

Время выполнения: 240 мин.

1.1. При каких n последовательность $a_n = \frac{10^n}{n!}$ ($n=1,2,\dots$) принимает наибольшее значение?

1) 9; 10

2.1. Дан равнобедренный прямоугольный треугольник с катетом 10. Рассмотрим равнобедренные прямоугольные треугольники, вершины которых лежат на разных сторонах исходного треугольника. Найдите наименьшее значение площадей таких треугольников.

1) 10

3.1. Найдите все значения a , при которых среди решений неравенства

$$2\sqrt{x} > a(x+2)$$

имеются ровно два целых числа.

1) $a \in \left[\frac{2}{3}; \frac{2\sqrt{3}}{5} \right)$

4.1. Математическая модель некоторого процесса описывается неравенством:

$$\frac{B(x^2 - Ax)}{x^2 + 2x - 3} \leq 0$$

Параметры A и B – это характеристики процесса, которые могут принимать любые действительные значения. Для анализа процесса необходимо построить компьютерную модель, которая по заданным характеристикам A и B , будет вычислять, сколько целочисленных решений есть у данного неравенства. Реализуйте такую модель на языке программирования. Программа должна считывать два числа – параметры A и B соответственно. Результат работы программы: количество целочисленных решений неравенства, если оно конечно, и символ $\&$, если неограниченно.

Пример.

Ввод	Вывод
-5,5 2	3
2,241 -1,5	&

1) Решение на языке Паскаль

```
var a,b:real;
begin
  read(a,b);
  if b<=0 then
    write('&')
  else
    if a=0 then
      write(3)
    else
      write(trunc(abs((a+ abs(a))/2-1))+trunc(abs((a-abs(a))/2+3)))
end.
```

5.1. Система счисления Майя является комбинацией пятеричной и двадцатеричной систем счисления. Первые четыре цифры обозначаются соответствующим количеством точек. Далее каждая очередная пятерка обозначается горизонтальной чертой. В системе счисления Майя нашлось место и для цифры ноль – она обозначается символом, похожим на глаз. На рисунке 1 приведены цифры Майя.

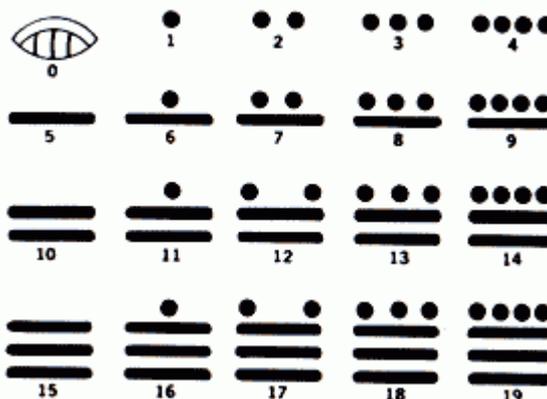


Рис.1 – Цифры Майя



Так у Майя изображалась число 20. При представлении числа цифры писались не слева направо, как привычно для нас, а сверху вниз (т.е. цифра старшего разряда находилась на «вершине»). Логично предположить, что «вес» разряда получается из «веса» предыдущего разряда домножением на 20, но в этой цепочке есть исключение: вместо $20 \times 20 = 400$ для следующего разряда берется $20 \times 18 = 360$, поэтому и «веса» следующих разрядов не являются степенью 20: 1; 20; 360; 7 200; 144 000; 2 880 000 и т.д. (Существуют разные гипотезы о причинах такого нарушения).

Приведем еще несколько примеров (рис. 2):

$$19 \times 360 + 13 \times 20 + 13 = 7113_{10}$$

$$10 \times 360 + 0 \times 20 + 7 = 3607_{10}$$

$$2 \times 7200 + 0 \times 360 + 6 \times 20 + 5 = 14525_{10}$$

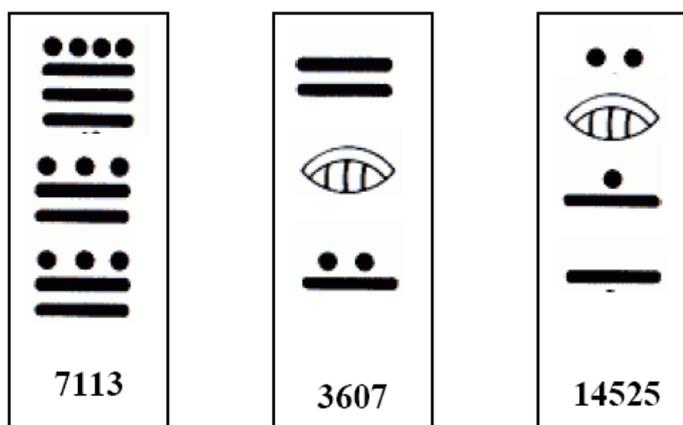


Рис. 2 Примеры

Напишите программу, которая будет переводить целые числа из диапазона от 0 до 10 000 000 в систему Майя. На вход подается число. На выходе – набор строк. При выводе для отображения точек использовать звездочки, для изображения горизонтальных линий – тире, для нуля – обычную цифру ноль. Между цифрами добавлять пустые строчки.

Пример:

ВВОД	ВЫВОД
7113	**** - - - *** - - *** - -
3607	- - 0 ** - - -
14525	** 0 * - - -

```

1) Решение на языке Паскаль
var n:longint;
procedure maya(n:longint;k:integer);
var p,c,i:integer;
begin
if n<>0 then
begin
if k = 2 then p := 18 else p := 20;
c := n mod p;
maya(n div p,k+1);
if c =0 then
writeln(0)
else
begin
if c mod 5 <> 0 then
begin
for i := 1 to c mod 5 do
write('*');

```

```
writeln  
end;  
for i := 1 to c div 5 do  
  writeln('-')  
end;  
writeln  
end  
end;  
  
begin  
  readln (n);  
  maya(n,1);  
end.
```

6.1. Выпишем все буквы русского алфавита без «Ё» и «Й», но с добавлением пробела «_». Каждой букве соответствует десятичный порядковый номер – её код.

	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Приведём лишь часть таблицы, где буквам соответствуют двоичные пятизначные коды:

А	Б	В	Г	...	Э	Ю	Я
00001	00010	00011	00100	...	11101	11110	11111

Главная проблема столь простого способа передачи информации заключается в защите передаваемых данных. Один из способов защиты информации – это ввод ключа, или кодового слова, который складывают с кодом сообщения перед его посылкой по каналу связи. Сложение производится в двоичной системе счисления по правилам: $0 + 0 = 0$, $1 + 0 = 1$, $0 + 1 = 1$, $1 + 1 = 0$ (такая операция называется сложением по модулю два; обратите внимание, что при сложении единиц в старший разряд ничего не добавляется).

Приведём пример. Пусть дана строка: БУДУ ДОМА, кодовое слово: КЛЮЧ

Б	У	Д	У	_	Д	О	М	А
00010	10011	00101	10011	00000	00101	01110	01100	00001
01010	01011	11110	10111	01010	01011	11110	10111	01010
К	Л	Ю	Ч	К	Л	Ю	Ч	К
01000	11000	11011	00100	01010	01110	10000	11011	01011
З	Ш	Ь	Г	К	О	Р	Ы	Л

Результат кодирования: ЗЩЬГКОРЫЛ

Без кодового слова восстановить исходное сообщение невозможно. С другой стороны, алгоритм расшифровки очень прост – достаточно снова прибавить к сообщению ключ по тем же правилам:

З	Ш	Ь	Г	К	О	Р	Ы	Л
01000	11000	11011	00100	01010	01110	10000	11011	01011
01010	01011	11110	10111	01010	01011	11110	10111	01010
К	Л	Ю	Ч	К	Л	Ю	Ч	К
00010	10011	00101	10011	00000	00101	01110	01100	00001
Б	У	Д	У	_	Д	О	М	А

Напишите программу, которая на вход получает две строки, каждая длиной не более 250 символов: строку-сообщение и ключ, а на выходе выдаёт зашифрованное или расшифрованное сообщение.

Например,

Ввод	Вывод
БУДУ ДОМА КЛЮЧ	ЗЩЬГКОРЫЛ
БУБСМЦГААРЛ ЦЦ АБРАКАДАБРА	ВСТРЕЧА В КАФЕ

1) Решение на языке Паскаль

```
var s,key,news,str:string;
i,j,k1,k,k2:integer;
```

```
begin
  str := ' АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ';
  readln(s);
  readln(key);

  j := 0;
  news := "";
  for i := 1 to length(s) do
  begin
    j := j + 1;
    k := pos(copy(s,i,1),str)-1;
    k1 := pos(copy(key,j,1),str)-1;
    k2 := (k xor k1) + 1;
    news := news + copy(str,k2,1);
    j := j mod length(key)
  end;
  write(news)
end.
```

7.1. Карту местности можно представить в виде целочисленной матрицы $D(N,M)$ (размерность матрицы не превосходит 100×100). Каждый элемент матрицы описывает состояние соответствующего квадрата местности и может принимать значения: 0 – поляна или время в часах, необходимое для расчистки участка, от 1 до 5.

Группа туристов высаживается в квадрате с координатами (1, 1) и должна пройти в квадрат с координатами (N, M). Переход от одного квадрата к другому может происходить только через вертикальную или горизонтальную границу между квадратами, группа может двигаться только влево или вниз. Длина пути группы измеряется количеством пройденных квадратов, включая начальный и конечный квадрат.

Требуется составить программу, которая находит для данной карты местности такой путь группы, при котором на расчистку тратится наименьшее время. Если таких путей несколько, то можно вывести любой.

Вывести также время, которое необходимо для прохождения найденного пути.

Порядок ввода исходных данных:

N M

D(1, 1) D(1, 2) ... D(1, M)

D(2, 1) D(2, 2) ... D(2, M)

...

D(N, 1) D(N, 2) ... D(N, M)

Порядок вывода результатов:

Время_расчистки

LEFT Y1

X2 ВОТТОМ

...

Xk Yk

Пример:

Вход	Выход
3 5	10
1 2 1 0 4	1 1
4 1 0 2 2	1 2
1 2 3 3 2	2 2
	2 3
	2 4
	2 5
	3 5

1) Решение на языке Паскаль

```
var map:array[1..100,1..100]of integer;
path, hours:array[0..100,0..100]of integer;
i,j,n,m:integer;
procedure print(i,j:integer);
begin
if (i<>0)and(j<>0) then {пока не достигнуто начало пути}
begin
if path[i,j]=1 then {если пришли сверху, то распечатываем путь в верхнюю ячейку}
print(i-1,j)
else
print(i,j-1); {если пришли слева, то распечатываем путь в левую ячейку}
write(' [',i,',',j,'] '); {печатаем саму вершину}
end;
end;
```

```
begin
readln(n,m); { ввод размера карты и ее содержания }
for i:=1 to n do
  for j:=1 to m do
    readln(map[i,j]);
    { во вспомогательный массив, куда будем заносить минимальное время пути до каждой
    ячейки, добавляем нулевой столбец и нулевую строку, куда заносим заведомо большое
    время }
    for i:=0 to n do hours[i,0]:=100;
    for j:=0 to m do hours[0,j]:=100;
    { для старта }
    hours[0,1]:=0;
    hours[1,0]:=0;
    { для каждой ячейки выясняем как туда лучше, т.е. с минимальным временем, попасть:
    сверху или слева, в зависимости от этого заполняем ячейку массива hours как сумму
    минимального времени пути до этой ячейки + время на расчистку, если она требуется, а в
    массив path записываем признак 1, если мы пришли в эту ячейку с минимальным
    временем сверху, 2 – если слева }
    for i:=1 to n do
      for j:=1 to m do
        begin
          if hours[i-1,j]<hours[i,j-1] then
            begin
              hours[i,j]:=hours[i-1,j]+map[i,j];
              path[i,j]:=1
            end
          else
            begin
              hours[i,j]:=hours[i,j-1]+map[i,j];
              path[i,j]:=2
            end
          end;
        writeln('min=',hours[n,m]); { минимальное время пути в ячейку[n,m] }
        { рекурсивно распечатываем путь, восстанавливая по массиву path в обратную сторону }
        print(n,m); writeln;
      end.
```