

- 7* Напишите программу, которая нарисует ряд фигурок вдоль края графического окна. Используйте фигурки, которые рисовали в задании 6.
- 8 Выполните задания для примера 18.6.
1. Измените в программе значение $r = 10$ на $r = 20$. Почему получился такой рисунок? Поэкспериментируйте со значениями радиуса, установив прозрачную заливку.
 2. Какие изменения нужно внести в программу, чтобы экран заполнялся кругами с радиусом 20 без пересечений?
 3. Внесите изменения в программу так, чтобы все круги были красными или разноцветными.
 4. Внесите в программу изменения так, чтобы графическое окно можно было заполнять кругами введенного радиуса.
- 9* Напишите программу, которая заполнит все графическое окно:
1. Грибами (пример 18.2).
 2. Фигурками из задания 6.

§ 19. Использование основных алгоритмических конструкций для решения практических задач

19.1. Использование числовых последовательностей

Числовые последовательности позволяют описывать многие процессы, происходящие в природе и обществе.

Например, последовательность чисел 2, -1, 0, 2, 0, 1, -2 может задавать значения температуры по дням недели; последовательность 746, 751, 758 — значения средней заработной платы сотрудников предприятия за квартал и т. д.

Последовательности могут задаваться формулой, в которой значение элемента зависит от того, какой у него номер в последовательности (пример 19.1).

Другим способом задания элементов последовательности является определение значения нового элемента

Подтверждением важности числовых последовательностей является тот факт, что создана целая энциклопедия числовых последовательностей¹.

Пример 19.1. Элементы последовательности нечетных положительных чисел можно описать с помощью формулы $a_n = 2n - 1$. В этой формуле n — номер элемента в последовательности. Минимальное значение числа $n = 1$. Используя формулу, получим последовательность: 1, 3, 5, 7,

Элементы последовательности могут быть действительными числами. Например, формула $a_n = \frac{n}{n^2 + 1}$ задает следующую последовательность: 0.5, 0.4, 0.3, 0.235, 0.192,

¹ Онлайн-энциклопедия целочисленных последовательностей. Режим доступа: <http://oeis.org/?language=russian> (дата доступа: 17.01.2018).

Пример 19.2. Одной из наиболее известных последовательностей является последовательность Фибоначчи: 1, 1, 2, 3, 5, 8, 13, Несложно заметить, что каждый ее элемент, начиная с третьего, равен сумме двух предыдущих. Это можно записать так: $f_n = f_{n-1} + f_{n-2}$, $f_1 = 1$, $f_2 = 1$.

Пример 19.3. Рассмотрим последовательность 2, 4, 8, 16, Каждое число в этой последовательности является степенью числа 2, поэтому можно задать последовательность формулой $a_n = 2^n$. С другой стороны, каждый элемент последовательности, начиная со второго, в два раза больше предыдущего. Получим формулу $a_n = 2a_{n-1}$ (для $n > 1$, $a_1 = 2$).

Пример 19.4.

V. Программа:

```
var k, a: integer;
begin
  write('Количество k = '); read(k);
  for var n := 1 to k do
    begin
      a:= 2*n; write(a, ' ');
    end;
end.
```

VI. Тестирование.

Запустить программу и ввести значения $k = 5$. Результат:

Окно вывода

```
Количество k = 5
2 4 6 8 10
```

Запустить программу и ввести значение $k = 100$. Результат:

Окно вывода

```
Количество k = 100
2 4 6 8 10 12 14 16 18 20 22 24 26 28
30 32 34 36 38 40 42 44 46 48 50 52
54 56 58 60 62 64 66 68 70 72 74 76
78 80 82 84 86 88 90 92 94 96 98 100
102 104 106 108 110 112 114 116 118
120 122 124 126 128 130 132 134 136
138 140 142 144 146 148 150 152 154
156 158 160 162 164 166 168 170 172
174 176 178 180 182 184 186 188 190
192 194 196 198 200
```

через значение предыдущего (пример 19.2).

Есть последовательности, которые можно задавать как первым, так и вторым способом (пример 19.3). Последовательности могут строиться из случайных чисел.

Пример 19.4. Вывести на экран первые k четных чисел.

Этапы выполнения задания

I. Исходные данные: k (количество чисел).

II. Результат: k четных чисел, начиная с 2.

III. Алгоритм решения задачи.

1. Ввод числа k .

2. Для получения четного числа запишем формулу $a_n = 2n$.

3. Так как количество чисел заранее известно, то для их получения можно воспользоваться циклом **for**.

4. Текущее число будем хранить в переменной a . Значение a вычисляется по формуле и зависит от значения n — счетчика цикла. Переменная n будет изменяться от 1 (номер первого четного числа) до k (номер последнего числа).

5. Полученные числа будем выводить в цикле через пробел.

IV. Описание переменных: k , a — integer.

Пример 19.5. Вывести на экран все элементы последовательности Фибоначчи меньше x (x вводится).

Этапы выполнения задания

I. Исходные данные: x (граница для чисел).

II. Результат: числа Фибоначчи меньше x .

III. Алгоритм решения задачи.

1. Ввод числа x .

2. Вывод первых двух элементов.

3. Числа Фибоначчи, начиная с третьего, получают по формуле $a_n = a_{n-1} + a_{n-2}$ ($a_1 = a_2 = 1$). Для вывода чисел понадобятся три переменные: значение a , которое нужно вывести, и два предыдущих значения — b и c .

n	1	2	3	4	5	6	7
Числа Фибоначчи	1	1	2	3	5	8	13
Текущий шаг			c	b	a		
Следующий шаг				c	b	a	

После вывода значения a нужно «сдвинуть» значения переменных: $c := b$; $b := a$. Начальные значения: $c := 1$; $b := 1$; $a := 2$.

4. Так как количество чисел заранее неизвестно, то для их вычисления нужно использовать цикл **while**. Условие продолжения работы цикла: $a < x$.

5. В цикле выводим текущее значение a , «сдвигаем» значения переменных и получаем новое значение a .

IV. Описание переменных: x, a, b, c — integer.

Числа Фибоначчи обладают множеством интересных математических свойств. Например:

1. Могут быть вычислены по формуле

$$\text{ле Бине: } f_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}.$$

2. Отношение следующего числа к предыдущему является постоянной величиной ≈ 1.618 (начиная с 13-го). Это число называют золотым сечением.

3. Для трех последовательных чисел Фибоначчи верно соотношение Кассини: $f_{n+1}f_{n-1} - f_n^2 = (-1)^n$.

Пример 19.5.

V. Программа:

```
var a, b, c, x: integer;
begin
  write('Граница x = '); read(x);
  c := 1; b := 1; a := 2;
  write(c, ' ', b, ' ');
  while a < x do
  begin
    write(a, ' ');
    c := b; b := a; a := b + c;
  end;
end.
```

VI. Тестирование.

Запустить программу и ввести значение $x = 100$. Результат:

Окно вывода

```
Граница x = 100
1 1 2 3 5 8 13 21 34 55 89
```

Чтобы узнать, сколько чисел получили в качестве результата, определим переменную k . Переменная будет увеличивать свое значение на 1 каждый раз, когда выводится очередное число. После завершения цикла можно вывести значение k .

```
var a, b, c, x, k: integer;
begin
  write('Граница x = ');
  read(x);
  c := 1; b := 1; a := 2;
  write(c, ' ', b, ' ');
  k := 2;
  while a < x do
  begin
    write(a, ' '); k := k+1;
    c := b; b := a; a := b + c;
  end;
  writeln;
  write('k = ', k);
end.
```

Результат для $x = 1000$.

Окно вывода

```
Граница x = 1000
1 1 2 3 5 8 13 21 34 55 89 144 233
377 610 987
k = 16
```

Пример 19.6.

V. Программа:

```

var n, k, a: integer;
begin
  write('Количество билетов n = ');
  read(n); k := 0;
  for var i:= 1 to n do
  begin
    a:= random(1,100); write(a, ' ');
    if a mod 5 = 0 then
      k := k+1;
    end;
    writeln;
    writeln('Выиграло ', k,
      ' билета(-ов)');
  end.

```

VI. Тестирование. Запустить программу и ввести значение $n = 20$.
Результат:

Окно вывода

```

Количество билетов n = 20
56 81 10 34 3 20 43 34 57 54 92 59 47
23 44 90 83 79 29 91
Выиграло 3 билета(-ов)

```

При одном и том же значении n программа может выдавать различные результаты, поскольку числа получаются случайным образом.

Пример 19.7.

V. Программа:

```

var m : integer;
    a, S: real;
begin
  write('Количество дней m = ');
  read(m); S := 0;
  for var n := 1 to m do
  begin
    a := n*n*n/(sqrt(n*n*n)-n+1);
    S := S+a;
  end;
  writeln('Всего бактерий =
    = ', S, ' млн');
end.

```

VI. Тестирование.

Запустить программу и ввести значение $m = 3$. Результат:

Окно вывода

```

Количество дней m = 3
Всего бактерий = 13.8230024772972 млн

```

Пример 19.6. Катя и Петя решили организовать благотворительную лотерею. Для этого они случайным образом генерируют номер билета. Номера билетов принадлежат промежутку $[1..100]$. Выигрышным билетом будет тот, номер которого кратен 5. Определить, сколько будет выигрышных билетов среди n сгенерированных Катей и Петей.

Этапы выполнения задания

I. Исходные данные: n (количество билетов).

II. Результат: k — количество выигрышных билетов.

III. Алгоритм решения задачи.

1. Ввод числа n .

2. До начала генерации количество выигрышных билетов равно нулю ($k := 0$).

3. Так как количество билетов заранее известно, то для получения их номеров можно воспользоваться циклом `for`.

4. Текущий номер билета будем хранить в переменной a . Номера билетов будем получать по формуле $a = \text{Random}(1,100)$ и выводить на экран. Для каждого номера будем проверять, равен ли 0 остаток от деления числа на 5. И если равен, то увеличим на 1 значение переменной k .

5. Вывод результата.

IV. Описание переменных: n, k, a — `integer`.

19.2. Нахождение суммы элементов числовой последовательности

Пример 19.7. В лаборатории выводят полезные бактерии. Экспериментально

было установлено, что количество бактерий (в млн) зависит от номера дня, в который проводится эксперимент, следующим образом: $a_n = \frac{n^3}{\sqrt{n^3 - n + 1}}$.

Определите, сколько бактерий вывели за m дней.

Этапы выполнения задания

I. Исходные данные: m (число дней).

II. Результат: S (общее количество бактерий).

III. Алгоритм решения задачи.

1. Ввод числа m .

2. Для вычисления общего количества бактерий необходимо последовательно прибавлять количество бактерий, выведенных в текущий день, к уже полученному количеству бактерий. Начальное значение суммы равно 0.

3. Так как количество бактерий заранее известно, для вычисления суммы можно воспользоваться циклом **for**.

4. Количество бактерий в текущий день будем хранить в переменной a . Значение a зависит от значения n — счетчика дней. Переменная n изменяется от 1 до m .

5. Вывод результата S .

IV. Описание переменных: m — integer, S , a — real.

19.3. Возведение числа в степень

Пример 19.8. Возвести вещественное число a в целую степень n .

Этапы выполнения задания

I. Исходные данные: a (основание степени), n (показатель степени).

II. Результат: S (значение степени).

III. Алгоритм решения задачи.

Пример 19.7. Продолжение.

Для проверки правильности результата можно посчитать значение суммы на калькуляторе:

$$a_1 = 1; a_2 = \frac{2^3}{\sqrt{8 - 2 + 1}} \approx 4.38;$$

$$a_3 = \frac{3^3}{\sqrt{27 - 3 + 1}} \approx 8.44; S \approx 13.82.$$

Запустить программу и ввести значение $m = 30$. Результат:

Окно вывода

```
Количество дней m = 30
Всего бактерий = 2613.36198051392 млн
```

Пример 19.8.

V. Программа:

```
var n, m: integer;
    a, S: real;
begin
  write('Основание a = '); read(a);
  write('Показатель n = '); read(n);
  S := 1; m := abs(n);
  for var i:= 1 to m do
    S := S*a;
  if n<0 then S:= 1/S;
  writeln('Степень = ',S);
end.
```

VI. Тестирование программы.

Запустить программу и ввести значения $a = 5$, $n = 3$. Результат:

Окно вывода

```
Основание a = 5
Показатель n = 3
Степень = 125
```

Запустить программу и ввести значения $a = 3$, $n = 0$. Результат:

Окно вывода

```
Основание a = 3
Показатель n = 0
Степень = 1
```

Запустить программу и ввести значения $a = 5$, $n = -2$. Результат:

Окно вывода

```
Основание a = 5
Показатель n = -2
Степень = 0.04
```

VII. Правильность вычислений проверить на калькуляторе.

Пример 19.9.

V. Программа:

```

var k:integer;
    x,y,h:real;
begin
  writeln('Количество значений');
  readln(k); x := -3; h := 0.5;
  for var n:= 1 to k do
  begin
    y := (x+2)/(x*x+3);
    writeln(x:7:2,y:10:3); x := x+h;
  end;
end.

```

VI. Тестирование программы.

Запустить программу и ввести значение $k = 5$. Результат:

Окно вывода	
Количество значений	
5	
-3.00	-0.083
-2.50	-0.054
-2.00	-0.000
-1.50	-0.059
-1.00	-0.250

Добавим в программу вывод границ таблицы:

```

var k:integer;
    x,y,h: real;
begin
  writeln('Количество значений');
  readln(k); x := -3; h := 0.5;
  writeln('-----');
  writeln('| x | y |');
  writeln('-----');
  for var n := 1 to k do
  begin
    y := (x+2)/(x*x+3);
    writeln('|',x:7:2,'|',y:10:3,'|');
    x := x+h;
  end;
  writeln('-----');
end.

```

Результат при $k = 4$:

Окно вывода	
Количество значений	
4	
x	y
-3.00	-0.083
-2.50	-0.054
-2.00	0.000
-1.50	0.095

1. Ввод чисел a, n .

2. Для возведения числа в целую неотрицательную степень нужно последовательно умножать на основание степени то значение, которое получили на предыдущем шаге. Это вытекает из равенства: $a^n = a^{n-1} \cdot a$.

3. Если степень отрицательная, то результат можно получить из равенства: $a^{-n} = \frac{1}{a^n}$. Количество повторений цикла m можно определить как $m = |n|$.

4. Для вычисления произведения можно воспользоваться циклом **for**. Начальное значение степени равно 1.

5. В конце проверим, является ли значение n положительным или отрицательным. Если оно отрицательно, то нужно изменить значение переменной S .

6. Вывод результата S .

IV. Описание переменных: m, n — integer, a, S — real.

19.4. Построение таблицы значений функции

Пример 19.9. Вывести на экран таблицу значений функции $y = \frac{x+2}{x^2+3}$.

Количество значений вводится. Начальное значение $x = -3$, значения аргумента выводятся с шагом $h = 0,5$.

Этапы выполнения задания

I. Исходные данные: k (количество точек).

II. Результат: k значений аргумента и соответствующих им значений функции.

III. Алгоритм решения задачи.

1. Ввод числа k .

2. Для получения таблицы нужно в цикле вычислять и выводить значение

аргумента и соответствующее ему значение функции:

1) начальное значение аргумента $x = -3$. Для получения очередного значения аргумента нужно к текущему значению прибавить шаг h ;

2) значение функции вычисляется по формуле $y = \frac{x+2}{x^2+3}$;

3) полученные значения выводятся на экран. Для того чтобы значения выводились строго одно под другим, нужно использовать форматный вывод. Для этого задать количество позиций для вывода значения и определить количество цифр после запятой. Запись $x:7:2$ означает, что для вывода переменной используется 7 позиций, после запятой выводятся 2 цифры.

3. Поскольку количество точек известно, воспользуемся циклом **for**.

IV. Описание переменных: k — integer, x, y, h — real.

19.5. Выделение цифр из числа

Пример 19.10. Дано натуральное число n . Вывести цифры числа по одной в строке (начиная с разряда единиц). Определить, сколько цифр в числе.

Этапы выполнения задания

I. Исходные данные: n (число).

II. Результат: z (текущая цифра числа), k (количество цифр в числе n).

III. Алгоритм решения задачи.

1. Ввод исходных данных — число n .

2. Определение начального значения счетчика для количества цифр ($k := 0$).

3. Количество цифр числа равно количеству десятичных разрядов в числе.

Понятие числа возникло в глубокой древности из практической потребности людей. Для записи чисел используют цифры. Любая информация в компьютере представляется с помощью всего двух цифр (0 и 1).

Числовой код имеет каждая страна мира, цифры задают пин-код банковской карты. Сегодня с помощью цифр можно получить числовой образ практически любого объекта.

Пример 19.10.

V. Программа:

```
var k,n,z: integer;
begin
  write('Введите n = ');
  read(n);
  k := 0;
  while n > 0 do
  begin
    //Текущая цифра
    z := n mod 10;
    writeln(z);
    //Уменьшение числа в 10 раз
    n := n div 10;
    //Подсчет кол-ва цифр
    k := k + 1;
  end;
  writeln('в числе ', k,
    ' цифр(-а/-ы)');
end.
```

VI. Тестирование.

Запустить программу и ввести значение $n = 13579$. Результат:

```
Окно вывода
Введите n = 13579
9
7
5
3
1
В числе 5 цифр (-а/-ы)
```

Запустить программу и ввести значение $n = 1$. Результат:

```
Окно вывода
Введите n = 10
1
В числе 1 цифр (-а/-ы)
```

Алгоритм Евклида — алгоритм для нахождения наибольшего общего делителя двух целых чисел. Алгоритм назван в честь древнегреческого математика Евклида (III в. до н. э.), который впервые описал его в книгах «Начала». Это один из старейших численных алгоритмов, используемых в наше время.

Евклид использовал данный алгоритм не только для чисел, но и для геометрических величин: длин, площадей, объемов. В XIX в. алгоритм был обобщен на другие математические объекты. Сегодня алгоритм Евклида используется при шифровании данных.

Пример 19.11. Алгоритм Евклида для чисел 42 и 24:

№	a	b
1	42	24
2	18 (42–24)	24
3	18	6 (24–18)
4	12 (18–6)	6
5	6 (12–6)	6

Пример 19.12.

V. Программа:

```
var a,b:integer;
begin
  write('Значки у Иры a = ');
  read(a);
  write('Значки у Игоря b = ');
  read(b);
  while a<>b do
    if a>b then
      a := a - b
    else
      b := b - a;
  writeln('Всего друзей = ',a);
end.
```

VI. Тестирование.

Запустить программу и ввести значения: $a = 42$, $b = 24$. Результат:

Окно вывода

```
Значки у Иры a = 42
Значки у Игоря b = 24
Всего друзей = 6
```

Для нахождения каждой цифры числа нужно:

- разделить число на 10;
- найти целую часть от деления и остаток (остаток и будет очередной цифрой) и увеличить счетчик количества цифр;
- вывести полученную цифру;
- поскольку количество цифр в числе заранее неизвестно, будем использовать цикл **while**. Пока целая часть от деления больше 0, в числе еще есть цифры и нужно перейти к выполнению пункта а), иначе все цифры найдены.

4. Вывод значения переменной k .

IV. Описание переменных: k , n , z — integer.

19.6. Наибольший общий делитель двух чисел

Наибольшим общим делителем (НОД) для двух целых чисел называют наибольший из их общих делителей. Пример: для чисел 42 и 24 наибольший общий делитель равен 6.

Существуют несколько алгоритмов нахождения НОД. С одним из них вы познакомились на уроках математики. Нужно разложить каждое из чисел на простые множители, выбрать общие и перемножить.

Рассмотрим другой алгоритм, который называется алгоритм Евклида.

1. Из большего числа вычитаем меньшее.

2. Если получается 0, то числа равны друг другу и это значение является НОД.

3. Если результат вычитания не равен 0, то большее число заменяем на разность большего и меньшего.

4. Переходим к пункту 1.

(Рассмотрите пример 19.11.)

Пример 19.12. Ира и Игорь коллекционируют значки. У Иры в коллекции a значков, а у Игоря — b . Поскольку значков много, ребята решили поделиться своими значками с друзьями. Какое наибольшее количество общих друзей может быть у Иры и Игоря, если каждый из них хочет разделить все свои значки между друзьями без остатка? Например, если $a = 42$ и $b = 24$, то значки можно разделить, если у Иры и Игоря 1, 2, 3 или 6 общих друзей. Наибольшее количество — 6.

Этапы выполнения задания

I. Исходные данные: a и b (количество значков у Иры и у Игоря).

II. Результат: наибольшее количество общих друзей.

III. Алгоритм решения задачи.

1. Ввод чисел a, b .

2. Поскольку значки нужно делить без остатка, то ответом на задачу может быть только общий делитель чисел a и b . Среди всех делителей нужно найти наибольший.

3. Для решения задачи напомним программу вычисления НОД(a, b) по алгоритму Евклида. Пока числа a и b не равны, выполняем следующее:

1) сравниваем два числа;

2) если $a > b$, заменяем a на разность $a - b$, иначе заменяем b на разность $b - a$.

4. Вывод результата. Вывести можно как значение a , так и b (поскольку они равны).

IV. Описание переменных: a, b — integer.

Пример 19.13*. Написать программу вычисления НОД(x, y, z).

Пример 19.12. Продолжение.

Для значений $a = 1449$, $b = 596$ получим:

Окно вывода

```
Значки у Иры a = 1449
Значки у Игоря b = 596
Всего друзей = 1
```

Данный результат означает, что числа 1449 и 596 взаимно простые и все значки можно отдать только одному другу.

*В Pascal, кроме процедур, часто используются вспомогательные алгоритмы в виде функций. Функция всегда возвращает значение, которое нужно присвоить какой-либо переменной или использовать в любом выражении:
 $b := \text{abs}(x)$, $t := 2 + \text{sqrt}(a)$.

Общий вид функции:

```
function <имя>(<список
параметров>:тип): тип результата;
var ... //Может отсутствовать
begin
  <команды>
  <имя> := <значение>;
end;
```

Пример 19.13*.

V. Программа:

```
var x,y,z,d,f:integer;
function NOD
  (a,b:integer):integer;
begin
  while a<>b do
    if a>b then
      a := a-b
    else
      b := b-a;
  NOD := a;
end;
begin
  write('Введите x = ');
  read(x);
  write('Введите y = ');
  read(y);
  write('Введите z = ');
  read(z);
  d := NOD(x,y);
  f := NOD(d,z);
  writeln('НОД = ',f);
end.
```

Пример 19.13*. *Продолжение.*

VI. Тестирование.

Запустить программу и ввести значения: $x = 26$, $y = 143$, $z = 65$. Результат:

```
Окно вывода
Введите x = 26
Введите y = 143
Введите z = 65
НОД = 13
```

Запустить программу и ввести значения: $x = 354$, $y = 847$, $z = 125$. Результат:

```
Окно вывода
Введите x = 354
Введите y = 847
Введите z = 125
НОД = 1
```

Данный результат означает, что числа 354, 847 и 125 взаимно простые.

Этапы выполнения задания

I. Исходные данные: x , y и z (три числа).

II. Результат: НОД (x , y , z).

III. Алгоритм решения задачи.

1. Ввод чисел x , y , z .

2. Используем то, что $\text{НОД}(x, y, z) = \text{НОД}(\text{НОД}(x, y), z)$. То есть сначала вычислим $d = \text{НОД}(x, y)$, а затем $f = \text{НОД}(d, z)$.

3. Для вычисления НОД двух чисел составим функцию $\text{NOD}(a, b)$. Команды функции NOD рассмотрены в примере 19.12.

4. Вывод результата.

IV. Описание переменных: x , y , z , d , f — integer.



Упражнения

- 1 Выполните задания для примера 19.4.
 1. Внесите в программу изменения так, чтобы числа выводились в обратном порядке — от большего к 2.
 2. Какие изменения нужно внести в программу, чтобы вывести все четные числа меньше введенного числа x ?
 3. Внесите изменения в программу так, чтобы выводились числа, кратные 3, 5; кратные введенному числу x .
 4. Измените программу так, чтобы можно было вывести все числа последовательности, заданной формулой $a_n = \frac{n}{n^2 + 1}$ (пример 19.1).
- 2 Выполните задания для примера 19.5.
 1. Запустите программу для разных значений x .
 2. Какое максимальное значение x можно ввести?
 3. Замените в программе тип integer на тип int64. Сколько чисел последовательности Фибоначчи можно найти теперь?
 4. Измените программу так, чтобы она выводила значение числа Фибоначчи по введенному номеру.
- 3 Выполните задания для примера 19.6.
 1. Запустите программу несколько раз для одного и того же значения (например, 20). Какое число получается в ответе чаще всего? Почему?

2. Измените диапазон случайных чисел на $[1, 1000]$ и проверьте, какое число получается в результате чаще других для тех же 20 элементов.
3. Измените программу так, чтобы вычислялось количество чисел, кратных введенному числу x .
4. Измените программу так, чтобы можно было посчитать количество чисел из промежутка $[a; b]$, a и b вводятся.
- 4* Напишите программу, которая будет выводить на экран элементы последовательности трибоначчи — первые элементы последовательности: 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, Каждый элемент, начиная с четвертого, равен сумме трех предыдущих: $a_n = a_{n-1} + a_{n-2} + a_{n-3}$.
 1. По заданному n вывести элемент последовательности.
 2. Для заданного x вывести элементы последовательности меньше x .
- 5 Выполните задания для примера 19.7.
 1. Замените в решении задачи цикл **for** на цикл **while**.
 2. Выполните программу для $m = 2000$. Почему в ответе сумма = NaN? Какие изменения нужно внести в программу для получения результата? Найдите значение для $m = 2000$, $m = 10\,000$.
- 6 Найдите сумму первых m элементов последовательности. Число m вводится. Элементы последовательности задаются формулой $a_n = \frac{1}{n^3}$.
 1. На сколько большим должно быть значение m , чтобы программа выдала ответ «сумма = Infinity»? Почему так произошло?
 2. Какие изменения нужно внести в программу для получения правильного результата?
 3. Замените в решении задачи цикл **for** на цикл **while**. Найдите сумму для $m = 20\,000$, $m = 1\,000\,000$.
- 7 Выполните задание для примера 19.8. Измените форму вывода результата таким образом, чтобы результат выводился в виде $a^n = S$ (например, для значений $a = 2$, $n = 3$ должно быть напечатано $2^3 = 8$).
- 8 Факториалом числа n называют произведение всех натуральных чисел, не превосходящих n . Обозначают факториал так: $n!$. По определению факториал числа 0 равен 1. Напишите программу, которая вычислит значение факториала целого неотрицательного числа n . Для проверки можно использовать следующее: $0! = 1$; $2! = 2$; $5! = 120$, $10! = 3\,628\,800$.
- 9 Выполните задание для примера 19.9. Замените в решении задачи цикл **for** на цикл **while**. В качестве условия в цикле **while** можно использовать следующее: $x <= 3$.
- 10 Постройте таблицы значений для указанных функций.
 1. $y = x^2 - 5x - 3$, $x \in [-3, 3]$, вводится значение шага h .
 2. $y = 2 + \frac{3x^3 - 7}{x}$, $x \in [a, b]$ ¹, вводятся значения a , b и количество точек.

¹ Подсказка: $h = \frac{a+b}{k-1}$.

11 Выполните задания для примера 19.10.

1. 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?

2. Изменится ли результат работы программы, если вместо условия цикла $n < 0$ использовать условие $n > 1$?

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

12 Программу из примера 19.10 изменили. Сформулируйте задачу, которая решается с помощью данной программы.

```

var i,k,n,z: integer;
begin
  write('Введите n = ');
  read(n);
  write('Введите i = ');
  read(i);
  k:=0;
  while n > 0 do
  begin
    z := n mod 10; //Текущая цифра
    k := k + 1;
    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');
  end.

```

13 Дано натуральное число n . Определите, каких цифр в числе больше — четных или нечетных.

14 Дано натуральное число n . Выведите номера разрядов, в которых стоят цифры, кратные 3, или сообщение, что таких цифр нет.

15 Задано натуральное число. Напишите программу, которая для числа с нечетным количеством цифр выведет на экран цифру, стоящую на средней позиции числа.

Если в числе четное количество цифр, то вывести соответствующее сообщение. Например, для числа 23 452 ответом будет 4, а для числа 56 — сообщение «В числе четное количество цифр».

16 Число является палиндромом, если оно одинаково читается слева направо и справа налево. Напишите программу, которая по введенному натуральному числу определит, является оно палиндромом или нет. Примеры: 12321, 6776 — палиндромы, 12335 — нет.

17* Натуральное число n называется числом Армстронга, если сумма цифр числа, возведенных в n -ю степень, где n — количество цифр в числе, равна самому числу. Например, $153 = 1^3 + 5^3 + 3^3$. Напишите программу, которая найдет:

1. Все трехзначные числа Армстронга.
2. Все четырехзначные числа Армстронга.

18 В 1626 г. индейцы продали остров Манхэттен за 20 долларов. Если бы эти деньги были помещены в банк на текущий счет и ежегодный прирост составлял бы x %, какова была бы стоимость капитала в этом году? Напишите программу, которая ответит на данный вопрос.

19 Имеются два сосуда. В первом находится C_1 л воды, а во втором C_2 л воды. Из первого сосуда переливают половину воды во второй, а затем из второго переливают половину воды в первый (одно переливание) и т. д. Напишите программу, которая определит, сколько воды окажется в каждом из сосудов после k переливаний.

20 Дано натуральное число n . Написать программу, которая выведет все числа, взаимно простые с n , а также числа, которые меньше его.

21* Измените программу из примера 19.13:

1. Найти НОД четырех чисел.
2. Найти НОК (наименьшее общее кратное) двух чисел.
3. Вводятся числитель и знаменатель правильной дроби. Сократите дробь.
- 4*. Две правильные дроби заданы своими числителями и знаменателями. Найдите их сумму. Ответ выведите в виде смешанной дроби.