468	
2	476	
3	233	
4	441	
5	119	
6	1111	

3. Какие случаи расположения цифр в трёхзначном числе не были рассмотрены? Предложите свои тесты на эти случаи.

4. Как по-другому можно вычислить значение второй цифры трёхзначного числа?

5. Где в программе используются составные условия? Какие логические операции используются при этом?

6. Можно ли проверку цифр в числе записать следующим образом?

```

if a1 mod 2=0 then
  Writeln('цифра ',a1,' чётная')
else
  if a2 mod 2=0 then
    Writeln('цифра ',a2,' чётная')
  else
    if a3 mod 2=0 then
      Writeln('цифра ',a3, ' чётная')
    else
      Writeln('в числе нет чётных цифр');

```

Аргументируйте свой ответ.

### Индивидуальные задания

1. Дано натуральное число. Проверьте, является ли число двузначным, у которого первая цифра кратная второй.

2. Дано натуральное число. Проверьте, является ли оно четырёхзначным и содержит ли одинаковые цифры.

3. Дано натуральное число. Проверьте, является ли оно четырёхзначным палиндромом (т. е. равна ли первая цифра четвёртой, а вторая — третьей).

4. Дано натуральное число. Проверьте, является ли число шестизначным, «счастливым» (т. е. сумма первых трёх цифр равна сумме последних трёх).

**Задание 5.** Напишите программу для решения задачи.

1. Задано число  $x$ . Нарисуйте на экране компьютера круг с радиусом, равным  $x$ . Если  $x < 1$  или  $x > 300$ , то выдайте сообщение о том, что круг нарисовать нельзя.

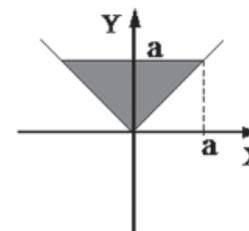
2. Задано число  $x$ . Нарисуйте на экране компьютера отрезок длиной, равной  $x$ . Если  $x < 1$  или  $x > 600$ , то выдайте сообщение о том, что отрезок нарисовать нельзя.

3. Заданы два числа:  $x$ ,  $y$ . Нарисуйте на экране компьютера прямоугольник с длинами сторон, равными  $x$  и  $y$ . Если  $x < 1$ , или  $x > 800$ , или  $y < 1$ , или  $y > 600$ , то выдайте сообщение о том, что прямоугольник нарисовать нельзя. За-

мените 800 и 600 значениями, соответствующими разрешению экрана (размерам окна).

4. Заданы два числа:  $x$ ,  $y$ . Нарисуйте на экране компьютера эллипс с горизонтальным и вертикальным радиусами, равными  $x$  и  $y$ . Если значения  $x$  и  $y$  таковы, что эллипс не поместится целиком в окне, или они не положительны, то выдайте сообщение о том, что эллипс нарисовать нельзя.

**Задание 6.** Определите, принадлежит ли точка  $A(x, y)$  данной области. Ответом должно служить слово «да» или «нет».



### Этапы выполнения задания

I. Определение исходных данных: переменные  $x$  и  $y$  (координаты точки).

II. Определение результатов: слово «да» или «нет», в зависимости от принадлежности точки закрашенной области.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Область образована тремя прямыми:  $y = x$ ,  $y = -x$ ,  $y = a$ . Точка принадлежит области, если

одновременно выполняются три условия:  $y \geq x$ ,  $y \geq -x$ ,  $y \leq a$ . Для того чтобы определить, какой из знаков поставить, достаточно взять произвольную точку, принадлежащую области, и посмотреть, как будут выполняться условия для этой точки. Например, при  $a = 5$  точка с координатами (1,3) принадлежит области, и выполняются следующие условия:  $3 \geq 1$ ,  $3 \geq -1$ ,  $3 \leq 5$ .

3. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *real*.

V. Программа:

Var x, y, a: real;

Begin

Write('введите значения координаты x=');

Readln(x);

Write('введите значения координаты y=');

Readln(y);

Write('введите значения a=');

Readln(a);

If (y>=x) and (y>=-x) and (y<=a) then

    Writeln('да')

else

    Writeln('нет');

readln;

End.

VI. Тестирование.

1. Запустите программу, введите значения и проверьте результат.

$x$	$y$	$a$	результат
1	3	5	да
-3	1	5	нет
-2	-2	5	нет

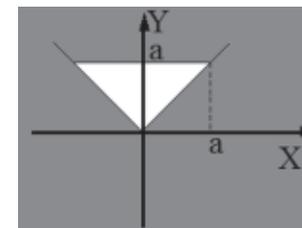
2. Заполните таблицу:

	$x$	$y$	$a$	результат
1	-1.3	2.8	4	
2	0.5	10.2	8	
3	0	0	5	

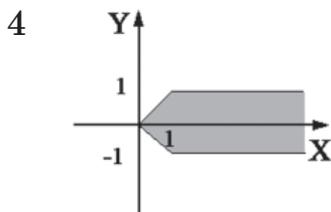
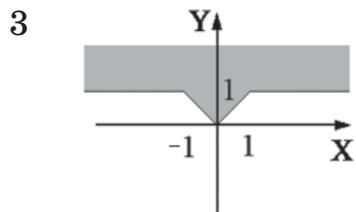
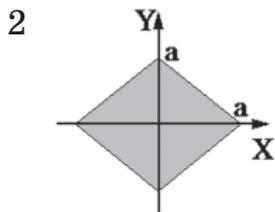
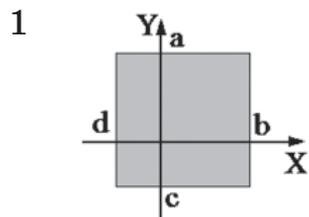
3. Добавьте в таблицу свои значения. Предусмотрите различные возможные случаи расположения точки относительно области.

4. Проверьте точки, лежащие на границе области. Какой ответ выдаёт программа? Какие изменения нужно внести в программу, если считать, что точки, лежащие на границе, не принадлежат области?

5. Как изменится условие принадлежности точки области, если областью считать всю плоскость, за исключением рассмотренного «треугольника».



## Индивидуальные задания



**Задание 7.** Напишите программу для решения задачи.

1. Посчитайте количество чётных и количество нечётных среди чисел  $a, b, c$ .

2. Определите, имеется ли среди чисел  $a, b, c$  хотя бы одна пара равных между собой чисел. Укажите какие.

3. Дан номер года. Определите, является ли он високосным (год является високосным, если его номер делится на 4, за исключением тех, которые делятся на 100 и не делятся на 400).

4. Даны три числа:  $a, b, c$ . Проверьте, могут ли они быть датой (например, 3, 7, 1972 — дата). Для определения високосного года см. вариант 3.

**Задание 8.** Напишите программу для решения геометрической задачи.

1. Определите, является ли треугольник со сторонами  $a, b, c$  равнобедренным. Если «да», то что является основанием, а что — боковыми сторонами.

2. Определите, является ли треугольник со сторонами  $a, b, c$  прямоугольным. Если «да», то что является гипотенузой, а что — катетами.

3. Даны три числа:  $a, b, c$ . Определите, существует ли треугольник со сторонами  $a, b, c$ . Если он существует, определите его вид (двумерный, остроугольный, тупоугольный).

4. Даны действительные положительные числа:  $a, b, c, d$ . Выясните, можно ли прямоугольник со сторонами  $a, b$  уместить внутри прямоугольника со сторонами  $c, d$  так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника.

**Задание 9.** Напишите программу для вычисления суммы, имеющей своими слагаемыми

$$a_n = \frac{n + 2^n}{3^n}.$$

Вычислять до тех пор, пока  $|a_n| \geq E$  ( $E$  вводится).

### Этапы выполнения

I. Определение исходных данных: переменная  $E$  (точность вычислений).

II. Определение результатов: переменная  $S$  (значение суммы).

### III. Алгоритм решения задачи.

#### 1. Ввод числа $E$ .

2. Для вычисления суммы необходимо последовательно прибавлять очередное слагаемое, удовлетворяющее условию задачи, к уже полученной сумме:

а) так как количество слагаемых заранее не известно, то для вычисления суммы нужно воспользоваться циклом с условием (`while` или `repeat`). Начальное значение суммы равно 0 (число 0 не может изменить конечное значение суммы);

б) текущее значение слагаемого будем хранить в переменной  $a$ . Значение  $a$  зависит от значения  $n$  — счётчика слагаемых в сумме. Переменная  $n$  будет изменяться на 1 до тех пор, пока выполняется условие задачи. Начальное значение равно 0.

в) отдельно нужно получать  $2^n$  и  $3^n$ . Для этого заведём дополнительные переменные  $d$  и  $g$ . Начальное значение переменных равно 1 (число 1 не может изменить конечное значение произведения). Для каждого следующего слагаемого предыдущее значение переменной  $d$  увеличивается в два раза, переменной  $g$  — в три раза.

#### 3. Вывод результата $S$ .

### IV. Описание переменных:

переменная  $n$  имеет тип *integer*;

переменные  $E$ ,  $a$ ,  $S$ ,  $d$  имеют тип *real*.

### V. Программа:

```
Var n: integer;
    E,a,S,d,g: real;
Begin
  Write('введите значение E=');
  Readln(E);
  S:=0;
  n:=0;
  d:=1;
  g:=1;
  Repeat
    d:=d*2;
    g:=g*3;
    n:=n+1;
    a:=(n+d)/g;
    S:=S+a;
  Until abs(a)<E;
  writeln('сумма=',S:5:3);
  readln;
End.
```

### I. Тестирование.

1. Проверьте работу программы для следующих значений:

Ввод E	Вывод S
0.1	2.569
0.0001	2.750

2. Заполните таблицу:

	E	S
1	0.01	
2	0.00001	

3. Замените в программе цикл `repeat` на цикл `while`.

4. Какое количество слагаемых даёт искомую сумму? Распечатайте это количество. Измените формат печати суммы так, чтобы после десятичной точки выводилось 10 цифр.

5. Задайте количество слагаемых ( $k$ ) в сумме и определите, достаточно ли этих слагаемых для получения погрешности вычислений, не превосходящей  $E$ .

### Индивидуальные задания

1)  $a_n = \frac{1}{4n}$ ;

2)  $a_n = \frac{1}{5^n}$ ;

3)  $a_n = \frac{n+1}{n!}$ ;

4)  $a_n = \frac{2^n}{n!}$ ;

5)  $a_n = (-1)^n \frac{2^{3n}}{3^n + n!}$ .

**Задание 10.** Напишите программу для решения задачи.

### Образец выполнения задания

Задача: найти первый меньший  $10^{-3}$  элемент последовательности  $\sin x, \sin 2x, \sin 3x, \dots, \sin kx$  и его номер.

### Этапы выполнения задания

I. Определение исходных данных: переменная  $x$ .

II. Определение результатов: переменные  $a$  (первый элемент в последовательности меньший  $10^{-3}$ ) и  $n$  (номер этого элемента в последовательности).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Определение начального значения номера элемента последовательности ( $k := 0$ ).

3. В цикле вычисляем очередной элемент последовательности  $a$  и проверяем условие  $a < 0.001$ . Цикл заканчивает работу при выполнении условия.

4. Вывод результата.

I. Описание переменных:

переменные  $a$  и  $x$  имеют тип *real*, переменная  $k$  — тип *integer*.

II. Программа:

```
Var x,a: real;
```

```
k: integer;
```

```
Begin
```

```
Write('введите значение x');
```

```
Readln(x);
```

```
n:=0;
```

```
repeat
```

```
  k:=k+1;
```

```
  a:=sin(k*x);
```

```
until a<0.001;
```

```
writeln('искомый элемент - ',a:6:3,' имеет номер ',k);
```

```
readln;
```

```
End.
```

### III. Тестирование.

1. Запустите программу и введите значение  $x=1$

Проверьте. Результат должен быть следующим: искомый элемент  $-0.756$  имеет номер 4

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу:

	$x$	$a$	$k$
1	-1		
2	25.32		
3	421		

3. Добавьте в таблицу свои значения  $x$ . Попробуйте подобрать такие значения  $x$ , чтобы  $n = 2, 3, 5$ .

4. Попробуйте найти такое  $x$ , чтобы значение  $n$  было большим, чем для  $x = 421$ .

5. При тестировании данной программы перебрали все целые значения  $x$  от 1 до 10 000. Какое максимальное значение  $n$  было получено?

6. Найдите такое значение  $x$ , при котором результат получится для положительного значения  $a$ . (Подсказка: это значение может быть не целым.)

#### *Индивидуальные задания*

1. Найдите номер первого элемента, меньшего  $10^{-4}$ , в последовательности  $\sin 1, \sin \sin 1, \sin \sin \sin 1, \dots$

2. Начиная с какого номера разность между соседними элементами последовательности  $\cos 1, \cos \cos 1, \cos \cos \cos 1, \dots$  будет меньше  $10^{-4}$ ?

3. Дано действительное число  $a$ . Найдите среди чисел  $1, 1+1/2, 1+1/2+1/3, \dots$  первое, большее  $a$ .

4. Дано целое число  $m > 1$ . Получите наибольшее целое  $k$ , при котором  $4^k < m$ .

5. Дано натуральное число  $n$ . Получите наименьшее число вида  $r!$ , ( $r \in \mathbb{N}$ ), превосходящее  $n$ .

**Задание 11.** Напишите программу для решения задачи.

Заполните графическое окно окружностями радиуса 10.

#### *Этапы выполнения задания*

I. Результат работы программы не зависит от исходных данных.

II. Определение результатов: рисунок, изображающий окружности по всему экрану.

III. Алгоритм решения задачи.

1. Данная задача является обобщением задачи из задания 1 предыдущего урока. Все действия, которые выполнялись в этой задаче, следует повторить для нескольких рядов. Количество рядов будет определяться высотой экрана.

2. Радиус  $r = 10$ . Определяем вертикальное положение центра первой окружности  $y = 11$ .

3. Вертикальное положение любой другой окружности, удовлетворяющей условию задачи, будет определяться координатой  $y$ . Для каждого

вертикального положения в цикле, пока  $y$  не станет большим, чем вертикальный размер окна, выполняем следующее:

а) задаём горизонтальное положение координаты центра окружности  $x = 11$ ;

б) в цикле, изменяющем значение  $x$ , рисуем окружность и изменяем горизонтальное положение окружности. Цикл должен завершиться, когда значение координаты  $x$  станет большим, чем горизонтальный размер окна;

в) изменяем вертикальное положение центра окружности.

IV. Описание переменных:

переменные  $x$ ,  $y$ ,  $r$  имеют тип *integer*.

V. Программа:

```
Uses graphABC;
```

```
Var x,y,r:integer;
```

```
Begin
```

```
  r:=10;
```

```
  y:=11;
```

```
  while y<= WindowHeight do
```

```
  begin
```

```
    x:=11;
```

```
    while x<= WindowWidth do
```

```
    begin
```

```
      circle(x,y,r);
```

```
      x:=x+20;
```

```
    end;
```

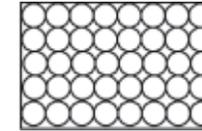
```
    y:=y+20;
```

```
  end;
```

```
End.
```

## VI. Тестирование.

1. Запустите программу, проверьте. Результат должен быть следующим:



2. Какие алгоритмические структуры использовались при написании данной программы?

3. Измените в программе значение  $r = 10$  на  $r = 30$ . Почему получился такой рисунок? Поэкспериментируйте со значениями радиуса.

4. Какие изменения нужно внести в программу, чтобы экран заполнялся окружностями радиусом 30?

5. Внесите изменения в программу так, чтобы все окружности были красными или разноцветными.

6\*. Внесите в программу изменения так, чтобы экран можно было заполнять окружностями введённого радиуса.

**Задание 12.** Напишите программу для решения задачи с использованием оператора *case*.

Вводится время в часах от 0 до 23. Определите, к какой части суток относится данное время.

### Этапы выполнения задания

I. Определение исходных данных: переменная  $t$  (время).

II. Определение результатов. Сообщение: утро, день, вечер или ночь, в зависимости от введённого времени.

III. Алгоритм решения задачи.

1. Ввод исходного числа.

2. Выбор значения переменной  $t$  из возможных диапазонов времени:

а) от 23 до 5 — ночь;

б) от 6 до 10 — утро;

в) от 11 до 17 — день;

г) от 18 до 22 — вечер.

3. Вывод результата.

IV. Описание переменных:

переменная  $t$  имеет тип *integer*.

V. Программа:

```
Var t: integer;
```

```
Begin
```

```
  Write('введите значение t=');
```

```
  Readln(t);
```

```
  Case t of
```

```
    23, 0..5: Writeln('Ночь');
```

```
    6..10: Writeln('Утро');
```

```
    11..17: Writeln('День');
```

```
    18..22: Writeln('Вечер');
```

```
    else Writeln('ошибочный ввод');
```

```
  end;
```

```
  readln;
```

```
End.
```

VI. Тестирование.

1. Проверьте работу программы для следующих значений:

Ввод $t$	Вывод
2	Ночь
18	Вечер

2. Заполните таблицу, предусмотрев проверку всех возможных ситуаций. Сколько строк должно быть в таблице?

	$t$	Вывод
1		
2		
3		

3. Как будет реагировать программа на значение  $t = 30$ , если из программы убрать блок *else*?

4. В каких случаях следующая программа решения этой же задачи будет работать некорректно?

```
Var t: integer;
```

```
Begin
```

```
  Write('введите значение t=');
```

```
  Readln(t);
```

```
  If (t<0) or (t>23) then
```

```
    Writeln('ошибочный ввод');
```

```
  else
```

```
    Case t of
```

```
      23, 0..5: Writeln('Ночь');
```

```
      6..10: Writeln('Утро');
```

```

11..17: Writeln('День');
else Writeln('Ночь');

end;
readln;
End.

```

Аргументируйте свой ответ.

5. Решите данную задачу с использованием неполной записи оператора *case*.

### *Индивидуальные задания*

1. Введите число от 1 до 7, определяющее день недели. Дайте название этого дня (1 — понедельник, 2 — вторник, ..., 7 — воскресенье).

2. Введите число от 1 до 10. Дайте название этого числа (1 — один, 2 — два, ..., 10 — десять).

3. Введите число от 1 до 12, определяющее месяц года. Определите, к какой поре года принадлежит месяц.

4. Дано натуральное число  $N$  ( $N < 100$ ), определяющее возраст человека в годах. Дайте для этого числа наименование: «год», «года», «лет».

5. Введите число от 1 до 100. Дайте название этого числа (1 — один, 2 — два, ..., 100 — сто).

## **ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ**

### **Вопросы для повторения**

1. Что такое вспомогательный алгоритм? Для чего он предназначается?

2. Какие виды подпрограмм есть в языке Pascal?

3. Чем подпрограмма-функция отличается от подпрограммы-процедуры?

4. Как описываются процедуры?

5. Как обращаются к процедуре?

6. Как описываются функции?

7. Как обращаются к функции?

8. Что такое формальные параметры? Как они описываются?

9. Что такое фактические параметры?

10. Какое соответствие должно быть между фактическими и формальными параметрами?

11. Для чего используются *var*-параметры?

12. Какой тип могут иметь параметры?

13. Что такое локальные и глобальные переменные?

14. Какова область действия локальных переменных? глобальных переменных?

### **Задания для практических работ**

**Задание 1.** Напишите программу для решения задачи. Оформите вычисление *min* и *max* из двух чисел в виде функций.

Даны три числа. Найдите попарные произведения этих чисел. Вычислите минимальное и максимальное из этих значений.

### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $x, y, z$ .

II. Определение результатов: переменные *minpr*, *maxpr*, обозначающие минимальное и максимальное произведения соответственно.

### III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Находим попарные произведения чисел  $x$ ,  $y$ ,  $z$ . Таких произведений будет три:  $xy$ ,  $yz$ ,  $xz$ . Для нахождения минимального (или максимального) значения из трёх чисел воспользуемся следующим свойством:

$$\min(a1, a2, a3) = \min(\min(a1, a2), a3).$$

3. Вывод результата.

### IV. Описание переменных:

все переменные имеют тип *integer*.

### V. Программа:

```
Var x, y, z, a1, a2, a3, minpr, maxpr, s: integer;
```

```
Function min(c, d: integer): integer;
```

```
Begin
```

```
  if c < d then
```

```
    min := c
```

```
  else
```

```
    min := d;
```

```
End;
```

```
Function max(c, d: integer): integer;
```

```
Begin
```

```
  if c > d then
```

```
    max := c
```

```
  else
```

```
    max := d;
```

```
End;
```

```
Begin
```

```
  Write('введите x=');
```

```
  Readln(x);
```

```
  Write('введите y=');
```

```
  Readln(y);
```

```
  Write('введите z=');
```

```
  Readln(z);
```

```
  a1 := x * y;
```

```
  a2 := y * z;
```

```
  a3 := x * z;
```

```
  s := min(a1, a2);
```

```
  minpr := min(s, a3);
```

```
  writeln('мин. произведение = ', minpr);
```

```
  s := max(a1, a2);
```

```
  maxpr := max(s, a3);
```

```
  writeln('максим. произведение = ', maxpr);
```

```
  readln;
```

```
End.
```

### VI. Тестирование.

1. Запустите программу и введите значения:

$x = 3$

$y = -2$

$z = -4$

Проверьте. Результат должен быть следующим:

минимальное произведение = -12

максимальное произведение = 8

2. Заполните таблицу:

	$x$	$y$	$z$	Минимальное произведение	Максимальное произведение
1	5	1	4		
2	2	-5	3		
3	-5	-12	-4		

3. Добавьте в таблицу свои значения. Проверьте ситуацию, в которой одна из переменных (или не одна) равна нулю.

4. Что нужно изменить в программе, для того чтобы можно было работать с дробными числами? В каких случаях полученные произведения будут больше (меньше) введённых чисел?

5. Добавьте в программу проверку: какое из значений будет больше  $\max(x, y, z)$  или  $\max_{pr}$ . (Аналогично — меньше —  $\min(x, y, z)$  или  $\min_{pr}$ .)

### Индивидуальные задания

1. Найдите  $\min$  и  $\max$  из трёх чисел.
2. Найдите  $\min$  и  $\max$  среди четырёх чисел.
3. Даны три различных числа. Какое из этих чисел не является ни минимальным, ни максимальным?
4. Даны четыре различных числа. Определите наибольшее и наименьшее среди тех двух чисел, которые не равны  $\min$  и  $\max$ .

**Задание 2.** Напишите программу для решения задачи. Оформите вычисление указанных функций в качестве подпрограмм.

Дано действительное число  $x$ . Получить

$$f(5x, -3) + f(x^3, 5x), \text{ где } f(a, b) = \frac{\min(a, b)}{\max(a, b)}$$

### Этапы выполнения задания

I. Определение исходных данных: переменная  $x$ .

II. Определение результатов: переменная  $S$  (значение искомого выражения).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для вычисления значения выражения опишем функцию  $f$ . Для вычисления значения функции опишем функции для вычисления  $\min$  и  $\max$  из двух чисел.

3. Вывод результата.

IV. Описание переменных: переменные  $x$  и  $S$  имеют тип *real*.

V. Программа:

```

Var  x, S: real;
Function min(c,d:real):real;
Begin
  if c<d then
    min:=c
  else
    min:=d;
End;

```

```

Function max(c,d:real):real;
Begin

```

```

if c>d then
  max:=c
else
  max:=d;
End;

```

```

Function f(a,b:real):real;
Begin
  F:=min(a,b)/max(a,b)
End;

```

```

Begin
  Write('введите x=');
  Readln(x);
  S:=f(5*x,-3)+f(x*x*x,5*x);
  writeln('значение выражения= ',S:5:2);
  readln;
End.

```

#### VI. Тестирование.

1. Запустите программу и введите значение  $x = 3$

Проверьте. Результат должен быть следующим:

значение выражения=0.36

2. Заполните таблицу:

	$x$	Значение выражения
1	1	
2	0.2	
3	-0.5	

3. Добавьте в таблицу свои значения  $x$ . Подберите такие значения  $x$ , чтобы получить значение суммы больше 100, меньше -100.

4. Подберите два таких значения  $x$  (положительное и отрицательное), чтобы выполнялось неравенство  $|S| < 10^{-3}$ .

5. Как описать функцию  $f$ , не используя описаний для функций  $\min$  и  $\max$ ? Какое из решений оптимальное? Почему?

#### *Индивидуальные задания*

1. Дано действительное число  $x$ . Получите:

$$f(2x, 3) + f(3, 5 - x), \text{ где } f(a, b) = \frac{2a - b}{5 - a}.$$

2. Даны действительные числа  $s$  и  $t$ . Получите:

$$f(1, s) + f(t, 2) \text{ где } f(a, b) = \frac{a^2 + b^2}{(a + b)^2}.$$

3. Даны действительные числа  $s$  и  $t$ . Получите:

$$f(t, -2s, 1.17) + f(2.2, t, t - s),$$

$$\text{где } f(a, b, c) = \frac{2a - b + \sqrt{c}}{5 + |c|}.$$

4. Даны действительные числа  $s$  и  $t$ . Получите:

$$f(1.2, s) + f(t, s) - f(2s - 1, st),$$

$$\text{где } f(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}.$$

**Задание 3.** Напишите программу для решения геометрической задачи.

Выпуклый четырёхугольник задан координатами своих вершин. Найдите периметр и площадь четырёхуголь-

ника. Вычисление длины отрезка и площади треугольника оформите в виде подпрограмм.

### Этапы выполнения задания

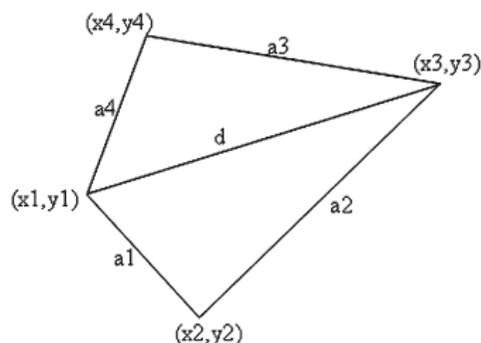
I. Определение исходных данных:  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$  (координаты вершин).

II. Определение результатов:  $P$  и  $S$  — периметр и площадь четырёхугольника.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Периметр четырёхугольника — сумма длин его сторон. Длина стороны — длина отрезка, соединяющего вершины. Вычисление длины отрезка оформляем в виде функции *dlin*.



3. Для вычисления площади проведём диагональ, соединяющую вершины с координатами  $(x_1, y_1)$  и  $(x_3, y_3)$  (или  $(x_2, y_2)$  и  $(x_4, y_4)$ ), четырёхугольник разобьётся на два треугольника. Площадь четырёхугольника — сумма площадей двух треугольников. Для вычисления площади треугольника нужно воспользоваться формулой Герона,

так как для треугольников известны длины сторон. Обозначения вспомогательных переменных см. на рисунке. Вычисление площади треугольника оформляем в виде функции *plos*.

4. Вывод результата.

IV. Описание переменных:

все переменные в программе имеют тип *real*.

V. Программа:

```
Var x1,y1,x2,y2,x3,y3,x4,y4,a1,a2,a3,a4,d,S1,S2,S,P:
    real;
```

```
Function dlin(x,y,x0,y0:real):real;
```

```
Begin
```

```
    dlin:=sqrt(sqr(x-x0)+sqr(y-y0));
```

```
End;
```

```
Function plos(a,b,c:real):real;
```

```
Var pl:real;
```

```
Begin
```

```
    pl:=(a+b+c)/2;
```

```
    plos:=sqrt(pl*(pl-a)*(pl-b)*(pl-c));
```

```
End;
```

```
Begin
```

```
    Write('введите координаты вершины 1 - ');
```

```
    Readln(x1,y1);
```

```
    Write('введите координаты вершины 2 - ');
```

```
    Readln(x2,y2);
```

```
    Write('введите координаты вершины 3 - ');
```

```
    Readln(x3,y3);
```

```
    Write('введите координаты вершины 4 - ');
```

```

Readln(x4,y4);
a1:=dlin(x1,y1,x2,y2); {длины сторон}
a2:=dlin(x2,y2,x3,y3);
a3:=dlin(x3,y3,x4,y4);
a4:=dlin(x4,y4,x1,y1);
d:=dlin(x1,y1,x3,y3);
P:=a1+a2+a3+a4; {периметр}
S1:=plos(a1,a2,d); {площади треугольников}
S2:=plos(a3,a4,d);
S:=S1+S2;
Writeln('периметр= ',P:5:2,' площадь=',S:5:2);
readln;
End.

```

#### VI. Тестирование.

1. Запустите программу, введите значения и проверьте результат.

	Координаты вершин				P	S
	1	2	3	4		
1	0 0	0 1	1 1	1 0	4.00	1.00
2	-5 -1	-3 1	3 5	2 -2	24.18	27.00

2. Заполните таблицу:

	Координаты вершин				P	S
	1	2	3	4		
1	3 -1	5 -3	3 -5	1 -3		
2	5.1 3.2	7.2 3.2	9.4 5.7	7.3 5.7		
3	2 -3	3.5 -4.2	10 12.3	-7.2 5.6		

3. Добавьте в таблицу свои значения, предварительно изобразив четырёхугольники на координатной плоскости. Вычислите площадь прямоугольника, трапеции.

4. В первом примере для тестирования периметр больше площади, во втором меньше. Попробуйте найти такой четырёхугольник (не квадрат), у которого периметр приближённо равен площади.

5. Можно ли использовать предложенный алгоритм для вычисления площади невыпуклого четырёхугольника? Почему? В каких случаях возможно использование данного алгоритма? Внесите в программу необходимые изменения.

#### Индивидуальные задания

1. Треугольник задан координатами своих вершин. Найдите периметр треугольника. Вычисление длины отрезка оформите в виде подпрограммы.

2. Четырёхугольник задан координатами своих вершин. Найдите периметр четырёхугольника. Вычисление длины отрезка оформите в виде подпрограммы.

3. Выпуклый четырёхугольник задан длинами своих сторон и диагональю. Найдите площадь четырёхугольника как сумму площадей двух треугольников. Вычисление площади треугольника оформите в виде подпрограммы.

4. Выпуклый пятиугольник задан длинами своих сторон и двумя диагоналями, проведёнными из одной вершины. Найдите площадь пяти-

угольника как сумму площадей трёх треугольников. Вычисление площади треугольника оформите в виде подпрограммы.

**Задание 4.** Даны два натуральных числа:  $N$  и  $M$  ( $N < M$ ). Выдайте на печать все числа, расположенные между  $N$  и  $M$  и удовлетворяющие нижеследующим условиям (предусмотрите случай, когда таких чисел нет). Проверку одного числа оформите в виде подпрограммы.

В числе — нечётное количество цифр, и сумма цифр кратна 5.

#### *Этапы выполнения задания*

I. Определение исходных данных: переменная  $N$  и  $M$ .

II. Определение результатов: числа из промежутка от  $N$  до  $M$ , удовлетворяющие условию задачи.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. В цикле от  $N$  до  $M$  проверяем текущее число. Для проверки опишем функцию, которая будет проверять число. Результатом функции будет *true* или *false*.

3. Вывод результата осуществляется в цикле.

IV. Описание переменных:

переменные  $N$  и  $M$  имеют тип *integer* (или *longint*). Тип переменной  $i$  — счётчика цикла, совпадает с типом  $M$ ,  $N$ .

V. Программа:

```
Var M, N, i: longint;
Function prov(x:longint):boolean;
Var z, k, s: longint;
Begin
  s:=0;
  k:=0;
  while x>=1 do
  begin
    s:=s+x mod 10;
    k:=k+1;
    x:=x div 10;
  end;
  if (k mod 2=1) and (s mod 5=0) then
    prov:=true
  else
    prov:=false;
End;

Begin
  Write('введите N=');
  Readln(N);
  Write('введите M=');
  Readln(M);
  For i:=N to M do
    If prov(i)=true then
      writeln(i);
  readln;
End.
```

## VI. Тестирование.

1. Запустите программу и введите значения:

$N=100$

$M=120$

Проверьте. Результат должен быть следующим:

104

109

113

118

2. Заполните таблицу.

	$N$	$M$	Результат
1	1	9	
2	950	1100	
3	10000	10101	

3. Добавьте в таблицу свои значения  $x$ . Для любых ли введённых значений программа выдаёт результат?

4. Измените программу таким образом, чтобы для случая, когда в промежутке нет чисел, удовлетворяющих условию задачи, выдавалось соответствующее сообщение.

5. Измените программу так, чтобы числа с чётным количеством цифр не проверялись. (Подсказка: проверьте границы промежутка.)

### *Индивидуальные задания*

1. Число является палиндромом (перевёртышем).

2. Сумма квадратов его цифр равна числу  $x$  ( $x$  вводится).

3. Сумма кубов его цифр равна самому числу.

4. Сумма цифр числа, стоящих на нечётных местах, равна сумме цифр числа, стоящих на чётных местах.

**Задание 5.** Напишите программу для решения задачи, оформив подходящие подпрограммы.

1. Треугольник задан длинами своих сторон. Найдите длины его медиан.

2. Треугольник задан длинами своих сторон. Найдите длины его биссектрис.

3. Треугольник задан длинами своих сторон. Найдите длины его высот.

4. Даны координаты вершин треугольника и координаты некоторой точки внутри него. Найдите расстояние до ближайшей стороны треугольника. (Это расстояние является высотой одного из полученных треугольников.)

## **РЕШЕНИЕ ГЕОМЕТРИЧЕСКИХ ЗАДАЧ**

### **Вопросы для повторения**

1. Какие простейшие геометрические фигуры можно задавать на плоскости?

2. Как задаётся точка?

3. Как задаётся отрезок?

4. Как задаётся прямая?

5. Какие формы записи уравнения прямой вам известны?

6. Как записать общее уравнение прямой?
7. Как записать уравнение прямой с угловым коэффициентом?
8. Как записать уравнение прямой, проходящей через две точки?
9. Как определить положение точки относительно прямой?
10. Как определить положение двух точек относительно прямой?
11. Как определить взаимное положение двух отрезков?
12. Как определить точку пересечения двух отрезков?
13. Как определить взаимное положение двух прямых?
14. Как вычисляется расстояние между двумя точками?
15. Как вычисляется расстояние от точки до прямой?
16. Как вычисляется расстояние от точки до отрезка?

### Задания для практических работ

**Задание 1.** Напишите программу для решения задачи.

Уравнение прямой с угловым коэффициентом задано в виде  $y = kx + l$ . Получите его в виде  $y - y_0 = k(x - x_0)$ . Абсцисса точки  $x_0$  вводится.

#### Этапы выполнения задания

I. Определение исходных данных: переменные  $k$ ,  $l$ ,  $x_0$ .

II. Определение результатов: значение переменных  $k$ ,  $y_0$ .

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для решения задачи необходимо преобразовать уравнение от одного вида к другому. Коэффициент  $k$  не изменяет своё значение. Для нахождения значения  $y_0$  нужно подставить значение  $x_0$  в исходное уравнение:  $y_0 = kx_0 + l$ .

3. Вывод результата.

IV. Описание переменных:

все переменные имеют тип *real*.

V. Программа:

```
Var k,l,x0,y0: real;
```

```
Begin
```

```
  Write('введите коэф. уравнения k и l:');
```

```
  Readln(k,l);
```

```
  Writeln('исходное уравнение:');
```

```
  Writeln('y=',k:5:2,'x+',l:5:2);
```

```
  Write('введите начальную точку');
```

```
  Readln(x0);
```

```
  y0:=k*x0+l;
```

```
  writeln('преобразованное уравнение:');
```

```
  Writeln('y-',y0:5:2,'=',k:5:2,'(x-',x0:5:2,')');
```

```
  readln;
```

```
End.
```

VI. Тестирование.

1. Запустите программу и введите значения:

```
k=1  l=2
```

```
x0=3
```

Проверьте. Результат должен быть следующим:  
исходное уравнение:  $y = 1.00x + 2.00$

преобразованное уравнение:

$$y - 5.00 = 1.00(x - 3.00)$$

2. Заполните таблицу:

	$k$	$l$	$x_0$	Результат
1	1.5	0.5	5	
2	-3	5	1	
3	3	-5	1.5	

3. Как выглядит запись уравнения, полученного в третьем тесте? Измените программу так, чтобы коэффициенты печатались в скобках.

4. Измените программу так, чтобы в уравнении перед коэффициентом печатался знак '+' или '-', в зависимости от значения коэффициента.

5. Решите обратную задачу: уравнение прямой задано в виде:  $y - y_0 = k(x - x_0)$ . Получите его в виде  $y = kx + l$ .

### Индивидуальные задания

1. Уравнение прямой задано в виде  $y = kx + l$ . Получите его в виде  $Ax + By + C = 0$ .

2. Уравнение прямой задано в виде  $\frac{(x - x_1)}{(x_2 - x_1)} = \frac{(y - y_1)}{(y_2 - y_1)}$ . Получите его в виде  $y = kx + l$ .

3. Уравнение прямой задано в виде  $y = kx + b$ . Получите его в виде  $\frac{(x - x_1)}{(x_2 - x_1)} = \frac{(y - y_1)}{(y_2 - y_1)}$ . Абсциссы

точек  $x_1$  и  $x_2$  вводятся.

4. Уравнение прямой задано в виде  $Ax + By + C = 0$ . Получите его в виде  $\frac{(x - x_1)}{(x_2 - x_1)} = \frac{(y - y_1)}{(y_2 - y_1)}$ . Абсциссы точек  $x_1$  и  $x_2$  вводятся.

**Задание 2.** Напишите программу для решения задачи.

1. Определите коэффициенты уравнения прямой, проходящей через точку с координатами  $(x_0, y_0)$ , и параллельной прямой  $y = B$ .

2. Определите коэффициенты уравнения прямой, проходящей через точку с координатами  $(x_0, y_0)$ , и параллельной прямой  $y = kx + b$ .

3. Определите коэффициенты уравнения прямой, проходящей через точку с координатами  $(x_0, y_0)$ , и параллельной прямой  $Ax + By + C = 0$ .

4. Определите коэффициенты уравнения прямой, проходящей через точку с координатами  $(x_0, y_0)$ , и параллельной прямой, определяемой точками  $(x_1, y_1)$ ,  $(x_2, y_2)$ .

**Задание 3.** Напишите программу для решения задачи из задания 2 для перпендикулярной (а не параллельной) прямой.

Уравнение прямой перпендикулярной прямой  $Ax + By + C = 0$  может быть записано в виде  $Bx - Ay + D = 0$ , где коэффициент  $D$  определяется координатами точки, через которую провели прямую, перпендикулярную прямой  $Ax + By + C = 0$ .

**Задание 4.** Напишите программу для решения задачи.

Определите взаимное расположение трёх точек  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  и прямой  $Ax + By + C = 0$ .

#### Этапы выполнения задания

I. Определение исходных данных: переменные  $A, B, C, x_1, y_1, x_2, y_2, x_3, y_3$ .

II. Определение результатов: сообщение о взаимном расположении точек и прямой.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для трёх точек и прямой возможны следующие случаи взаимного расположения:

а) три точки лежат на прямой;

б) две точки лежат на прямой, одна не принадлежит прямой;

в) одна точка лежит на прямой, две — в одной полуплоскости относительно прямой;

г) одна точка лежит на прямой, две — по разные стороны относительно прямой;

д) три точки лежат в одной полуплоскости относительно прямой;

е) две точки лежат в одной полуплоскости относительно прямой, одна — в другой полуплоскости.

Для определения принадлежности точки прямой нужно координаты точки подставить в уравнение прямой. Если значение выражения  $Ax + By + C$  равно нулю, то точка лежит на прямой, если значение отрицательно, то точка ле-

жит в одной полуплоскости относительно прямой («отрицательной»), а если положительно — то в другой («положительной»). Посчитаем, сколько точек лежит на прямой, в «отрицательной» и «положительной» полуплоскостях. В зависимости от полученных значений определим один из возможных случаев расположения.

Для проверки принадлежности точки прямой определим процедуру.

3. Вывод результата.

IV. Описание переменных:

все переменные, хранящие значения коэффициентов и координат, имеют тип *real*, счётчики количества  $k_1$  (на прямой),  $k_2$  (в «отрицательной» полуплоскости),  $k_3$  (в «положительной» полуплоскости) — тип *integer*.

V. Программа:

```
Var A,B,C,x1,y1,x2,y2,x3,y3: real;
```

```
k1,k2,k3,z1,z2,z3:integer;
```

```
Procedure t_n_p(x0,y0,A1,B1,C1:real;  
                var n,o,p:integer);
```

```
Var z:real;
```

```
Begin
```

```
n:=0; o:=0; p:=0;
```

```
Z:=A1*x0+B1*y0+C1;
```

```
If z=0 then n:=1 {точка на прямой}
```

```
Else
```

```
If z<0 then o:=1 {точка в «отрицатель-  
                    ной» полуплоскости}
```

```
Else p:=1; {точка в «положительной»  
            полуплоскости}
```

```

End;
Begin
  Write('введите коэф. уравнения A, B, C:');
  Readln(A,B,C);
  Write('введите координаты точек');
  Readln(x1,y1,x2,y2,x3,y3);
  k1:=0; k2:=0; k3:=0;
  t_n_p(x1,y1,A,B,C,z1,z2,z3);
  k1:=k1+z1; k2:=k2+z2; k3:=k3+z3;
  t_n_p(x2,y2,A,B,C,z1,z2,z3);
  k1:=k1+z1; k2:=k2+z2; k3:=k3+z3;
  t_n_p(x3,y3,A,B,C,z1,z2,z3);
  k1:=k1+z1; k2:=k2+z2; k3:=k3+z3;
  case k1 of
    3: writeln('3 точки на прямой');
    2: writeln('2 точки на прямой, 1 не на
              прямой');
    1: if (k2=1) or (k3=1) then writeln('1
              точка на прямой, 2 точки по
              разные стороны от прямой')
        else writeln('1 точка на прямой, 2 точ-
              ки с одной стороны от прямой');
    0: if (k2=3) or (k3=3) then writeln('3 точ-
              ки с одной стороны от прямой')
        else writeln('2 точки с одной стороны
              от прямой, 1 точка — с другой');
  end;
  readln;
End.

```

## VI. Тестирование.

1. Запустите программу и введите значения, проверьте результат.

	Коэф. прямой	1-я точка	2-я точка	3-я точка	Резуль- тат
1	1 1 0	0 0	1 -1	-1 1	3-й точ- ки на прямой
2	1 1 0	1 2	3 4	5 6	3-й точ- ки с одной стороны

2. Заполните таблицу:

	Коэф.	1-я точка	2-я точка	3-я точка	Резуль- тат
1	1 1 0	0 0	1 -1	-1 -5	
2	1 1 0	0 0	3 4	-5 -6	
3	1 1 0	1 2	3 4	-5 -6	

3. Все ли случаи расположения точек и прямой проверены? Если нет, добавьте свои тесты для нерассмотренных случаев.

4. Можно ли записать решение задачи без использования оператора case? Как это сделать? Какая запись оптимальнее и почему?

5. Как изменится решение этой задачи, если прямая будет задана уравнением другого вида?

6. Сколько добавиться новых возможностей расположения точек и прямой, если добавить ещё одну точку? ещё одну прямую?

### *Индивидуальные задания*

1. Определите взаимное расположение точек с координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$  и прямой  $y = kx + b$ .

2. Определите взаимное расположение точки с координатами  $(x_0, y_0)$  и параллельных прямых  $y = kx + b_1$  и  $y = kx + b_2$ .

3. Определите взаимное расположение точки с координатами  $(x_0, y_0)$  и параллельных прямых  $Ax + By + C_1 = 0$  и  $Ax + By + C_2 = 0$ .

4. Определите взаимное расположение точек с координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$  и параллельных прямых  $Ax + By + C_1 = 0$  и  $Ax + By + C_2 = 0$ .

**Задание 5.** Напишите программу для решения задачи.

1. На плоскости заданы три точки. Проверьте, лежат ли они на одной прямой.

2. На плоскости заданы четыре точки. Через первые две проведена прямая. Определите положение двух других относительно проведённой прямой.

3. На плоскости заданы четыре точки. Можно ли построить две параллельные прямые так, чтобы каждая из прямых проходила через две

заданные точки? Если «да», то введите дополнительно координаты пятой точки и определите положение этой точки относительно полученных прямых.

4. На плоскости заданы четыре точки. Точки соединили отрезками в том порядке, в котором они вводились. Получился ли выпуклый четырёхугольник? Если «нет», то определите порядок соединения точек, чтобы получить выпуклый четырёхугольник.

**Задание 6.** Напишите программу для решения задачи.

На плоскости заданы прямая в виде  $Ax + By + C = 0$  и отрезок, не пересекающий эту прямую. Найдите расстояние от отрезка до прямой и длину проекции отрезка на прямую.

### *Этапы выполнения задания*

I. Определение исходных данных:  $x_1, y_1, x_2, y_2$  — координаты концов отрезка,  $A, B, C$  — коэффициенты уравнения прямой.

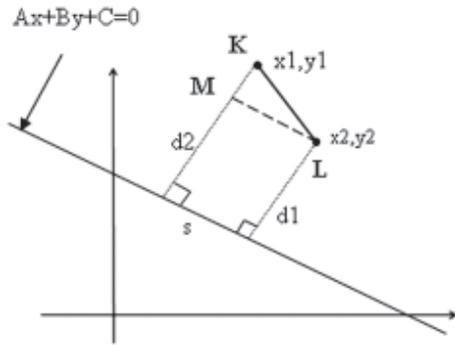
II. Определение результатов:  $d$  — расстояние от отрезка до прямой,  $s$  — длина проекции.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Расстояние от отрезка до прямой вычисляется как минимальное из расстояний от концов отрезка до прямой. Расстояние от точки до прямой

вычисляется по формуле:  $d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$ ,



$(x_0, y_0$  — координаты точки). То есть необходимо найти минимальное значение из  $d_1$  и  $d_2$  (см. рисунок). Для нахождения длины проекции отрезка на прямую рассмотрим прямоугольный треугольник KLM. Длина отрезка KL равна  $g = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ , длина отрезка KM —  $k = |d_2 - d_1|$ , тогда длину отрезка LM, равную  $s$ , можно вычислить по теореме Пифагора:  $s = \sqrt{g^2 - k^2}$ . Преобразуем данную формулу.

$$d_1 = \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}}; \quad d_2 = \frac{|Ax_2 + By_2 + C|}{\sqrt{A^2 + B^2}}, \text{ тогда } =$$

$$k = |d_2 - d_1| = \left| \frac{|Ax_2 + By_2 + C|}{\sqrt{A^2 + B^2}} - \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}} \right| =$$

$$= \left| \frac{Ax_2 + By_2 + C - (Ax_1 + By_1 + C)}{\sqrt{A^2 + B^2}} \right| =$$

$$= \left| \frac{A(x_2 - x_1) + B(y_2 - y_1)}{\sqrt{A^2 + B^2}} \right|. \text{ Возведём } k \text{ в квадрат.}$$

$$k^2 = \frac{(A(x_2 - x_1) + B(y_2 - y_1))^2}{A^2 + B^2}. \text{ Подставим получен-$$

ные значения в итоговую формулу.

$$s = \sqrt{g^2 - k^2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 - \frac{(A(x_1 - x_2) + B(y_1 - y_2))^2}{A^2 + B^2}} =$$

$$= \sqrt{\frac{((x_1 - x_2)^2 + (y_1 - y_2)^2)(A^2 + B^2) - (A^2(x_1 - x_2)^2 - 2AB(x_1 - x_2)(y_1 - y_2) + B^2(y_1 - y_2)^2)}{A^2 + B^2}} =$$

$$= \sqrt{\frac{(A^2(x_1 - x_2)^2 + A^2(y_1 - y_2)^2 + B^2(x_1 - x_2)^2 + B^2(y_1 - y_2)^2) - (A^2(x_1 - x_2)^2 - 2AB(x_1 - x_2)(y_1 - y_2) + B^2(y_1 - y_2)^2)}{A^2 + B^2}} =$$

$$= \sqrt{\frac{A^2(y_1 - y_2)^2 + B^2(x_1 - x_2)^2 + 2AB(x_1 - x_2)(y_1 - y_2)}{A^2 + B^2}} = \sqrt{\frac{(A(y_1 - y_2) + B(x_1 - x_2))^2}{A^2 + B^2}} =$$

$$= \frac{|A(y_1 - y_2) + B(x_1 - x_2)|}{\sqrt{A^2 + B^2}}.$$

$$\text{В итоге получаем } s = \frac{|A(y_1 - y_2) + B(x_1 - x_2)|}{\sqrt{A^2 + B^2}}.$$

3. Вывод результата.

III. Описание переменных:

все переменные в программе имеют тип *real*.

IV. Программа:

Var A,B,C,x1,y1,x2,y2,d1,d2,d,s: real;

Begin

Write('введите координаты концов отрезка');

Readln(x1,y1,x2,y2);

Write('введите коэф. уравнения прямой');

Readln(A,B,C);

d1:= ABS(A\*x1+B\*y1+C)/SQRT(A\*A+B\*B);

d2:= ABS(A\*x2+B\*y2+C)/SQRT(A\*A+B\*B);

if d1<d2 then d:=d1 else d:=d2;

s:= ABS(A\*(y1-y2)+B\*(x1-x2))/SQRT(A\*A+B\*B);

```
writeln('расстояние до прямой равно d=',d:5:2);
writeln('длина проекции отрезка равна s=',s:5:2);
End.
```

V. Тестирование.

1. Запустите программу и введите значения, проверьте результат.

	Отрезок	Коэффициенты уравнения прямой	$d$	$s$
1	2 2 5 5	0 1 -1	1.00	3.00
2	1 2 3 4	5 6 7	3.07	2.82

2. Заполните таблицу:

	Отрезок	Коэффициенты уравнения прямой	$d$	$s$
1	-1 3 4 -8	1 1 1		
2	-1 3 4 -8	-3 6 18		
3	4 7 -2 -5	1 1 1		

3. Добавьте свои значения в таблицу.

### Индивидуальные задания

1. На плоскости заданы треугольник и точка. Найдите расстояние от точки до треугольника.

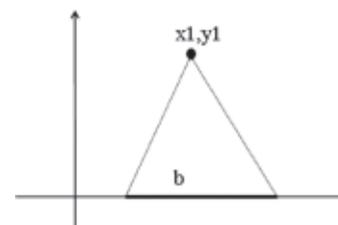
2. На плоскости заданы четырёхугольник и точка. Найдите расстояние от точки до четырёхугольника.

3. На плоскости заданы треугольник и отрезок. Найдите расстояние между ними.

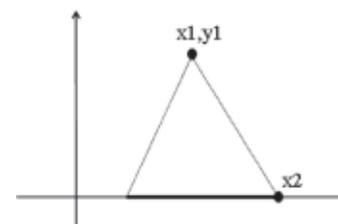
4. На плоскости заданы два треугольника. Найдите расстояние между ними.

*Задание 7.* Напишите программу для вычисления периметра фигуры, описанной ниже (см. рисунки):

1) равнобедренного треугольника, основание  $b$  которого лежит на оси  $OX$ , а вершина задаётся точкой  $(x_1, y_1)$ ;

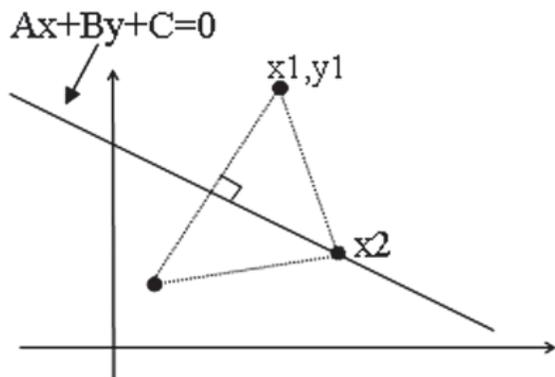


2) равнобедренного треугольника, основание которого лежит на оси  $OX$  и начинается в точке с абсциссой  $x_2$ , а вершина задаётся точкой  $(x_1, y_1)$ ;

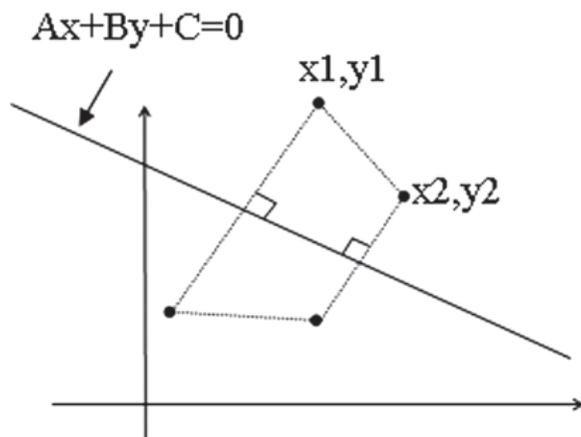


3) треугольника, который получается следующим образом: на плоскости заданы прямая  $Ax + By + C = 0$  и точка  $(x_1, y_1)$ , не принадлежа-

щая данной прямой (первая вершина треугольника). На прямой отмечают точку с абсциссой  $x_2$  (вторая вершина). Третья вершина — точка, симметричная точке  $(x_1, y_1)$  относительно заданной прямой;



4) четырёхугольника, который получается следующим образом: на плоскости заданы прямая  $Ax + By + C = 0$  и две точки  $(x_1, y_1)$  и  $(x_2, y_2)$ ,



не принадлежащие данной прямой и расположенные в одной полуплоскости относительно её (первая и вторая вершины четырёхугольника). Третья и четвёртая вершины — точки, симметричные точкам  $(x_1, y_1)$  и  $(x_2, y_2)$  относительно заданной прямой.

**Задание 8.** Напишите программу для решения задачи. При решении задачи необходимо принять во внимание, что если два отрезка и более пересекаются в одной точке, то это одна общая точка.

На плоскости заданы четырёхугольник и отрезок. Сколько у них точек пересечения, если известно, что отрезок не может пересекать четырёхугольник в вершине?

**Этапы выполнения задания**

I. Определение исходных данных:  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$  — координаты вершин четырёхугольника,  $x_1, y_1, x_2, y_2$  — координаты концов отрезка.

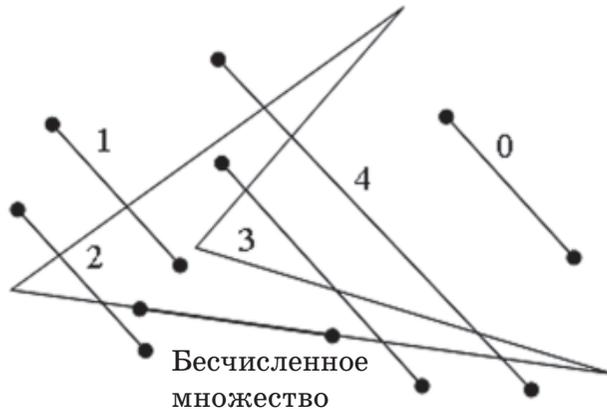
II. Определение результатов:  $k$  — количество точек пересечения.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Количество точек пересечения отрезка и четырёхугольника получается как сумма количества точек пересечения отрезка с каждой из сторон четырёхугольника. Сторона четырёхугольника является, в свою очередь, также отрезком, поэтому задача сводится к нахождению

точки пересечения двух отрезков. Два отрезка могут иметь одну точку пересечения, не иметь точек пересечения или иметь бесчисленное количество точек пересечения. Количество точек пересечения четырёхугольника и отрезка может быть следующим: 0, 1, 2, 3, 4 или бесчисленное множество (в случае, если отрезок полностью или частично лежит на стороне четырёхугольника). Данные случаи представлены на рисунке. Если один отрезок полностью или частично лежит на другом, то количество точек пересечения равно  $-1$ .



### 3. Вывод результата.

#### II. Описание переменных:

все переменные, хранящие значения координат, имеют тип *real*, счётчик количества  $k$  – тип *integer*.

#### III. Программа:

```
Var A,B,C,x1,y1,x2,y2,x3,y3,x4,y4,xx1,yy1,xx2,yy2:
real;
```

```
k,z1,z2,z3,z4:integer;
{проверка, пересекаются ли два отрезка}
Function peres(a1, b1, a2, b2, a3, b3, a4,
               b4:real):integer;
var zn1,zn2,zn3,zn4,k1,k2,k3,k4,lx,ly,rx,ry:real;
    p:integer;
Function min(a,b:real):real;
Begin
  If a<b then min:=a
  Else min:=b;
End;

Function max(a,b:real):real;
Begin
  If a>b then max:=a
  Else max:=b;
End;

Begin
zn1:=(a3-a1)*(b2-b1)-(b3-b1)*(a2-a1);
zn2:=(a4-a1)*(b2-b1)-(b4-b1)*(a2-a1);
if (zn1=0) and (zn2=0) then
  if (a1=a2)and(a1=a3)and(a1=a4) then
  begin
    {отрезки лежат на одной прямой, парал-
                                     лельной оси Y}

    k1:=min(b1,b2);
    k2:=max(b1,b2);
    k3:=min(b3,b4);
```

```

k4:=max(b3,b4);
ly:=max(k1,k3);
ry:=min(k2,k4);
if ly>ry then
  p:=0
else
  if ly=ry then
    p:=1
  else
    p:=-1;
end
else
begin
{отрезки лежат на одной прямой, парал-
ельной оси X}

k1:=min(a1,a2);
k2:=max(a1,a2);
k3:=min(a3,a4);
k4:=max(a3,a4);
lx:=max(k1,k3);
rx:=min(k2,k4);
if lx>rx then
  p:=0
else
  if lx=rx then
    p:=1
  else
    p:=-1;
end

```

```

else
begin
{отрезки лежат на пересекающихся прямых}
  if zn1*zn2>0 then
    p:=0
  else
begin
  zn3:=(a1-a3)*(b4-b3)-(b1-b3)*(a4-a3);
  zn4:=(a2-a3)*(b4-b3)-(b2-b3)*(a4-a3);
  if zn3*zn4>0 then
    p:=0
  else
    p:=1;
end;
end;
peres:=p;
End;

Begin
Write('введите координаты концов отрезка');
Readln(xx1,yy1,xx2,yy2);
Write('введите коорд. вершин четырёх-
угольника');
Readln(x1,y1,x2,y2,x3,y3,x4,y4);
Z1:=peres(xx1,yy1,xx2,yy2,x1,y1,x2,y2);
Z2:=peres(xx1,yy1,xx2,yy2,x2,y2,x3,y3);
Z3:=peres(xx1,yy1,xx2,yy2,x3,y3,x4,y4);

```

```

Z4:=peres(xx1,yy1,xx2,yy2,x4,y4,x1,y1);
If (z1=-1)or(z2=-1)or(z3=-1)or(z4=-1)
then
  Writeln('бесчисленное количество то-
          чек пересечения')
Else
Begin
  K:=z1+z2+z3+z4;
  Writeln('количество точек пересечения
          равно ', k);
End;
Readln;
End.

```

#### IV. Тестирование.

1. Запустите программу и введите значения, проверьте результат.

	Отрезок	Четырёхугольник	Результат
1	1 3 7 4	2 2 4 6 6 2 4 4	4
2	4 4 5 5	2 2 4 6 6 2 4 4	1

2. Заполните таблицу:

	Отрезок	Четырёхугольник	Результат
1	-5 -2 -3 -1	2 2 4 6 6 2 4 4	
2	2 5 6 5	2 2 4 6 6 2 4 4	
3	1 1 3 3	2 2 4 6 6 2 4 4	

3. Все ли случаи расположения четырёхугольника и отрезка проверены? Если нет, добавьте свои тесты для нерассмотренных случаев.

4. Выведите координаты точек пересечения, если их конечное количество.

5. Рассмотрите возможность пересечения отрезка с вершиной многоугольника.

#### *Индивидуальные задания*

1. На плоскости заданы три отрезка. Проверьте, пересекаются ли все они в одной точке.

2. На плоскости заданы три отрезка. Посчитайте количество точек пересечений этих отрезков.

3. На плоскости заданы треугольник и отрезок. Сколько у них точек пересечения?

4. На плоскости заданы два треугольника. Сколько у них общих точек?

**Задание 9.** На плоскости заданы две фигуры. Определите их взаимное расположение. Если фигуры имеют общие точки, то найдите координаты точек пересечения сторон фигур.

1. Два прямоугольника со сторонами, параллельными осям координат.

2. Два треугольника.

3. Треугольник и четырёхугольник.

4. Два четырёхугольника.

## **РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ МАССИВОВ**

#### **Вопросы для повторения**

1. Как описываются одномерные массивы в языке Pascal?

2. Какого типа могут быть элементы массива?
3. Какого типа могут быть индексы элементов массива?
4. Что такое сортировка?
5. Какие виды сортировки вам известны?
6. Опишите алгоритм сортировки выбором?
7. Опишите алгоритм сортировки обменом?
8. Чему равно количество операций сравнения для алгоритмов сортировки обменом и выбором?
9. Чему равно количество перестановок элементов для алгоритмов сортировки обменом и выбором?
10. Для каких задач применяется алгоритм двоичного поиска?
11. Опишите алгоритм двоичного поиска.
12. Опишите алгоритмы сортировки простыми и бинарными вставками.
13. Что определяет размерность массива?
14. Как описывается двумерный массив в программе?
15. В чём отличие одномерных массивов от двумерных?
16. Что такое квадратная матрица?
17. Что такое диагональная матрица?
18. Какими способами может быть заполнен массив? Приведите примеры.
19. Как определить минимальный объём памяти, отводимой под массив?
20. Какие действия выполняют обычно над элементами массива?
- 21\*. Может ли массив быть элементом массива?

## Задания для практических работ

**Задание 1.** Элементы линейного массива хранятся в текстовом файле `input.txt`. Преобразуйте его элементы по заданному правилу и результат выведите в файл `output.txt`:

Положительные элементы массива увеличьте на 5, а отрицательные замените их модулями.

### Этапы выполнения задания

I. Определение исходных данных: переменная  $n$  (количество элементов в массиве),  $a$  — линейный массив.

II. Определение результатов: преобразованный массив  $a$ .

III. Алгоритм решения задачи.

1. Ввод исходных данных. Из файла считываем значение переменной  $n$  — количество элементов массива. Массив поэлементно читается из файла.

2. В цикле проверяем текущий элемент, если он положительный, прибавляем к нему 5, если отрицательный, то заменяем его модулем.

3. Вывод результата.

IV. Описание переменных:

переменная  $n$  имеет тип *integer*, элементы массива имеют тип *integer* (или *real*). Для работы также необходима целая переменная  $i$  — счётчик цикла.

V. Программа:

```
Var n, i: integer;
    a: array[1..10] of integer;
Begin
```

```

Assign(Input, 'input.txt');
Reset(input);
Readln(input, n);
For i:=1 to n do
  Read(input, a[i]);
Close(input);
For i:=1 to n do
  begin
    if a[i]>0 then
      a[i]:=a[i]+5;
    if a[i]<0 then
      a[i]:=abs(a[i]);
  end;
Assign(output, 'output.txt');
Rewrite(output);
For i:=1 to n do
  write(output, a[i]:6);
close(output);
End.

```

#### VI. Тестирование.

1. Создайте текстовый файл input.txt следующего содержания:

```

5
3 -2 0 -1 5

```

Запустите программу.

В результате работы программы должен быть создан текстовый файл output.txt следующего содержания:

```

8 2 0 1 10

```

2. Заполните таблицу:

	input.txt	output.txt
1	3 -2 -3 -5	
2	5 1 2 3 4 5	
3	10 1 -3 -2 0 4 0 2 -4 0 2	

3. Добавьте в таблицу свои значения.

4. Как другим способом можно заменить отрицательный элемент массива его модулем?

5. Можно ли заменить команды

```

if a[i]>0 then
  a[i]:=a[i]+5;
if a[i]<0 then
  a[i]:=abs(a[i]);

```

командой?

```

if a[i]<0 then
  a[i]:=abs(a[i]);
else
  a[i]:=a[i]+5;

```

В каких случаях программа будет давать неверный результат? Сформулируйте условие задачи, решением которой служит полученная программа.

#### *Индивидуальные задания*

1. Положительные элементы замените на 1, а отрицательные — на -1.

2. Чётные элементы замените на 1, а нечётные на  $-1$ .

3. Элементы, кратные трём, оставьте без изменения, а остальные замените остатками от деления на 3.

4. Чётные кратные трём элементы замените на 0, чётные не кратные трём — на 1, остальные — на  $-1$ .

**Задание 2.** В линейном массиве определите:

- 1) число соседств двух разных знаков;
- 2) число соседств двух нечётных чисел;
- 3) число соседств трёх упорядоченных по неубыванию чисел;
- 4) число соседств трёх чисел одинаковой чётности.

**Задание 3.** Дан линейный массив.

1. Отсортируйте его методом обмена в порядке возрастания его модулей.

2. Отсортируйте его методом выбора при помощи поиска *min* в порядке убывания.

3. Переставьте все нулевые элементы в конец массива (вспомогательный массив не использовать).

4. Переставьте все нулевые элементы так, чтобы все чётные стояли в начале массива, а нечётные — в конце (вспомогательный массив не использовать).

**Задание 4.** Задан упорядоченный массив. Воспользуйтесь алгоритмом двоичного поиска.

Удалите из массива элементы, расположенные между числами  $x$  и  $2x$ , вместе с самими числами. (Предполагается, что такие элементы есть в массиве.) Массив отсортирован в порядке неубывания.

#### *Этапы выполнения задания*

I. Определение исходных данных: переменная  $a$  — исходный массив.

II. Определение результатов:  $a$  — преобразованный массив.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Поиск места, которое занимают в массиве числа  $x$  и  $2x$ . Поскольку поиск необходимо производить дважды, можно оформить функцию *poisk*, которая будет осуществлять поиск.

3. Определение места крайнего левого элемента, равного  $x$ , и места крайнего правого элемента, равного  $2x$ .

4. Для удаления необходимых элементов нужно сдвинуть все элементы, большие  $2x$ , влево на позицию, номер которой определяется как  $m2 - m1$ .

5. Вывод результата.

IV. Описание переменных:

все переменные и элементы массива имеют тип *integer*.

V. Программа:

```
Var i,n,m1,m2,x,k: integer;  
a: array [1..20] of integer;
```

```

Function poisk(y:integer):integer;
Var ii,m,L,R:integer;
    P:boolean;
Begin
  L:=1; R:=N; P:=false;
  While (L<=R) and (P=false) do
  begin
    m:=(L+R) div 2;
    if a[m]=y then
      p:=true
    else
      if a[m]<y then
        L:=m+1
      else
        R:=m-1;
    end;
    poisk:=m;
  End;

Begin
  Writeln('введите кол. элем. в массиве');
  Readln(n);
  Writeln('введите элем. в порядке возраст. ');
  For i:=1 to n do
    Read(a[i]);
  Writeln('введите число x');
  Readln(x);
  m1:= poisk(x); {позиция x}
  m2:=poisk(2*x); {позиция 2x}

```

```

{поиск крайнего левого элемента, равно-
                                     го x}
if m1<>1 then
  while a[m1-1]=x do
    m1:=m1-1;
{поиск крайнего правого элемента, рав-
                                     ного 2x}
if m2<>n then
  while a[m2+1]=2*x do
    m2:=m2+1;
{сдвиг элементов на k=m2-m1 позиций}
k:=m2-m1;
for i:=m1 to n-k+1 do
  a[i]:=a[i+k+1];
for i:=1 to n-k-1 do
  write(a[i]:5);
writeln;
readln;
End.

```

VI. Тестирование.

1. Запустите программу и введите значения:

```

n=5
1 2 3 4 5
x=2

```

Проверьте. Результат должен быть следующим:

```

1 5

```

Действительно: элемент, равный 2, стоит на втором месте, равный  $2*2=4$ , стоит на четвертом

месте. Значит, необходимо удалить все элементы, стоящие на местах 2, 3, 4.

2. Заполните таблицу:

	<b>n</b>	<b>Массив</b>	<b>x</b>	<b>Результат</b>
1	7	1 2 3 4 5 6 7	3	
2	10	1 2 3 4 5 6 7 8 9 10	4	
3	5	1 2 3 4 5	1	
4	10	1 2 2 2 2 3 4 4 4 5	2	

3. Добавьте в таблицу свои значения. Рассмотрите случаи: а) элемент, равный  $2x$ , стоит на последнем месте; б) элемент  $x$  стоит на первом месте, а  $2x$  — на последнем.

4. В каком случае из массива выбросятся все элементы? Предусмотрите вывод сообщения: «в массиве не осталось элементов».

5. Как изменится программа, если неизвестно, есть ли в массиве элементы  $x$  и  $2x$ ? Внесите соответствующие изменения.

6. Внесите в программу изменения так, чтобы ввод данных осуществлялся из текстового файла input.txt, а вывод результата — в текстовый файл output.txt.

### *Индивидуальные задания*

1. Получите количество элементов, меньших  $x$ . (Предполагается, что такой элемент существует и он единственный.) Массив упорядочен в порядке неубывания.

2. Получите количество элементов, расположенных между  $x$  и  $y$ . (Предполагается, что такие элементы существуют и они единственные.) Массив упорядочен в порядке невозрастания.

3. Получите количество элементов, меньших  $x$ . (Элементов, равных  $x$ , может быть несколько.) Массив упорядочен в порядке невозрастания.

4. Вставьте элемент, равный  $x$ , не нарушая упорядоченности. (Такой элемент может быть.) Массив упорядочен в порядке невозрастания.

**Задание 5.** Двумерный массив  $A(m, n)$  хранится в текстовом файле input.txt. Посчитайте сумму и выведите её в текстовый файл output.txt.

Посчитайте сумму тех элементов, у которых модуль разности номера строки и номера столбца равен числу  $x$ .

### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$  и  $m$  — количество строк и столбцов в массиве,  $a$  — двумерный массив,  $x$  — число.

II. Определение результатов: переменная  $S$  (сумма искоемых элементов).

III. Алгоритм решения задачи.

1. Ввод исходных данных. В первой строке файла хранятся два числа  $m$  и  $n$  — количество строк и столбцов. Затем идут  $m$  строк по  $n$  чисел в каждой — значения элементов массива. Затем число  $x$ .

2. Определение начального значения для суммы ( $S:=0$ ).

3. Проверяем текущий элемент, и если он удовлетворяет условию, добавляем его к сумме.

4. Вывод результата.

IV. Описание переменных:

переменные  $n$ ,  $m$  и  $x$  имеют тип *integer*, элементы массива имеют тип *integer* (или *real*). Тип переменной  $S$  совпадает с типом элементов массива. Для работы также необходимы целые переменные  $i$  и  $j$  — счётчики цикла.

V. Программа:

```
Var n, m, i, j, x, S: integer;  
    a: array[1..10, 1..10] of integer;
```

Begin

```
Assign(Input, 'input.txt');
```

```
Reset(input);
```

```
Readln(input, m, n);
```

```
For i:=1 to m do
```

```
    For j:=1 to n do
```

```
        Read(input, a[i,j]);
```

```
Readln(input, x);
```

```
Close(input);
```

```
S:=0;
```

```
For i:=1 to m do
```

```
    For j:=1 to n do
```

```
        If abs(i-j)=x then
```

```
            S:=S+a[i,j];
```

```
Assign(output, 'output.txt');
```

```
Rewrite(output);
```

```
writeln(output, S);
```

```
close(output);
```

End.

VI. Тестирование.

1. Создайте текстовый файл input.txt следующего содержания и запустите программу:

```
3 3  
3 2 5  
-3 7 5  
2 4 5  
1
```

Проверьте файл output.txt. Результат должен быть следующим (элементы, удовлетворяющие условию, подчёркнуты):

```
8
```

2. Заполните таблицу:

	input.txt	output.txt
1	2 2 -2 -3 -5 -4 0	
2	3 3 1 2 3 4 5 6 7 8 9 2	
3	2 5 1 -3 -2 0 4 0 2 -4 0 2 1	

3. Добавьте в таблицу свои значения.

4. Какой результат выдаст программа, если в текстовый файл input.txt (см. пункт 1) записать в последней строке 10 вместо 1? Почему? Всегда ли значение 0 в файле output.txt говорит о том, что в массиве нет подходящих элементов? Измените программу так, чтобы выдавалось бы сообщение 'таких элементов нет' в соответствующих случаях.

5. Каким должно быть число  $x$ , чтобы в массиве всегда были элементы, удовлетворяющие условию задачи? Измените программу так, чтобы проверка проводилась только в том случае, когда число  $x$  задано корректно.

6\*. Как организовать вычисления в программе так, чтобы не нужно было вообще проверять условие?

### *Индивидуальные задания*

Посчитайте сумму:

1) элементов, у которых сумма номера строки и номера столбца равна числу  $x$ . Число  $x$  вводится;

2) элементов, которые расположены на главной и побочной диагоналях;

3) элементов, которые расположены ниже главной диагонали;

4) элементов, которые расположены в заштрихованной области.



**Задание 6.** Дан двумерный массив  $A(m, n)$ . Исходный массив прочитайте из файла input.txt. Результат запишите в файл output.txt.

1. Положительные элементы замените на 1, а отрицательные — на  $-1$ .

2. Чётные элементы замените на 1, а нечётные — на  $-1$ .

3. Элементы, кратные трём, оставьте без изменения, а остальные замените остатками от деления на 3.

4. Чётные, кратные трём, элементы замените на 0, чётные, не кратные трём, — на 1, остальные — на  $-1$ .

**Задание 7.** Напишите программу для решения задачи. Исходный массив прочитайте из файла input.txt. Результат запишите в файл output.txt.

1. Дан двумерный массив  $A(m, n)$ . Получите линейный массив  $B(m)$ , в котором  $B(i)$  — сумма элементов  $i$ -й строки массива  $A(m, n)$ , ( $i$  изменяется от 1 до  $m$ ).

2. Дан двумерный массив  $A(m, n)$ . Получите линейный массив  $B(m)$ , в котором  $B(i)$  —  $\max$  элемент  $i$ -й строки массива  $A(m, n)$ , ( $i$  изменяется от 1 до  $m$ ).

3. Дан двумерный массив  $A(m, n)$ . Получите линейный массив  $B(m)$ , в котором  $B(i)$  — целая часть среднего арифметического  $i$ -й строки массива  $A(m, n)$ , ( $i$  изменяется от 1 до  $m$ ).

4. Дан линейный массив  $A(n)$ . Получите двумерный массив  $B(n, m)$ , в котором столбец с номером  $i$  содержит элементы массива  $A(n)$ , возведённые в степень  $i$  ( $i$  изменяется от 1 до  $m$ ).

**Задание 8.** Дан двумерный массив  $A(m, n)$ .

1. Напечатайте номер столбца, содержащего  $\max$  элемент.

2. Напечатайте номер строки, сумма элементов которой  $\min$ .

3. Напечатайте номера строк, в которых есть элементы, равные  $\max$ .

4. Напечатайте номера столбцов, в которых есть элементы, равные  $\min$ .

**Задание 9.** Заполните прямоугольную таблицу числами по образцу:

1. 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4	2. 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 1 1
3. 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1	4. 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1

**Задание 10.**

1. Заполните квадратную матрицу размером  $n \times n$  числами от 1 до  $n^2$ . Число 1 находится в первом столбце первой строки. Порядок заполнения по строкам: нечётная строка — слева направо, чётная — справа налево.

2. Заполните квадратную матрицу размером  $n \times n$  числами от 1 до  $n^2$ . Число 1 находится в последнем столбце последней строки. Порядок заполнения по столбцам:  $n$ -й столбец — снизу вверх,  $n-1$  столбец — сверху вниз и т. д.

3. Заполните по спирали квадратную матрицу размером  $n \times n$  числами от 1 до  $n^2$ . Число 1 находится в первом столбце первой строки. Порядок заполнения: вправо, вниз, влево, вверх.

4. Заполните по спирали квадратную матрицу размером  $n \times n$  числами от 1 до  $n^2$ . Число 1 находится в «центре» матрицы, позиции  $((n \div 2)+1, (n \div 2)+1)$ . Порядок заполнения: вверх, влево, вниз, вправо.

## ЗАДАЧИ ЦЕЛОЧИСЛЕННОЙ АРИФМЕТИКИ

### Вопросы для повторения

1. Что называют делителем числа?
2. Что называют кратным данного числа?
3. Какие числа называют чётными? нечётными?
4. Какие числа называют простыми? составными?

5. Как проверить, является ли число простым?
6. Как получить простые числа, не превосходящие заданного числа  $n$ ?
7. Как разложить число на простые множители?
8. Какое разложение называется каноническим?
9. Какие числа называют взаимно простыми?
10. Что такое наибольший общий делитель двух чисел?
11. Как получить наибольший общий делитель двух чисел?
12. Что такое наименьшее общее кратное двух чисел?
13. Как получить наименьшее общее кратное двух чисел?

### Задания для практических работ

**Задание 1.** Напишите программу для решения задачи. Дано натуральное число  $n$ .

Посчитайте количество цифр числа и определите, какая цифра стоит в разряде с номером  $i$  (разряды нумеруются с конца, т. е. первый номер имеет разряд единиц).

#### Этапы выполнения задания

- I. Определение исходных данных: переменная  $n$ .
- II. Определение результатов: переменная  $k$  (количество цифр в числе  $n$ ).
- III. Алгоритм решения задачи.
  1. Ввод исходных данных
  2. Определение начального значения счётчика количества цифр ( $k:=0$ ).

3. Количество цифр числа равно количеству десятичных разрядов в числе. Для нахождения всех цифр числа нужно:

- а) разделить число на 10;
- б) найти целую часть от деления и остаток (остаток и будет очередной цифрой) и увеличить счётчик количества цифр;
- в) если целая часть от деления больше 1, то в числе ещё есть цифры и нужно перейти к выполнению пункта а), иначе все цифры обработаны.

4. Для определения цифры, стоящей в разряде  $i$ , нужно после получения очередной цифры сравнить значения  $k$  и  $i$ . Если они совпадают, напечатайте полученную цифру.

5. Вывод результата.

IV. Описание переменных:

все переменные, определённые для решения задачи, имеют тип *longint*.

V. Программа:

```

Var i,k,n,z: longint;
Begin
  Write('введите значение n=');
  Readln(n);
  Write('введите значение i=');
  Readln(i);
  k:=0;
  while n>=1 do
  begin
    z:=n mod 10; {текущая цифра}
    k:=k+1;
  end
end

```

```

if k=i then
  writeln('в разряде ', i, ' стоит цифра ', z);
  n:=n div 10; {уменьшение числа в 10 раз}
end;
if i>k then
  writeln('в числе', k, 'цифр, в разряде ', i,
    ' нет цифр')
else
  writeln('в числе ', k, ' цифр');
readln;
End.

```

#### VI. Тестирование.

1. Запустите программу и введите значения:

$n=12345$

$i=2$

Проверьте. Результат должен быть следующим:

в разряде 2 стоит цифра 4

в числе 5 цифр

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу:

	$n$	$i$	Цифра в разряде $i$	$k$
1	2	2		
2	111111111	5		
3	1020304	1		

3. Добавьте в таблицу свои значения  $n$  и  $i$ . Какие максимальные значения  $n$  и  $i$  можно ввести?

2. Команду `writeln('в числе', k, 'цифр')` заменили командой `writeln('число', n, 'состоит из ', k, 'цифр')`. Какой результат будет получен и почему? Какие изменения нужно внести в программу для получения правильного результата?

3. Изменится ли результат работы программы, если вместо условия цикла  $n \geq 1$  использовать условие  $n > 1$ ?

4. Проверьте работу программы для  $n=0$ . Почему получился такой результат? Что нужно изменить в программе для получения правильного результата?

#### *Индивидуальные задания*

1. Посчитайте сумму цифр числа.

2. Определите, каких цифр в числе больше — чётных или нечётных.

3. Поменяйте порядок цифр числа на обратный.

4. Выбросьте из записи числа  $n$  все цифры, равные 1, оставив при этом прежним порядок остальных цифр. Например, из 5101234  $\rightarrow$  50234.

**Задание 2.** Напишите программу, которая посчитает:

сколько в промежутке от  $n$  до  $k$  чисел, состоящих только из двух различных цифр, причём цифры в числе чередуются.

#### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$  и  $k$ .

II. Определение результатов:  $m$  — количество чисел, удовлетворяющих условию задачи.

### III. Алгоритм решения задачи.

#### 1. Ввод исходных данных.

2. Просматриваем последовательно все числа из промежутка. Для каждого из чисел получаем массив цифр. Затем проверяем, удовлетворяет ли число условию задачи.

3. Для проверки просматриваем элементы массива. Все элементы массива, стоящие на нечётных местах (и на чётных), должны быть равны между собой. Для получения цифр числа и проверки создадим функцию *chislo*.

#### 4. Вывод результата.

### IV. Описание переменных:

все переменные имеют тип *longint*.

### V. Программа:

```
Var i,n,k,m: longint;
```

```
Function chislo(a:longint):boolean;
```

```
Var kk,j: integer;
```

```
    B:array[1..10] of integer;
```

```
    P:Boolean;
```

```
Begin
```

```
    P:=true;
```

```
    If a<10 then
```

```
        P:=false
```

```
    Else
```

```
        begin
```

```
            {получение цифр числа}
```

```
            Kk:=0;
```

```
            While a>=1 do
```

```
                Begin
```

```
                    Kk:=kk+1;
```

```
                    B[kk]:=a mod 10;
```

```
                    a:=a div 10;
```

```
                End;
```

```
                {проверка чисел, стоящих на нечётных  
                    местах}
```

```
                j:=1;
```

```
                while (j<=kk-2) and (b[j]=b[j+2]) do
```

```
                    j:=j+2;
```

```
                if ((j<kk) and (kk mod 2=1)) or
```

```
                    ((j<kk-1) and (kk mod 2=0)) then
```

```
                        p:=false;
```

```
                {проверка чисел, стоящих на чётных  
                    местах}
```

```
                j:=2;
```

```
                while (j<=kk-2) and (b[j]=b[j+2]) do
```

```
                    j:=j+2;
```

```
                if ((j<kk-1) and (kk mod 2=1)) or
```

```
                    ((j<kk) and (kk mod 2=0)) then
```

```
                        p:=false;
```

```
                end;
```

```
                if b[1]=b[2] then
```

```
                    p:=false;
```

```
                chislo:=p;
```

```
            End;
```

```
Begin
```

```
    Write('введите число n');
```

```
    Readln(n);
```

```

Write('введите число k');
Readln(k);
m:=0;
For i:=n to k do
  If chislo(i) then
    m:=m+1;
writeln('в промежутке ',m,' чисел');
End.

```

#### VI. Тестирование.

1. Запустите программу и введите значения:

$n=100$

$k=300$

Проверьте. Результат должен быть следующим:  
в промежутке 18 чисел

Действительно это: 101, 121, 131, 141, 151, 161, 171, 181, 181, 202, 212, 232, 242, 252, 262, 272, 282, 292. Числа 111 и 222 не подходят, так как они состоят только из одинаковых цифр.

2. Заполните таблицу:

	$n$	$k$	Результат
1	1	100	
2	10	100	
3	100	1000	
4	10000	1000000	

3. Добавьте в таблицу свои значения. Подберите такие значения, чтобы в промежутке не было ни одного такого числа. Объясните результаты второго и третьего тестов.

4. Объясните условие  $((j < k) \text{ and } (k \bmod 2 = 1))$  or  $((j < k - 1) \text{ and } (k \bmod 2 = 0))$  (и аналогичное этому в другом операторе if).

5. В каком порядке заносятся цифры числа в массив? Имеет ли значение порядок цифр для получения результата? Измените функцию *chislo* так, чтобы порядок получения цифр был другой.

6\*. Продумайте, как можно решить эту задачу, не проверяя все числа из промежутка.

#### Индивидуальные задания

1. Сколько в промежутке от  $n$  до  $k$  чисел, состоящих только из чётных цифр?

2. Сколько в промежутке от  $n$  до  $k$  чисел, цифры которых образуют возрастающую последовательность?

3. Сколько в промежутке от  $n$  до  $k$  чисел, являющихся палиндромами?

4. Сколько в промежутке от  $n$  до  $k$  чисел, все цифры которых разные?

**Задание 3.** Напишите программу для решения задачи.

1. Проверьте, имеются ли в промежутке от  $n$  до  $k$  числа с суммой цифр, равной  $s$ . Если «да», то посчитайте их количество.

2. Проверьте, имеются ли в промежутке от  $n$  до  $k$  числа с суммой цифр, кратной  $s$ . Если «да», то посчитайте их количество.

3. Проверьте, имеются ли в промежутке от  $n$  до  $k$  числа с суммой цифр, кратной количеству

цифр в числе. Если «да», то посчитайте их количество.

4. В промежутке от  $n$  до  $k$  найдите число с наименьшей суммой цифр.

**Задание 4.** Напишите программу для решения задачи. Используйте подпрограмму, которая распознает простые числа.

В промежутке от  $n$  до  $k$  найдите все пары последовательных простых чисел, сумма которых кратна  $m$ .

#### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$  и  $k$  — начало и конец промежутка,  $m$  — введённое число.

II. Определение результатов: пары чисел, удовлетворяющих условию задачи.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. В промежутке от  $n$  до  $k$  просматриваем все нечётные числа (если в промежуток входит число 2, то начинаем проверку с него). Проверяем с помощью функции *prov*, является ли текущее число простым. Находим следующее простое число. Проверяем, кратна ли сумма двух найденных простых чисел числу  $m$ . Если «да», то печатаем полученные числа.

IV. Описание переменных:

все переменные имеют тип *integer*.

V. Программа:

```
Var i,n,k,m,a,b: integer;
Function prov(y:integer):boolean;
Var ii:integer;
    P:boolean;
Begin
    P:=true;
    ii:=2;
    while (ii<trunc(sqrt(y))+1) and (y mod ii<>0) do
        ii:=ii+1;
    if (y mod ii=0) and (ii<y) then
        p:=false;
    prov:=p;
End;
```

```
Begin
    Write('введите начало промежутка');
    Readln(n);
    Write('введите конец промежутка');
    Readln(k);
    Write('введите число m');
    Readln(m);
    i:=n; {определение первого нечётного
            числа в промежутке}
    if n mod 2=0 then
        i:=i+1;
    If n<=2 then
    begin
        if (2+3) mod m=0 then
```

```

    writeln(2:5,3:5);
    i:=3;
end;
while (i<=k) do
begin
    {поиск первого простого числа}
    while (not prov(i)) and (i<=k) do
        i:=i+2;
    a:=i;
    {поиск второго простого числа}
    i:=i+2;
    while (not prov(i)) and (i<=k)do
        i:=i+2;
    b:=i;
    if (b<=k) and((a+b) mod m=0) then
        writeln(a:5,b:5);
    end;
End.

```

## VI. Тестирование.

1. Запустите программу и введите значения:

$n=10$

$k=20$

$m=3$

Проверьте. Результат должен быть следующим:

11 13

13 17

17 19

Действительно:  $11 + 13 = 24$ ,  $13 + 17 = 30$ ,  $17 + 19 = 36$ . Числа 24, 30, 36 делятся на 3 без остатка.

2. Заполните таблицу:

	<b>n</b>	<b>k</b>	<b>m</b>	<b>Результат</b>
<b>1</b>	2	10	5	
<b>2</b>	10	100	5	
<b>3</b>	1	200	10	
<b>4</b>	10	50	20	

3. Добавьте в таблицу свои значения. Постарайтесь подобрать такие исходные данные, чтобы условию задачи удовлетворяла только одна пара чисел, удовлетворяли все простые числа из промежутка.

4. Объясните условие:  $(b \leq k)$ . Каким может быть результат, если это условие опустить?

5. Добавьте в программу счётчик, подсчитывающий количество полученных чисел. Подберите такие исходные данные, чтобы в промежутке не было ни одной пары чисел, удовлетворяющей условию задачи. Предусмотрите вывод сообщения: «нет чисел, удовлетворяющих условию задачи».

### *Индивидуальные задания*

1. Найдите количество простых чисел из промежутка от  $n$  до  $k$ .

2. Найдите наибольшее простое число из промежутка от  $n$  до  $k$ .

3. Найдите  $k$ -е простое число в последовательности 11, 21, 31, 41, ...

4. Дано натуральное число  $n$ . Выясните, имеются ли среди чисел  $n$ ,  $n + 1$ , ...,  $2n$  числа-близ-

нецы, т. е. простые числа, разность между которыми равна 2. (Например: 11 и 13, 17 и 19.)

**Задание 5.** Напишите программу для решения задачи. При необходимости оформите подпрограмму, которая разлагает число на множители.

1. Дано натуральное число  $n$ . Проверьте, верно ли, что в разложении этого числа на простые множители входит число  $x$  (число  $x$  вводится). Если «да», то выдайте множители на печать.

2. Дано натуральное число  $n$ . Проверьте, верно ли, что в разложении этого числа на простые множители, все множители различны. Если «да», то выдайте множители на печать.

3. Дано натуральное число  $n$ . Получите все числа, меньшие  $n$ , разложение которых на простые множители содержит ровно  $k$  сомножителей.

4. Дано натуральное число  $n$ . Получите все числа, меньшие  $n$ , в разложении которых на простые множители все множители различны.

**Задание 6.** Даны два натуральных числа:  $N$  и  $M$  ( $N < M$ ). Выдайте на печать все числа, расположенные между  $N$  и  $M$  и удовлетворяющие нижеследующим условиям (предусмотреть случай, когда таких чисел нет). Проверку одного числа оформите в виде подпрограммы.

1. Число является совершенным. (Натуральное число является совершенным, если оно равно сумме всех своих делителей, за исключением себя самого. Например,  $6 = 1 + 2 + 3$ .)

2. Число является числом Эйлера. (Натуральное число является числом Эйлера, если оно простое и представимо в виде  $n^2 + n + 41$ .)

3. Число является числом Армстронга. (Натуральное число из  $k$  цифр является числом Армстронга, если сумма его цифр, возведённых в  $k$ -ю степень, равна самому числу.)

4. Число является числом Мерсена. (Натуральное число является числом Мерсена, если оно простое и представимо в виде  $2^n - 1$ .)

**Задание 7.** Напишите программу для решения задачи. Используйте подпрограмму, которая вычисляет НОД( $x, y$ ).

Найдите первое, не меньшее  $n$ , число, не являющееся взаимно простым с  $k$ .

#### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$  и  $k$ .

II. Определение результатов: число, удовлетворяющее условию задачи.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Просматриваем числа, большие  $n$ , и находим НОД текущего числа и числа  $k$ . Останавливаемся, когда НОД не равен 1.

3. Вывод результата.

IV. Описание переменных:

все переменные имеют тип *integer*.

```

V. Программа:
Var n,k: integer;
Function NOD(a,b:integer):integer;
Begin
  While (a<>0) and (b<>0) do
    If a>b then
      a:=a mod b
    else
      b:=b mod a;
  NOD:= a+b;
End;

```

```

Begin
  Write('введите число n');
  Readln(n);
  Write('введите число k');
  Readln(k);
  while NOD(n,k)=1 do
    n:=n+1;
  writeln('искомое число ',n);
End.

```

VI. Тестирование.

1. Запустите программу и введите значения:

n=6  
k=5

Проверьте. Результат должен быть следующим:

искомое число 10

2. Заполните таблицу:

	<b>n</b>	<b>k</b>	<b>Результат</b>
<b>1</b>	<b>3</b>	<b>2</b>	
<b>2</b>	<b>11</b>	<b>100</b>	
<b>3</b>	<b>9</b>	<b>7</b>	
<b>4</b>	<b>123</b>	<b>101</b>	

3. Добавьте в таблицу свои значения. Подберите такие значения, чтобы искомое число совпадало с  $n$ ,  $k$ , располагалось между  $n$  и  $k$ .

4. Как изменится программа, если нужно найти первое, большее  $n$ , число?

5. Найдите наибольшее, меньшее  $n$ , число, удовлетворяющее условию задачи.

### *Индивидуальные задания*

1. Найдите НОД( $a$ ,  $b$ ,  $c$ ).

2. Даны два целых числа:  $a$ ,  $b$ . Сократите обыкновенную дробь  $a/b$ . Ответ получите в виде обыкновенной несократимой дроби.

3. Даны четыре целых числа:  $a$ ,  $b$ ,  $c$ ,  $d$ . Вычислите сумму обыкновенных дробей  $a/b + c/d$ . Ответ получите в виде обыкновенной несократимой дроби.

4. Даны четыре целых числа:  $a$ ,  $b$ ,  $c$ ,  $d$ . Вычислите разность обыкновенных дробей  $a/b - c/d$ . Ответ получите в виде обыкновенной несократимой дроби.

## СИСТЕМЫ СЧИСЛЕНИЯ

### Вопросы для повторения

1. Как перевести число из десятичной системы счисления в систему счисления с другим основанием?
2. Как перевести число из какой-либо системы счисления в десятичную?
3. Как можно хранить число, заданное в какой-либо системе счисления?
4. Как сравнить два числа, заданные в виде массива цифр?
5. Как сложить два числа в заданной системе счисления?
6. Как найти разность двух чисел в заданной системе счисления?
7. Как найти произведение двух чисел в заданной системе счисления?

### Задания для практических работ

**Задание 1.** Напишите программу для решения задачи.

Дано натуральное число  $n$ . Определите число, которое получится, если в двоичной записи числа  $n$  поменять местами первую и вторую цифры, третью и четвёртую, пятую и шестую и т. д. Если количество цифр нечётное, то последняя цифра не изменяется. Например, для числа  $19_{10}$  (двоичная запись  $10011_2$ ) получим  $01101_2$  или  $13_{10}$ .

### Этапы выполнения задания

- I. Определение исходных данных: переменная  $n$ .
- II. Определение результатов:  $m$  — число, полученное после преобразования.
- III. Алгоритм решения задачи.
  1. Вводим число  $n$  и переводим его в двоичную систему счисления, записывая цифры числа в массив в обратном порядке.
  2. Осуществляем перестановки элементов в массиве, согласно условию задачи. Перестановки начинаем с последнего элемента массива.
  3. Переводим преобразованное число в десятичную систему счисления.
  4. Выводим результат.
- IV. Описание переменных:  
цифры числа будут храниться в массиве  $dv$ ; переменная  $nz$  — количество двоичных цифр в числе. Счётчики имеют тип *longint*.

V. Программа:

```
Type mas=array[1..100] of integer;  
Var dv:mas;  
n,nz,m:integer;
```

```
{перевод числа в двоичную систему счисления}  
procedure perevod_2(x:integer; var a:mas;  
var k:integer);  
  
begin  
k:=0;
```

```

while x>0 do
begin
  k:=k+1;
  a[k]:=x mod 2;
  x:=x div 2
end;
end;

{перестановка двоичных цифр}
procedure perest(var a:mas;k:integer);
var i, temp:integer;
begin
  if k mod 2=0 then
    i:=1
  else
    i:=2;
  while i<k do
  begin
    temp:=a[i];
    a[i]:=a[i+1];
    a[i+1]:=temp;
    i:=i+2;
  end;
end;

{перевод числа в десятичную систему
счисления}
procedure perevod_10(a:mas;k:integer; var
rez:integer);

```

```

var i:integer;
begin
  rez:=0;
  for i:=k downto 1 do
    rez:=rez*2+a[i];
  end;

Begin
  Writeln(' введите число n ');
  readln(n);
  perevod_2(n,dv,nz);
  perest(dv,nz);
  perevod_10(dv,nz,m);
  writeln('получили число ',m);
End.

```

#### VI. Тестирование.

1. Запустите программу и введите значение  $n=19$

Проверьте. Результат должен быть следующим:

получили число 13

2. Заполните таблицу:

	<b>n</b>	<b>Результат</b>
1	1	
2	10	
3	31	
4	45	
5	98	

3. Добавьте в таблицу свои значения. Какие числа не изменяются после преобразования? Какие числа увеличиваются?

4. Внесите в программу изменения так, чтобы выводились двоичные числа.

### *Индивидуальные задания*

1. Дано натуральное число  $N$ . Определите число, которое получится, если в двоичной записи числа  $N$  заменить все единицы, стоящие на нечётных местах (1, 3...), считая от конца числа, нулями. Например, для числа  $57_{10}$  (двоичная запись  $111001_2$ ) получим  $101000_2$  или  $40_{10}$ .

2. Дано натуральное число  $N$ . Определите число, которое получится, если в двоичной записи числа  $N$  поменять местами левую и правую половины. Если количество цифр нечётное, то перед первой цифрой приписать 0. Например, для числа  $19_{10}$  (двоичная запись  $10011_2$ , так как количество цифр нечётное, приписываем 0 —  $010011_2$ ) получим  $011010_2$  или  $26_{10}$ .

3. Дано натуральное число  $N$ . Определите число, которое получится, если в двоичной записи числа  $N$  записать вначале все 1, а затем все 0. Например, для числа  $25_{10}$  (двоичная запись  $11001_2$ ) получим  $11100_2$  или  $28_{10}$ .

4. Определите все натуральные числа от  $N$  до  $M$  ( $M > N$ ), запись которых в двоичной и десятичной системах счисления читается одинаково слева направо и справа налево. Например, число  $33_{10}$  (двоичная запись  $100001_2$ ) читается одинаково, число  $44_{10}$  ( $101100_2$ ) — нет.

**Задание 2.** Напишите программу для решения задачи, считая, что числа  $n$  и  $k$  задаются массивом цифр.

Сколько в промежутке от  $n$  до  $k$  чисел, состоящих только из двух различных цифр, причём цифры в числе чередуются?

### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$  и  $k$ .

II. Определение результатов:  $m$  — количество чисел, удовлетворяющих условию задачи.

III. Алгоритм решения задачи.

1. Для ввода исходных данных в массив цифр создадим процедуру *vvod*. Цифры числа вводим через пробел. Конец ввода цифр числа — число  $-1$ . Цифры вводятся в массив по порядку, затем массив «переворачивают» для хранения цифр в обратном порядке.

2. Для просмотра всех чисел из промежутка необходимо моделирование операции сложения текущего числа с единицей, начиная с числа  $n$  (цифры хранятся в массиве  $c1$ ) до тех пор, пока не получим число  $k$  (цифры хранятся в массиве  $c2$ ). Для проверки окончания цикла создадим функцию *равно*.

3. Для проверки условия задачи просматриваем элементы массива. Все элементы массива, стоящие на нечётных местах (и на чётных), должны быть равны между собой. Для проверки цифр текущего числа создадим функцию *chislo*.

#### 4. Вывод результата.

#### IV. Описание переменных:

цифры чисел будут храниться в массивах; переменные  $nz$ ,  $kz$  — количество цифр в числах  $n$ ,  $k$  — имеют тип *longint*. Счётчики имеют тип *longint*.

#### V. Программа:

```
Type mas=array[1..100] of integer;
Var i,m,kz,nz: longint;
    n,k:string;
    c1,c2:mas;
Procedure vvod(var a:mas;var z:longint);
Var j:longint;
    c:integer;
    temp:mas;
Begin
    z:=0;
    repeat
        z:=z+1;
        read(temp[z]);
    Until temp[z]=-1;
    z:=z-1;
    for j:=1 to z do
        a[j]:=temp[z-j+1];
End;
```

```
Function chislo(a:mas;kk:longint):boolean;
Var j: longint;
    P:Boolean;
```

```
Begin
    P:=true;
    If (kk=1) or (a[1]=a[2]) then
        P:=false
    Else
        begin
            { проверка цифр, стоящих на нечет-
                ных местах}
            j:=1;
            while (j<=kk-2) and (a[j]=a[j+2]) do
                j:=j+2;
            if ((j<kk) and (kk mod 2=1)) or
                ((j<kk-1) and (kk mod 2=0)) then
                p:=false;
            { проверка цифр, стоящих на четных
                местах}
            j:=2;
            while (j<=kk-2) and (a[j]=a[j+2]) do
                j:=j+2;
            if ((j<kk-1) and (kk mod 2=1)) or
                ((j<kk) and (kk mod 2=0)) then
                p:=false;
        end;
        chislo:=p;
    End;
```

```
Function ravno(a:mas;z1:longint;b:mas;
                z2:longint):boolean;
Var j:longint;
```

```

P:boolean;
Begin
P:=true;
If z1=z2 then
Begin
j:=z1;
While (j>=1) and (a[j]=b[j]) do
j:=j-1;
if j>=1 then
p:=false;
End
Else
P:=false;
Ravno:=p;
End;

Begin
Writeln(' введите число n, признак конца
ввода - -1');
vvod (c1,nz);
Writeln(' введите число k, признак конца
ввода - -1');
vvod(c2,kz);
m:=0;
repeat
If chislo(c1,nz) then
m:=m+1;
{прибавление 1}
c1[1]:=c1[1]+1;

```

```

i:=1;
while c1[i]>=10 do
begin
c1[i+1]:=c1[i+1]+1;
c1[i]:=c1[i] mod 10;
i:=i+1;
end;
if c1[nz+1]>0 then
nz:=nz+1;
until ravno(c1,nz,c2,kz);
writeln('в промежутке ', m, ' - чисел');
End.

```

#### VI. Тестирование.

1. Запустите программу и введите значения:

для  $n=100$  введите

1 0 0 -1

для  $k=300$  введите

3 0 0 -1

Проверьте. Результат должен быть следующим:

в промежутке 18 чисел

2. Заполните таблицу:

	<b>n</b>	<b>k</b>	<b>Результат</b>
<b>1</b>	<b>1</b>	<b>100</b>	
<b>2</b>	<b>10</b>	<b>100</b>	
<b>3</b>	<b>100</b>	<b>1000</b>	
<b>4</b>	<b>10000</b>	<b>1000000</b>	

3. Добавьте в таблицу свои значения. Подберите такие значения, чтобы в промежутке не было ни одного такого числа.

4. Измените тип данных `mas` так, чтобы нумерация начиналась с нуля. Нулевой элемент используйте для хранения количества цифр числа. Внесите соответствующие изменения в программу.

5. Какие изменения нужно внести в программу, что бы числа  $n$  и  $k$  рассматривались в системе счисления с основанием  $p$ .

### *Индивидуальные задания*

1. Сколько в промежутке от  $n$  до  $k$  чисел, состоящих только из цифр, кратных трём.

2. Сколько в промежутке от  $n$  до  $k$  чисел, являющихся палиндромами, записанными в системе счисления с основанием  $p$ .

3. Сколько в промежутке от  $n$  до  $k$  чисел, являющихся «счастливыми». (Сумма цифр, стоящих на чётных местах, равна сумме цифр, стоящих на нечётных местах.)

4. Сколько в промежутке от  $n$  до  $k$  чисел, все цифры которых разные в системе счисления с основанием  $p$ .

**Задание 3.** Напишите программу для решения задачи.

1. Напечатайте таблицы сложения и умножения в двоичной системе счисления.

2. Напечатайте таблицы сложения и умножения в четверичной системе счисления.

3. Напечатайте таблицы сложения и умножения в восьмеричной системе счисления.

4. Напечатайте таблицы сложения и умножения в шестнадцатеричной системе счисления.

**Задание 4.** Напишите программу для решения задачи. Исходное число в явном виде не получать.

Найдите остаток от деления числа, состоящего из  $k$  цифр, каждая из которых равна  $n$ , на число  $m$ . Число записано в системе счисления с основанием  $p$ . Значения  $k$ ,  $n$ ,  $m$ ,  $p$  вводятся в десятичной системе счисления.

### *Этапы выполнения задания*

I. Определение исходных данных: переменные  $n$ ,  $m$ ,  $k$ ,  $p$ .

II. Определение результатов:  $x$  — число, являющееся остатком.

III. Алгоритм решения задачи.

1. Вводим исходные данные.

2. Число, описанное в условии задачи, может быть записано следующим образом:

$$n * p^{k-1} + n * p^{k-2} + \dots + n * p + n.$$

Преобразуем данное выражение, используя схему Горнера, и получим:

$$n * p * (p * (\dots p * (p + 1) + 1) + \dots) + 1.$$

Для вычисления остатка нужно воспользоваться свойствами остатков. Свойство нахождения остатка от деления суммы двух чисел на число  $m$ :

$$(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m.$$

Свойство нахождения остатка произведения двух чисел:

$$(a * b) \bmod m = (a \bmod m * b \bmod m) \bmod m.$$

3. В цикле находим остаток от деления выражения на число  $m$ .

4. Выводим результат.

IV. Описание переменных:

переменные имеют тип *longint*.

V. Программа:

```
var n,m,k,p,i,x:longint;
```

```
Begin
```

```
  writeln('из каких цифр состоит число');
```

```
  readln(n);
```

```
  writeln('сколько цифр в числе');
```

```
  readln(k);
```

```
  writeln('введите основание системы счисления');
```

```
  readln(p);
```

```
  writeln('на какое число делим');
```

```
  readln(m);
```

```
  x:=0;
```

```
  for i:=1 to k do
```

```
    x:=((x*p) mod m + 1) mod m;
```

```
    x:=(x*n) mod m;
```

```
    writeln('остаток =',x);
```

```
End.
```

VI. Тестирование.

1. Запустите программу и введите значения:

из каких цифр состоит число

2

сколько цифр в числе

3

Введите основание системы счисления  
10

на какое число делим

8

Должен получиться следующий ответ:

остаток =6

Действительно, число 222, записанное в десятичной системе счисления, при делении на 8 даёт остаток, равный 6:  $222=27*8+6$ .

2. Заполните таблицу:

	n	k	p	m	Результат
1	1	5	2	9	
2	3	9	10	1024	
9	9	6	10	9	
4	4	20	5	12	
5	4	100	9	45	

3. Добавьте в таблицу свои значения.

4. Добавьте в программу проверку корректности ввода цифры числа и основания системы счисления.

5. Пусть число  $m$  будет тоже записано в системе счисления с основанием  $p$ . Внесите в программу изменения для решения задачи в этом случае.

### *Индивидуальные задания*

1. Найдите остаток от деления числа, состоящего из  $k$  частей, каждая из которых состоит из трёх введённых цифр, на число  $m$ . Число задано в системе счисления с основанием  $p$ .

2. Найдите остаток от деления числа, заданного в виде массива цифр, на число  $m$ . Число задано в системе счисления с основанием  $p$ .

3. Найдите остаток от деления числа  $2^k$  на число  $m$ .

4. Найдите остаток от деления числа  $k!$  на число  $m$ .

**Задание 5.** Напишите программу, которая будет моделировать арифметические действия над двумя числами в указанных системах счисления.

1. Сложение и вычитание в двоичной системе счисления.

2. Сложение и вычитание в четверичной системе счисления.

3. Сложение и вычитание в восьмеричной системе счисления.

4. Сложение и вычитание в шестнадцатеричной системе счисления.

**Задание 6.** Напишите программу для решения задачи. Во всех вариантах  $n!$  — произведение натуральных чисел от 1 до  $n$ . Необходимо получать все значащие цифры числа  $n!$ .

В числе  $n!$  определите количество пар одинаковых, стоящих рядом, цифр.

#### *Этапы выполнения задания*

I. Определение исходных данных: переменная  $n$ .

II. Определение результатов:  $m$  — количество пар цифр, удовлетворяющих условию задачи.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Получение числа  $n!$ . Для получения всех цифр числа  $n!$  создадим массив цифр и будем моделировать умножение массива цифр на текущее число.

3. Просмотр массива цифр для подсчёта искомого количества цифр.

4. Вывод результата.

IV. Описание переменных:

все переменные имеют тип *longint*.

V. Программа:

```
Var i,n,k,m,q,j: longint;
    a:array[1..10000] of longint;
Begin
    Write('введите число n');
    Readln(n);
    if n=0 then
        n:=1;
    a[1]:=1;
    k:=1; { количество цифр в числе n!}
    for i:=1 to n do
    begin
        for q:=1 to k do
            a[q]:=a[q]*i; {умножение на текущее число}
        for j:=1 to k do {проверка цифр числа}
```

```

begin
  a[j+1]:=a[j+1]+a[j] div 10;
  a[j]:=a[j] mod 10;
end;
while a[k+1]<>0 do {увеличение ко-
                    личество цифр}
begin
  k:=k+1;
  a[k+1]:=a[k+1]+a[k] div 10;
  a[k]:=a[k] mod 10;
end;
end;
{вывод на экран числа n!}
for i:=k downto 1 do
  write(a[i]);
writeln;
m:=0;
for i:=1 to k-1 do
  if a[i]=a[i+1] then m:=m+1;
  writeln('в числе', n, '!', m, ' пар одинаковых
цифр');
End.

```

#### VI. Тестирование.

Запустите программу и введите значение

$n=9$

Проверьте. Результат должен быть следующим:

362880

в числе  $9!$  1 пар одинаковых цифр

Действительно, это пара из двух восьмёрок.

#### 2. Заполните таблицу:

	<b>n</b>	<b>Результат</b>
1	5	
2	10	
3	100	
4	1000	

3. Добавьте в таблицу свои значения. Для какого максимального числа можно получить  $n!$ ?

4. Объясните необходимость цикла `while a[k+1]<>0 do` для увеличения количества цифр. Почему нельзя обойтись проверкой только одной цифры старшего разряда?

5. Часто при работе с длинными числами элемент массива хранит не одну, а четыре цифры длинного числа. (Почему 4, а, например, не 5?) Это позволяет увеличить скорость работы программы с длинными числами и работать с ещё большими числами. Измените программу так, чтобы каждый элемент массива хранил по четыре цифры числа  $n!$ . Обратите внимание на вывод значения  $n!$ . Как теперь проверять пары одинаковых цифр?

#### *Индивидуальные задания*

Определите:

- 1) сколько раз встретилась в числе  $n!$  цифра  $k$ ;
- 2) каких цифр в числе  $n!$  больше: чётных или нечётных;
- 3) какая из цифр встречается в числе  $n!$  чаще всего;
- 4) в процентах вхождение каждой из цифр в число  $n!$ .

## ПРИЛОЖЕНИЯ

### Некоторые математические формулы

Треугольник ( $a, b, c$  — длины сторон треугольника;  $a, b, \gamma$  — величины углов треугольника,  $S$  — площадь,  $P$  — периметр,  $p$  — полупериметр,  $h$  — высота,  $m$  — медиана,  $l$  — биссектриса,  $r$  — радиус вписанной окружности,  $R$  — радиус описанной окружности).

$$P = a + b + c.$$

$$S = 1/2ah_a;$$

$$S = 1/2ab \sin \gamma;$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}; \text{ (формула Герона);}$$

$$S = abc/4R;$$

$$S = pr.$$

Длина медианы, высоты и биссектрисы, проведённых из вершины В:

$$m_b = 1/2 \sqrt{2a^2 + 2c^2 - b^2};$$

$$h_b = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{b};$$

$$l_b = \frac{2\sqrt{acp(p-b)}}{a+c}.$$

Теорема косинусов:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha;$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta;$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma.$$

Теорема синусов:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} = 2R;$$

Прямоугольный треугольник ( $a, b$  — катеты,  $c$  — гипотенуза,  $S$  — площадь).

Теорема Пифагора:

$$c^2 = a^2 + b^2.$$

Площадь:

$$S = 1/2ab.$$

Соотношения между длинами сторон:

$$a = c \cos \beta = c \sin \alpha = b \operatorname{tg} \alpha = b \operatorname{ctg} \beta.$$

$$b = c \cos \alpha = c \sin \beta = a \operatorname{ctg} \alpha = a \operatorname{tg} \beta.$$

Радиусы вписанной и описанной окружностей:

$$r = \frac{a+b-c}{2}; \quad R = c/2.$$

Равносторонний треугольник ( $a$  — длина стороны,  $S$  — площадь,  $r$  — радиус вписанной окружности,  $R$  — радиус описанной окружности).

$$R = a\sqrt{3}/3;$$

$$r = a\sqrt{3}/6;$$

$$S = a^2\sqrt{3}/4.$$

Трапеция ( $a, b$  — основания,  $h$  — высота,  $S$  — площадь).

$$S = h(a+b)/2.$$

Правильный  $n$ -угольник ( $a$  — длина стороны,  $S$  — площадь,  $P_n$  — периметр,  $r$  — радиус вписанной окружности,  $R$  — радиус описанной окружности).

$$S = \frac{nR^2}{2} \sin \frac{2\pi}{n};$$

$$S = \frac{P_n r}{2};$$

$$S = \frac{P_n R}{2} \cos \frac{\pi}{n}.$$

Окружность и круг ( $S$  — площадь,  $r$  — радиус,  $d$  — диаметр,  $C$  — длина окружности).

$$S = \pi r^2 = \pi d^2/4;$$

$$C = 2\pi r = \pi d;$$

## Некоторые функции языка Pascal

Название функции	Действие	Тип аргумента(ов)	Тип результата
<i>abs (x)</i>	Вычисляет модуль числа $x$	Любой числовой	Соответствующему аргументу
<i>arctan (x)</i>	Вычисляет арктангенс числа $x$	real	real
<i>copy (a,i,c)</i>	Вырезает подстроку строки $a$ из $c$ символов, начиная с позиции $i$	$a$ — string, $i, c$ — integer	string
<i>cos (x)</i>	Вычисляет косинус числа $x$	real	real
<i>exp (x)</i>	Вычисляет экспоненту числа $x$	real	real
<i>frac (x)</i>	Вычисляет дробную часть числа $x$	real	real
<i>int (x)</i>	Получает целую часть числа $x$	real	real
<i>length (s)</i>	Возвращает длину строки $s$	string	integer

## Окончание таблицы

Название функции	Действие	Тип аргумента(ов)	Тип результата
<i>odd (x)</i>	Проверяет число $x$ на нечётность	longint	boolean
<i>pi</i>	Возвращает значение числа $\pi$ с 19 знаками после запятой	Без аргумента	real
<i>pos (s1,s2)</i>	Ищет первое вхождение строки $s1$ в строке $s2$	$s1, s2$ — string	byte
<i>random (x)</i>	Возвращает случайное число из интервала от 0 до $x$	word	word
<i>round (x)</i>	Округляет число $x$	real	longint
<i>sin (x)</i>	Вычисляет синус числа $x$	real	real
<i>sqr (x)</i>	Вычисляет квадрат числа $x$	Любой числовой	Соответствующему аргументу
<i>sqrt (x)</i>	Вычисляет квадратный корень из числа $x$	real	real
<i>trunc (x)</i>	Отбрасывает дробную часть числа $x$	real	longint

## Некоторые процедуры языка Pascal

Название процедуры	Действие	Тип аргумента(ов)
<i>dec (i, n)</i>	Уменьшает переменную <i>i</i> на <i>n</i>	<i>i, n</i> — longint
<i>delete (s, n, k)</i>	Удаляет <i>k</i> символов из строки <i>s</i> , начиная с символа с номером <i>n</i>	<i>s</i> — string <i>i, n</i> — integer
<i>exit</i>	Осуществляет немедленный выход из текущего блока	Без параметров
<i>halt</i>	Останавливает выполнение программы	
<i>inc (i, n)</i>	Увеличивает переменную <i>i</i> на <i>n</i>	<i>i, n</i> — longint
<i>insert (s1, s2, n)</i>	Вставляет строку <i>s1</i> в строку <i>s2</i> , начиная с позиции <i>n</i>	<i>n</i> — integer <i>s1, s2</i> — string
<i>randomize</i>	Инициализирует датчик случайных чисел	Без параметров
<i>str (x, s)</i>	Преобразует число <i>x</i> в строковое представление <i>s</i>	<i>x</i> — число <i>s</i> — string

## Окончание таблицы

Название процедуры	Действие	Тип аргумента(ов)
<i>val (s, x, c)</i>	Преобразует строку <i>s</i> в число <i>x</i> и возвращает позицию ошибки <i>c</i> (если <i>c=0</i> , преобразование выполнено, иначе <i>c</i> возвращает номер позиции, начиная с которой преобразование невозможно)	<i>x</i> — число (real или integer) <i>s</i> — string <i>c</i> — integer

## СОДЕРЖАНИЕ

### ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ

#### С ПОМОЩЬЮ КОМПЬЮТЕРА . . . . . 3

Вопросы для повторения . . . . . 3

Задания для практических работ . . . . . 4

#### ЛОГИЧЕСКИЕ ЗАДАЧИ . . . . . 18

Вопросы для повторения . . . . . 18

Задания для практических работ . . . . . 18

### ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ

#### ДЛЯ РЕШЕНИЯ ЗАДАЧ . . . . . 47

Вопросы для повторения . . . . . 47

Задания для практических работ . . . . . 48

#### РЕШЕНИЕ ГЕОМЕТРИЧЕСКИХ ЗАДАЧ . . . . 62

Вопросы для повторения . . . . . 62

Задания для практических работ . . . . . 63

### РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ

#### МАССИВОВ . . . . . 84

Вопросы для повторения . . . . . 84

Задания для практических работ . . . . . 86

### ЗАДАЧИ ЦЕЛОЧИСЛЕННОЙ

#### АРИФМЕТИКИ . . . . . 100

Вопросы для повторения . . . . . 100

Задания для практических работ . . . . . 101

#### СИСТЕМЫ СЧИСЛЕНИЯ . . . . . 117

Вопросы для повторения . . . . . 117

Задания для практических работ . . . . . 117

#### ПРИЛОЖЕНИЯ . . . . . 135

Некоторые математические  
формулы . . . . . 135

Некоторые функции языка Pascal . . . . . 137

Некоторые процедуры языка Pascal . . . . . 139

Учебное издание

Лапо Анжелика Ивановна

# Информатика в задачах

8—9 классы

Пособие для учащихся учреждений  
общего среднего образования  
с белорусским и русским языками обучения

2-е издание

Редактор	<i>Я. Н. Важник</i>
Компьютерный набор	<i>В. Ю. Зарецкая</i>
Компьютерная вёрстка	<i>В. Ю. Зарецкая</i>
Корректор	<i>Я. Н. Важник</i>

Подписано в печать 10.03.2014 г. Формат 60×84<sup>1/16</sup>.

Бумага офсетная. Печать офсетная.

Усл. печ. л. 8,37. Уч.-изд. л. 3,4.

Тираж 800 экз. Заказ № 23.

Издатель и полиграфическое исполнение:

РУП «Издательство “Адукацыя і выхаванне”».

Свидетельство о государственной регистрации

издателя, изготовителя, распространителя печатных изданий

№ 1/19 от 02.08.2013.

№ 2/17 от 26.11.2013.

Ул. Будённого, 21, 220070, г. Минск.