

9 -11 класс

Задача А. Ставки

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Перед началом тараканьих бегов всем болельщикам было предложено сделать по две ставки на результаты бегов. Каждая ставка имеет вид "Таракан №А придет раньше, чем таракан №В".

Организаторы бегов решили выяснить, могут ли тараканы прийти в таком порядке, чтобы у каждого болельщика сыграла ровно одна ставка из двух (то есть чтобы ровно одно из двух утверждений каждого болельщика оказалось верным). Считается, что никакие два таракана не могут прийти к финишу одновременно.

Формат входного файла

В первой строке входных данных содержатся два разделенных пробелом натуральных числа: число K , не превосходящее 10, - количество тараканов и число N , не превосходящее 100, - количество болельщиков. Все тараканы пронумерованы числами от 1 до K . Каждая из следующих N строк содержит 4 натуральных числа A, B, C, D , не превосходящих K , разделенных пробелами. Они соответствуют ставкам болельщика "Таракан №А придет раньше, чем таракан №В" и "Таракан №С придет раньше, чем таракан №D".

Формат выходного файла

Если завершить бега так, чтобы у каждого из болельщиков сыграла ровно одна из двух ставок, можно, то следует вывести номера тараканов в том порядке, в котором они окажутся в итоговой таблице результатов (сначала номер таракана, пришедшего первым, затем номер таракана, пришедшего вторым и т. д.) в одну строку через пробел. Если таких вариантов несколько, выведите любой из них.

Если требуемого результата добиться нельзя, выведите одно число 0.

Примеры входных и выходных файлов

Входные данные	Выходные данные
3 2 2 1 2 3 1 2 3 2	3 2 1
3 4 1 2 1 3 1 2 3 1 1 2 2 3 1 2 3 2	0

Решение.

Алгоритм:

Переберём все перестановки чисел от 1 до K — все возможные исходы бегов (порядок прихода тараканов). Для каждой перестановки за $O(N)$ проверим, правда ли, что у каждого болельщика в таком случае сыграет ровно одна ставка. Если это действительно так, выводим текущую перестановку и завершаем работу программы. В конце программы выводим 0 (программа не завершилась раньше, следовательно, ответ не найден).

Программа:

```
type
```

```
  Tstavka = record  
    a1, a2, b1, b2: longint;  
  end;
```

```
var
```

```
  i, n, k, e, s, p, w: int64;  
  a: array [1..10] of int64;  
  st: array [1..100] of Tstavka;  
  go: boolean;
```

```
procedure change(x: longint);
```

```
var
```

```
  il, t: longint;
```

```
begin
```

```
  for il := x + 1 to x + ((k - x) div 2) do
```

```
  begin
```

```
    t := a[il];
```

```
    a[il] := a[k + 1 - il + x];
```

```
    a[k + 1 - il + x] := t;
```

```
  end;
```

```
end;
```

```
procedure next;
```

```
var
```

```
  need, ch, t, il: longint;
```

```
begin
```

```
  need := 0;
```

```
  for il := k - 1 downto 1 do
```

```
    if a[il] < a[il + 1] then
```

```
    begin
```

```
      need := il;
```

```
      break;
```

```
    end;
```

```
  if need = 0 then
```

```
    exit;
```

```
  for il := k downto 1 do
```

```
    if a[il] > a[need] then
```

```
    begin
```

```
      ch := il;
```

```
      break;
```

```
    end;
```

```
  t := a[need];
```

```
  a[need] := a[ch];
```

```

a[ch] := t;
if e <> 1 then
  change(need);
end;

begin
// assign(input,'input.txt'); reset(input);
read(k, n);
for i := 1 to n do
begin
  read(st[i].a1, st[i].a2, st[i].b1);
  readln(st[i].b2);
end;
if (k=1) and (st[i].a1=1) and (st[i].a2=1) and (st[i].b1=1) and (st[i].b2=1) then
begin
writeln(0);
exit;
end;
for i := 1 to k do a[i] := i; s := 1;
for i := 1 to k do s := s * i;
for e := 1 to s do
begin
  go := true;
  for w := 1 to n do
    if ((a[st[w].a1] < a[st[w].a2]) and (a[st[w].b1] < a[st[w].b2])) or ((a[st[w].a1] > a[st[w].a2])
and (a[st[w].b1] > a[st[w].b2])) then
      begin
        go := false;
        break;
      end;
  if go = true then
    begin
      p := 1;
      while p < k do
        begin
          for i := 1 to k do
            if a[i] = p then
              begin
                write(i, ' ');
                inc(p);
                break;
              end;
          end;
        end;
      for i := 1 to k do
        if a[i] = k then
          begin
            writeln(i);
            exit;
          end;
        end;
      end;
    end;
  next;
end;
end;

```

```
writeln(0);
//close(input);
end.
```

Задача В. Гонки на машинах

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как и у каждого мальчика, у Феде есть игрушечные машинки. Однако ему повезло больше, чем обычному мальчику — все n его машинок являются радиоуправляемыми. Целыми днями он может устраивать различные автогонки и играть с друзьями.

Из всех видов гонок Федя предпочитает гонки по прямой. В данном формате соревнования трасса имеет форму прямой и является бесконечной (соревнования идут до тех пор, пока Феде это не надоест). Изначально каждая из n машинок находится на некотором расстоянии от старта — имеет фору x_i метров. По команде все машинки начинают свое движение от старта, при этом каждая машинка движется во время гонки с постоянной скоростью v_i метров в секунду. Все машинки движутся в одном направлении — удаляются от старта.

Недавно Феде подарили видеокамеру, и он хочет заснять яркие моменты гонки. Прежде всего Федя хочет запечатлеть первый обгон гонки, то есть первый момент времени, в который две машины находятся на одном расстоянии от старта.

Так как этого события можно ждать очень долго, Федя хочет настроить камеру на автоматическое включение во время обгона. Однако, Федя самостоятельно не может найти время, которое пройдет со времени начала гонки до времени первого обгона. Помогите Феде — напишите программу, находящую искомую величину.

Формат входных данных

В первой строке входного файла содержится единственное число n — количество машинок на трассе ($2 \leq n \leq 100$). Каждая из следующих n строк содержит по два целых числа x_i и v_i — расстояние от старта (в метрах) и скорость машинки i (в метрах в секунду) соответственно ($1 \leq x_i \cdot v_i \leq 1000$).

Исходно никакие две машинки не находятся в одной точке. Гарантируется, что хотя бы один обгон во время гонки произойдет.

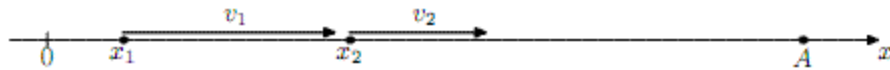
Формат выходных данных

В выходной файл выведите количество секунд, которое пройдет с момента старта до момента первого обгона, с точностью не менее 5 знаков после десятичной точки.

Примеры

Входные данные	Выходные данные
2 1 3 4 2	3.00000
2 12 20 2 21	10.00000

Пояснение для первого примера:



На рисунке точкой А обозначено место обгона.

Решение.

Алгоритм:

- 1) Машинка а когда-нибудь догонит машинку б, если её скорость больше, а координата меньше
- 2) Переберём каждую машинку с каждой и если одна машинка когда-нибудь обгонит другую (пункт 1), то посчитаем это время. Среди всех таких значений выберем минимальное. Это и будет ответом

Программа:

```

program gt;
var a,b:array[1..100000]of longint;
c:array[1..100000]of real;
j,n,i,k,l:longint;
min:real;
begin
read(n);k:=0;
for i:=1 to n do
begin
read(a[i],b[i]);
end;
for i:=1 to n do
begin
for j:=1 to n do
begin
if(i<>j)and(a[i]<a[j])and(b[i]>b[j])then begin
k:=k+1;
c[k]:=(a[j]-a[i])/(b[i]-b[j]);
end;
end;
end;
min:=c[1];
for i:=2 to k do
begin
if(c[i]<min)then min:=c[i];
end;
writeln(min:1:5);
end.

```

Задача С. Тестирование

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Одним из способов контроля знаний учащихся является проведение тестирований. На тестировании обычно предлагается несколько вопросов, для каждого из которых необходимо выбрать один из вариантов ответа. Для этого тестируемому выдается специальный бланк с вопросами и вариантами ответов.

Тестирования можно проводить одновременно для достаточно большого количества людей, поэтому возникает вопрос об эффективной обработке заполненных тестируемыми бланков. Во Флатландии эту проблему пытаются решить с помощью применения информационных технологий для автоматической обработки результатов тестирования.

Первым этапом является написание программы, которая бы вычисляла баллы за тест по известной информации о правильных ответах на вопросы и ответах, данных тестируемым. Ваша задача состоит в написании такой программы.

Формат входного файла

Первая строка входного файла содержит число n ($1 \leq n \leq 100000$) вопросов в тесте. Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — номера правильных вариантов ответов на каждый из вопросов. Третья строка входного файла содержит n целых чисел b_1, b_2, \dots, b_n — номера вариантов, выбранных тестируемым. Для чисел a_i и b_i верны неравенства $1 \leq a_i, b_i \leq 10$.

Формат выходного файла

В выходной файл выведите число вопросов, на которые тестируемый дал правильный ответ.

Примеры входных и выходных файлов

Входные данные	Выходные данные
4 1 2 3 4 1 2 4 3	2
4 1 2 3 4 4 3 2 1	0

Решение.

Алгоритм:

Очень легкая задача. Считываем данные в два массива. Затем запускаем цикл до n и проверяем: если i -й элемент 1-го массива равен i -му элементу 2-го массива, то мы увеличиваем счетчик на 1.

Программа:

```
var
  a,b:array[1..100000] of byte;
  i,n, p:longint;
```

BEGIN

```
  assign(input, 'input.txt');
  reset(input);
  assign(output, 'output.txt');
  rewrite(output);
  readln(n);
  for i:=1 to n do
    read(a[i]);
  readln;
  p:=0;
  for i:=1 to n do begin
    read(b[i]);
    if a[i]=b[i] then
      inc(p);
  end;
  writeln(p);
end.
```

Задача D. Соревнования картингистов

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

После очередного этапа чемпионата мира по кольцевым автогонкам на автомобилях с открытыми колёсами Формула-А гонщики собрались вместе в кафе, чтобы обсудить полученные результаты. Они вспомнили, что в молодости соревновались не на больших болидах, а на картах — спортивных автомобилях меньших размеров.

Друзья решили выяснить победителя в одной из гонок на картах. Победителем гонки являлся тот гонщик, у которого суммарное время прохождения всех кругов трассы было минимальным.

Поскольку окончательные результаты не сохранились, то каждый из n участников той гонки вспомнил и выписал результаты прохождения каждого из m кругов трассы. К сожалению, по этой информации гонщикам было сложно вычислить победителя той гонки. В связи с этим они попросили сделать это вас.

Требуется написать программу, которая вычислит победителя гонки на картах, о которой говорили гонщики.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 100$). Последующие $2 \cdot n$ строк описывают прохождение трассы каждым из участников. Описание прохождения трассы участником состоит из двух строк. Первая строка содержит имя участника с использованием только латинских букв (строчных и заглавных). Имена всех участников различны, строчные и заглавные буквы в именах различаются.

Вторая строка содержит m положительных целых чисел, где каждое число — это время прохождения данным участником каждого из m кругов трассы (каждое из этих чисел не превосходит 1000). Длина имени каждого участника не превышает 255.

Формат выходного файла

В выходной файл необходимо вывести имя победителя гонки на картах. Если победителей несколько, требуется вывести имя любого из них.

Примеры входных и выходных файлов

Входные данные	Выходные данные
5 3 Sumaher 2 1 1 Barikelo 2 1 2 Olonso 1 2 1 Vasya 1 1 1 Fedya 1 1 1	Fedya

Обращаем внимание на пробел в конце последней строки входных данных.

Решение.

Алгоритм:

Заведем отдельную строку для хранения фамилии игрока с лучшим результатом, и отдельную переменную для хранения этого результата. Затем присваиваем им результаты и фамилию 1-го гонщика, а затем считываем оставшиеся результаты и сравниваем с максимумом на данном шаге и, если требуется изменяем последний.

Программа:

```
var n,m,i,j,x,sum,min:longint;  
    s,s2:string;  
begin  
    min:=maxlongint;  
    readln(n,m);  
    for i:=1 to n do  
    begin  
        readln(s);  
        sum:=0;  
        for j:=1 to m do  
        begin  
            read(x);  
            sum:=sum+x;  
        end;  
        readln;  
        if (sum<min) then  
        begin
```



```

    min:=sum;
    s2:=s;
end;
end;
writeln(s2);
end.

```

Задача Е. Дипломы

Имя входного файла: input.txt
 Имя выходного файла: output.txt
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 64 мегабайта

Когда Петя учился в школе, он часто участвовал в олимпиадах по информатике, математике и физике. Так как он был достаточно способным мальчиком и усердно учился, то на многих из этих олимпиад он получал дипломы. К окончанию школы у него накопилось n дипломов, причём, как оказалось, все они имели одинаковые размеры: w — в ширину и h — в высоту.

Сейчас Петя учится в одном из лучших российских университетов и живёт в общежитии со своими одногруппниками. Он решил украсить свою комнату, повесив на одну из стен свои дипломы за школьные олимпиады. Так как к бетонной стене прикрепить дипломы достаточно трудно, то он решил купить специальную доску из пробкового дерева, чтобы прикрепить её к стене, а к ней — дипломы. Для того чтобы эта конструкция выглядела более красиво, Петя хочет, чтобы доска была квадратной и занимала как можно меньше места на стене. Каждый диплом должен быть размещён строго в прямоугольнике размером w на h . Дипломы запрещается поворачивать на 90 градусов. Прямоугольники, соответствующие различным дипломам, не должны иметь общих внутренних точек.

Требуется написать программу, которая вычислит минимальный размер стороны доски, которая потребуется Пете для размещения всех своих дипломов.

Формат входного файла

Входной файл содержит три целых числа: w, h, n ($1 \leq w \cdot h \cdot n \leq 10^9$).

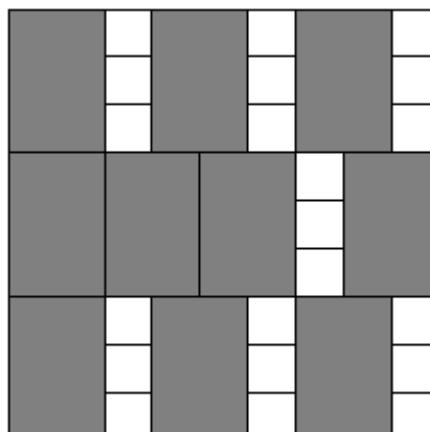
Формат выходного файла

В выходной файл необходимо вывести ответ на поставленную задачу.

Примеры входных и выходных файлов

Входные данные	Выходные данные
2 3 10	9

Иллюстрация к примеру:



Решение.

Алгоритм:

Из-за больших ограничений на n, w, h линейный перебор возможной длины стороны квадрата не проходит по времени, поэтому данную задачу следует решать, используя бинарный поиск по ответу. Очевидно, что размеры доски лежат в пределах от $\min(w, h)$ до $n * \max(w, h)$. За $O(1)$ легко проверить, поместятся ли все грамоты в квадрат со стороной a ($n \leq (a / w) * (a / h)$). Пусть мы уверены, что искомым ответом лежит в интервале от Min до Max , тогда проверим, удовлетворяет ли условию квадрат со стороной $\text{Mid} = (\text{Min} + \text{Max}) / 2$. Если да, то $\text{Max} = \text{Mid}$, иначе $\text{Min} = \text{Mid} + 1$. Выполняя эту процедуру до тех пор, пока не сойдутся Min и Max , мы будем всё точнее получать возможный диапазон ответа (каждый раз область поиска уменьшается в 2 раза). Когда Min станет равно Max , выведем значение Min и прекратим работу программы. Сложность такого решения: $O(\log(n * \max(w, h)))$.

Программа:

```
function maxwh (w2, h2 :int64) :int64;
begin
if w2>h2 then maxwh:=w2 else maxwh:=h2;
end;
function maxd (w1, h1, mid1 :int64) :int64;
begin
maxd:=(mid1 div w1)*(mid1 div h1);
end;
var
w, h, n: int64;
hi, lo, mid :int64;
begin
readln(w, h, n);
hi:=maxwh(w, h)*n;
lo:=0;
while hi-lo>1 do
begin
mid:=(hi+lo) div 2;
if maxd(w, h, mid)<n then lo:=mid else hi:=mid;
end;
writeln(hi);
end
```