

Задача 1. Измерение информации

100 баллов

Ограничение по времени	1 секунда
Ограничение по памяти	1 MiB
Имя входного файла	VINF.IN
Имя выходного файла	VINF.OUT

Жанна Лютикова очень любит информатику. В связи с этим она постоянно пытается измерить объём информации различных сообщений. Её всегда удивляли странности, которые преследуют производные единицы измерения информации, например, в одном килограмме 1000 граммов и это логично, но в одном килобайте 1024 байта, а в байте вообще 8 бит, что, по её мнению, очень странно (ведь Жанна ещё только изучает информатику). Поэтому у Жанны иногда возникают затруднения при переводе единиц измерения информации.

Жанне известен информационный объём, заданный в одной из производных единиц. Возможные обозначения единиц В (байт), KiB (килобайт), MiB (мегабайт), GiB (гигабайт), PiB (петабайт). Помогите ученице с переводом единиц.

Требуется по указанному информационному объёму определить его значение в битах.

Формат входных данных

Входные данные содержат одну строку, состоящую из двух значений, разделённых одним пробелом. Первое значение A ($0 < A \leq 10^5$) – целое число – количество информации, второе – текстовое – единица измерения информации.

Формат выходных данных

Выходные данные должны содержать одно число: количество информации в битах.

Пример входных и выходных данных

VINF.IN	VINF.OUT
12 MiB	100663296

Решение.

Предлагаемая задача на простую обработку строки текста, преобразование типов данных, знание единиц измерения информации.

Возможное решение на языке Pascal.

```
program solve01;
Uses sysutils, Math;
var f: textFile; s: string;
    a, k: integer;
begin
assignFile(f, 'vinf.in'); reset(f);
readln(f, s); close(f);
k := pos(' ', s);
a := StrToInt(copy(s, 1, k-1));
s := copy(s, k+1, length(s)-k);
a := a * 8;
assignFile(f, 'vinf.out'); rewrite(f);
if s = 'B' then write(f, a);
if s = 'KiB' then write(f, a*power(2,10):0:0);
if s = 'MiB' then write(f, a*power(2,20):0:0);
if s = 'GiB' then write(f, a*power(2,30):0:0);
```

```
if s = 'PiB' then write(f, a*power(2,40):0:0);  
closeFile(f);  
end.
```

Возможное решение на языке Python.

```
a = open ('VINF.IN')  
s,b = a.readline().strip().split()  
a.close()  
s = int(s)  
if b == 'B':  
    s = s*2**13  
elif b == 'MiB':  
    s = s*2**23  
elif b == 'GiB':  
    s = s*2**33  
elif b == 'PiB':  
    s = s*2**43  
a = open('VINF.OUT')  
a.write(str(s))  
a.close()
```

Задача 2. Необычная архивация

100 баллов

Ограничение по времени:	1 секунда
Ограничение по памяти	1 MiB
Имя входного файла	ARCHIVE.IN
Имя выходного файла	ARCHIVE.OUT

Жанна Лютикова, изучая разнообразное программное обеспечение, добралась до архиваторов. «Какие удивительные программы! – подумала Жанна. – Интересно, как они работают?». И так как Жанна очень интересовалась информатикой (см. задачу 1), то она тут же придумала, как может работать архиватор. Она решила так. Пусть есть некоторая последовательность символов латиницы, например «bbaaaaaaaaaaaccs».

Если буква встречается более 2 раз подряд, то она нуждается в «архивации». Последовательность до 9 символов включительно заменяется цифрой (количество повторений символа) и самим символом. В ситуации, когда одинаковых символов следующих подряд более 9, первые 9 заменяются цифрой «9» и символом, оставшаяся часть такой последовательности далее рассматривается по вышеуказанным правилам.

Тогда для выбранного примера, следуя логике Жанны, заархивированная строка будет выглядеть следующим образом «bb9aaa3c»

Требуется помочь Жанне автоматизировать процесс «архивации» предложенной строки.

Формат входных данных

Входные данные содержат одну строку, которую требуется заархивировать по алгоритму Жанны.

Формат выходных данных

Выходные данные должны содержать одну строку – результат архивации.

Пример входных и выходных данных

ARCHIVE.IN	ARCHIVE.OUT
bbaaaaaaaaaaccc	bb9aaa3c

Решение.

Предлагаемая задача на обработку строки текста, согласно описанному алгоритму.

Возможное решение на языке Pascal.

```
program solve02;
  uses SysUtils;
//Рекурсивная функция кодирования последовательности одинаковых символов
function code(a: string): string;
begin
  if length(a) < 3 then result := a else
    if length(a) < 10 then result := IntToStr(length(a)) + a[1]
    else begin
      a := copy(a,10,length(a)-9);
      result := '9' + a[1] + code(a);
    end;
end;
//Основная программа
var i: integer; f: textFile;
    s,t,res: string;
begin
  assignFile(f, 'ARCHIVE.IN'); reset(f);
  readln(f, s); closeFile(f);
  s := s + '!'; t := ''; res := '';
//Будем разбивать строку на подстроки - последовательности одинаковых символов
  for i := 1 to length(s)-1 do begin
    t := t + s[i];
    if s[i] <> s[i+1] then begin
//Если в череде повторяющихся символов встречен отличный от
//предыдущих, то подстроку отправляем на кодирование
      res := res + code(t);
      t := '';
    end;
  end;
//Учтём, что последний символ может быть пустым и тогда его кодировать не надо
  if t <> '' then res := res + code(t);
//Вывод результата
  assignFile(f, 'ARCHIVE.OUT'); rewrite(f);
  writeln(f, res); closeFile(f);
end.
```

Возможное решение на языке Python.

```
fname = 'ARCHIVE.IN'
a = open (fname)
s = a.readline().strip()+'*'
a.close()
buf = s[0]
```

```
s = s[1:]
count = 1
outs = ''
t = True
while t:
    #print(buf)
    if buf[-1] == s[0]:
        buf +=s[0]
        s = s[1:]
        if len(buf)==9:
            outs+='9'+buf[0]
            buf = s[0]
            s = s[1:]
    else:
        if len(buf)<3:
            outs+=buf
        else:
            outs += str(len(buf))+buf[0]
            buf = s[0]
            s = s[1:]
    if len(s) == 0: t = False
f = open (fname[:-2]+'OUT','w')
f.write(outs)
f.close()
```

Задача 3. «Змейка»

100 баллов

Ограничение по времени: 1 секунда
Ограничение по памяти 1 MiB
Имя входного файла SPIRAL.IN
Имя выходного файла SPIRAL.OUT

Жанна Лютикова часто задумывалась и рисовала на клетчатой бумаге. Она, вспомнив об известном математике Станиславе Мартине Уламе, решила писать числа в клетки с использованием определённой закономерности. Улам делал это по спирали. Жанна же придумал свой способ: заполнить квадратную область цифрами в порядке возрастания, «змейкой», начиная с нижнего левого угла вверх (вправо-вверх-влево-вверх и т.д.). После достижения последней цифры она начинала заново с 0. И так до тех пор, пока не получалась квадратная область со стороной N клеток.

5432
8901
7654
0123

Например, для $N = 4$ получался рисунок, приведённый на изображении.

Требуется написать программу, генерирующую изображение размером $N \times N$ по описанным правилам.

Формат входных данных

Входные данные содержат в первой строке одно число N ($0 < N < 1000$), – сторона квадрата изображения.

Формат выходных данных

N строк по N цифр без пробелов – сгенерированное изображение.

Пример входных и выходных данных

SPIRAL.IN	SPIRAL.OUT
3	678 543 012
4	5432 8901 7654 0123

Решение.

Предлагаемая задача на моделирование процессов согласно описанному алгоритму, либо поиск закономерностей.

Возможное решение на языке Pascal.

```
program solve03;
  var i, j, n, k, p, t: integer;
      f: textFile;
      a: array of array of byte;
begin
  assignFile(f, 'SPIRAL.IN'); reset(f);
  read(f, n); closeFile(f);
  //Создадим динамический массив для хранения данных
  setLength(a, n); k := 0; t := 0; p := 1;
  for i := 0 to n-1 do setLength(a[i], n);
  //Смоделируем заполнение в обратном направлении, отразив результат по вертикали
  for i := 0 to n-1 do begin
    for j := 0 to n-1 do begin
      a[i,k] := t;
      inc(t); if t = 10 then t := 0;
      k := k + p;
    end;
  //Чередование направления заполнения массива
    p := -1 * p;
    k := k + p;
  end;
  //Осуществим вывод в файл в правильном направлении
  assignFile(f, 'SPIRAL.OUT'); reWrite(f);
  for i := n-1 downto 0 do begin
    for j := 0 to n-1 do write(f, a[i,j]);
    writeln(f);
  end;
  closeFile(f);
end.
```

Задача 4. В поисках выхода

100 баллов

Ограничение по времени: 1 секунда
Ограничение по памяти: 1 MiB
Имя входного файла: SEARCH.IN
Имя выходного файла: SEARCH.OUT

Жанна Лютикова сидела дома и скучала. От скуки она рисовала на листочке в клетку стрелки: вверх, вниз, влево и вправо. А так как мысли её витали очень далеко, то стрелки она ставила случайным образом, как придётся. Очнувшись от своих мыслей, Жанна увидела на листке бумаги заполненную стрелками прямоугольную область. Так как Жанна очень любила информатику, то она тут же стала думать о том, что же делать с этими стрелками. «Стрелка – это направление движения, – подумала Жанна, – значит надо двигаться по ним». Тут же встал закономерный вопрос: а можно ли покинуть прямоугольник, если двигаться от клетки к клетке по стрелкам и если да, то сколько есть клеток, из которых можно выйти из прямоугольника?

→	↑	→	↓
←	→	↑	↓
↑	←	↑	←

Например, для приведённого рисунка клеток, из которых, двигаясь по стрелкам, можно покинуть квадрат – 5 (отмечены выделенным фоном).

Требуется по заданному заполнению прямоугольника определить, из скольких клеток можно, двигаясь по стрелкам, его покинуть.

Формат входных данных

Входные данные содержат в первой строке два числа N и M ($0 < N, M < 1000$), разделённых пробелом – количество строк и столбцов прямоугольника. Далее N строк по M чисел 1, 2, 3 или 4, разделённых одним пробелом – направления стрелок (1 – вверх ↑, 2 вправо →, 3 – вниз ↓, 4 – влево ←).

Формат выходных данных

Одно число – количество клеток, из которых возможен выход за пределы прямоугольника.

Пример входных и выходных данных

SEARCH.IN	SEARCH.OUT
3 4 2 1 2 3 4 2 1 3 1 4 1 4	5

Решение.

Предлагаемая задача является модификацией классической задачи на поиск выхода из лабиринта. Для её решения может быть применён немного модифицированный классический алгоритм рекурсивного обхода лабиринта.

Возможное решение на языке Pascal.

```
program project04;  
  var a: array of array of byte;  
      f: textFile;  
      n, m, i, j: integer;  
  
  //Процедура рекурсивного обхода лабиринта. Результат достижения  
  //выхода за его пределы в переменной res  
  procedure Go(y, x: integer; var res: boolean);
```

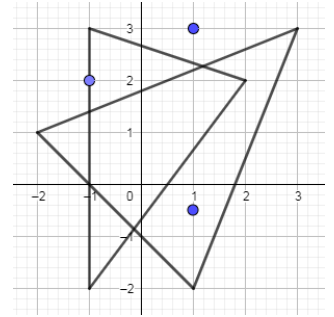
```
var k: byte;
begin
  if a[y,x] = 0 then begin
    if (x=0) or (y=0) or (x=m+1) or (y=n+1) then res := true
      else res := false;
    end else begin
      k := a[y,x];
      a[y,x] := 0;
      case k of
        1: Go(y-1,x,res);
        2: Go(y,x+1,res);
        3: Go(y+1,x,res);
        4: Go(y,x-1,res);
      end;
      a[y,x] := k;
    end;
  end;
end;
//Основная программа
var t: boolean; count, i1, j1: integer;
begin
  assignFile(f, 'search.in'); reset(f);
//Чтение данных в динамический массив
  Readln(f, n, m); count := 0;
  SetLength(a, n+2); for i := 0 to n+1 do SetLength(a[i], m+2);
  For i := 1 to n do
    For j := 1 to m do Read(f, a[i,j]);
  closeFile(f);
//Обход лабиринта из каждой его клетки
  for i1 := 1 to n do for j1 := 1 to m do begin
    t := false;
    Go(i1,j1,t);
    if t then inc(count);
  end;
//Вывод результата
  assignFile(f, 'search.out'); rewrite(f);
  write(f, count); closeFile(f);
end.
```

Задача 5. Невидимки

100 баллов

Ограничение по времени: 1 секунда
Ограничение по памяти: 1 MiB
Имя входного файла: ROBOTS.IN
Имя выходного файла: ROBOTS.OUT

Жанна Лютикова, поднаторев в информатике (см. предыдущие задачи), разрабатывает систему видеонаблюдения. На некоторой площадке установлено N камер. Через каждую камеру можно наблюдать за областью, представляющей собой треугольник, задаваемый положением камеры и двумя точками, задающими максимальный угол обзора и дальность наблюдения (рис. 1). Всё, что попадает в этот треугольник, в том числе и на его стороны, может увидеть оператор системы. Жанне известны все области обзора камер.



По площадке перемещаются M роботов, каждый из которых, в определённый момент времени t , одинаковый для всех устройств, передаёт информацию о координатах своего положения, так как все устройства снабжены системой ГЛОНАСС.



Любопытная Жанна заинтересовалась тем, сколько устройств в этот определённый момент времени не видны камерами видеонаблюдения?

Требуется определить количество роботов, которых в данный момент времени **не** может увидеть оператор.

Формат входных данных

Входные данные содержат в первой строке два целых числа N и M ($0 < N, M < 1000$), разделённых пробелом – количество камер и количество роботов соответственно. Далее N строк по 6 вещественных чисел, разделённых одним пробелом – координаты (x_i, y_i) камеры и двух точек, задающих её обзор. Далее M строк по 2 вещественных числа, разделённых одним пробелом – координаты (x_j, y_j) роботов в определённый момент времени t .

Формат выходных данных

Одно число – количество **невидимых** роботов.

Пример входных и выходных данных

ROBOTS.IN	ROBOTS.OUT
2 3 -1 -2 -1 3 2 2 1 -2 -2 1 3 3 1 -0.5 -1 2 1 3	1

Решение.

Для решения задачи воспользуемся функцией определения положения точки относительно треугольника, заданного его вершинами. Для этого требуется последовательно пройти по всем сторонам треугольника определить, с какой стороны точка располагается относительно прямой, которой принадлежит сторона. Если точка для

всех прямых (которым принадлежат стороны), располагается с одной стороны, то она принадлежит треугольнику.

Возможно использование и других методов проверки положения точки относительно треугольника. Например, метод геометрического луча.

Возможное решение на языке Pascal.

```
program solve05;
//Функция определения наличия точки внутри треугольника,
//либо на его границе(aPx, aPy – координаты точки)
function IsPointInTriangle(aAx, aAy, aBx, aBy, aCx, aCy,
    aPx, aPy: single): boolean;
    function Q(ax, ay, bx, by, atx, aty: single): single;
    begin
        Result := atx * (by - ay) + aty * (ax - bx) + ay * bx -
            ax * by;
    end;
    var q1, q2, q3: single;
begin
    q1 := Q(aAx, aAy, aBx, aBy, aPx, aPy);
    q2 := Q(aBx, aBy, aCx, aCy, aPx, aPy);
    q3 := Q(aCx, aCy, aAx, aAy, aPx, aPy);
    Result := ((q1 >= 0) and (q2 >= 0) and (q3 >= 0)) or
        ((q1 < 0) and (q2 < 0) and (q3 < 0));
end;
//Основная программа
var f: textFile;
    a: array of array of single;
    n, m, i, j, k: integer;
    x, y: single;
begin
    assignFile(f, 'ROBOTS.IN'); reset(f); readln(f, n, m);
//Чтение данных о треугольниках в динамический массив
    setLength(a, 6);
    for i := 0 to 5 do setLength(a[i], n);
    for i := 0 to n-1 do
        readln(f, a[0,i], a[1,i], a[2,i], a[3,i], a[4,i], a[5,i]);
    k := 0;
//Чтение координат точек из файла и проверка принадлежности хотя бы одному
//треугольнику
    for i := 0 to m-1 do begin
        readln(f, x, y);
        for j := 0 to n-1 do
            if IsPointInTriangle(a[0,j], a[1,j], a[2,j], a[3,j],
                a[4,j], a[5,j], x, y)
                then begin inc(k); break; end;
    end;
    closeFile(f);
//Вывод результата
    assignFile(f, 'ROBOTS.OUT'); rewrite(f);
    write(f, m-k); closeFile(f);
end.
```