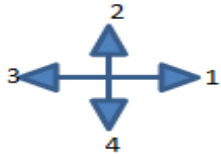


PHẦN 1: ĐỒ THỊ

Bài 1: GAMESHOW

Mã bài: ARROWS.???

Cho một lưới ô vuông kích thước $M \times N$, mỗi ô của bảng chứa một trong 4 giá trị trong tập $\{1, 2, 3, 4\}$. Một gameshow được tổ chức với luật chơi, nếu bạn đang đứng ở ô (i, j) thì bước tiếp theo bạn phải di chuyển sang ô chung cạnh theo hướng tương ứng với giá trị của ô đó. Gọi $P(x, y)$ là số ô khác nhau mà bạn đi qua trước khi ra khỏi bảng hoặc vào ô mà bạn đã vào trước đó nếu xuất phát từ ô (x, y) . Định nghĩa ô thuận lợi là ô mà bạn không ra được khỏi bảng khi bạn xuất phát từ ô đó. Giá trị của ô và hướng tương ứng như hình vẽ bên.



Yêu cầu:

1. Cho tọa độ ô (x, y) mà bạn đang đứng, hãy tính số điểm mà bạn đạt được. Nếu ô (x, y) là ô thuận lợi thì điểm của bạn là $P(x, y) \cdot 1000$ ngược lại điểm bạn là $P(x, y)$.
2. Hãy đếm số ô thuận lợi có trên bảng.
3. Hãy tìm điểm lớn nhất mà có thể đạt được.

Dữ liệu:

- Dòng đầu hai số m, n ($m, n \leq 3000$).
- Chứa x, y là tọa độ của ô xuất phát.
- M dòng tiếp theo, mỗi dòng gồm N số thuộc tập $\{1, 2, 3, 4\}$.

Kết quả:

- Gồm 3 dòng, mỗi dòng gồm một số nguyên duy nhất là kết quả của câu hỏi tương ứng.

ARROWS.INP	ARROWS.OUT
5 3	6000
4 1	9
2 3 3	7000
1 1 4	
1 1 4	

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

2 3 3	
4 4 3	

Solution:

Từ bảng ô vuông $M \times N$ ban đầu, coi mỗi ô của bảng là một đỉnh của đồ thị vô hướng $G(V, E)$.

Ô (i, j) sẽ tương ứng với đỉnh $(i-1) \cdot N + j$ (ta gọi nó là Id của ô (i, j)) của G .

↑	←	←
→	→	↓
→	→	↓
↑	←	←
↓	↓	←

Xây dựng thêm đỉnh giả $t = m \cdot n + 1$

Nếu từ $(i, j) \rightarrow (u, v)$ nằm trong bảng thì cộng thêm cung thuận $Add(id(i, j), id(u, v), true)$, cung ngược $Add(id(u, v), id(i, j), false)$.

Nếu từ $(i, j) \rightarrow (u, v)$ không nằm trong bảng thì cộng thêm cung thuận $Add(id(i, j), t, true)$, cung ngược $Add(t, id(i, j), false)$.

Yêu cầu 1: Gọi s là đỉnh xuất phát, và sls là số lượng đỉnh mà s đến được.

Từ s ta Dfs(s) theo các cung thuận (tức $e[i].t = true$) đến được đỉnh nào thì ta tăng sls thêm 1. Nếu s đến được t thì s không là đỉnh thuận lợi, nếu không thì s là đỉnh thuận lợi.

S là đỉnh thuận lợi thì $kq1 = sls \cdot 1000$ nếu không $kq1 = sls - 1$;

Yêu cầu 2: Từ t ta Dfs(t) theo các cung ngược (tức $e[i].t = false$). Gọi slt là số lượng mà t đến được.

Dễ thấy các đỉnh mà t đến được sẽ không là đỉnh thuận lợi, do đó số đỉnh thuận lợi là $kq2 = t - slt$;

Yêu cầu 3: Dùng thuật toán Tarjan tìm các thành phần liên thông mạnh. Coi mỗi thành phần liên thông mạnh là một đỉnh của đồ thị mới $G'(V', E')$. Ta thấy G' là một đồ thị mà mỗi thành phần liên thông của nó là một đồ thị vô hướng không chu trình.

Từ **Yêu cầu 2** đánh dấu các đỉnh trên G sao cho $cx[u] = true$ nếu u là thuận lợi, $= false$ nếu u là không thuận lợi.

Nếu u' là một đỉnh trong G' chứa u của G ta có $tl[u'] = cx[u]$.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Gọi $l[u]$ là số lượng đỉnh trong thành phần liên thông mạnh thứ u , nn là số lượng đỉnh của đồ thị G' .

Xây đỉnh thêm đỉnh giả $s=nn+1$ nối tới tất cả các đỉnh trong G' .

Dùng thuật toán sắp xếp Topo để đánh số lại đồ thị G' . Gọi $Num[u]=i$ là số thứ tự của đỉnh u sau khi sắp xếp Topo khi đó nếu tồn tại cung (u,v) thì $Num[u]<Num[v]$, $Q[i]=u$.

Gọi $f[v]$ là số lượng đỉnh đã đi qua khi đến v .

Ta có công thức. Nếu tồn tại cung (u,v) thì $f[v]:=Max(f[v],f[u]+l[v])$.

Code:

```
for j:=1 to nn+1 do
begin
    u:=Q[j];
    for v kề u f[v]:=Max(f[v],f[u]+l[v]).
end;
```

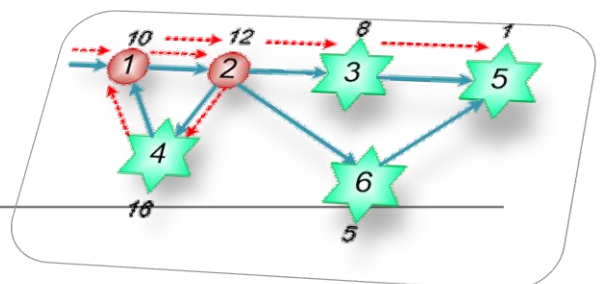
+Nếu $tl[u]=true$ thì $val:=f[u]*1000$ ngược lại $val:=f[u]-1$. Cập nhật $kq3:=Max(kq3,val)$ cho mỗi u .

Bài 2: TRỘM ATM

Mã bài: ATM.???

Tất cả các đường trong thành phố của Siruseri đều là một chiều. Theo luật của quốc gia này, tại mỗi giao lộ phải có một máy ATM. Điều đáng ngạc nhiên là các cửa hàng chơi điện tử cũng nằm ở các giao lộ, tuy nhiên, không phải tại giao lộ nào cũng có cửa hàng chơi điện tử. Banditji là một tên trộm nổi tiếng. Hắn quyết định làm một vụ động trời: khoắng sạch tiền trong các máy ATM trên đường đi, sau đó ghé vào một cửa hàng chơi điện tử để thư giãn. Nhờ có mạng lưới thông tin rộng rãi, Banditji biết được số tiền có ở mỗi máy ATM ngày hôm đó. Xuất phát từ trung tâm, tên trộm lái xe đi dọc theo các phố, vét sạch tiền ở các ATM gặp trên đường đi. Banditji có thể đi lại nhiều lần trên một số đoạn phố, nhưng sẽ không thu gì được thêm từ các ATM đã bị khoắng trước đó. Lộ trình của Banditji phải kết thúc ở giao lộ có cửa hàng chơi điện tử. Banditji biết cách vạch lộ trình để tổng số tiền trộm được là lớn nhất.

Ở ví dụ nêu trên hình vẽ, thành phố có 6 giao lộ đánh số từ 0 đến 6, số tiền trong ATM được ghi ở



CHUYÊN ĐỀ TỔNG HỢP 2013-2014

bên cạnh nút. Các nút hình sao chỉ giao lộ có cửa hàng chơi điện tử. Đường chấm chấm cho biết lộ trình của tên trộm. Tổng cộng, hắn đã lấy được 47 đồng.

Yêu cầu: Cho biết n – số giao lộ, m – số đoạn đường nối 2 giao lộ, p – số giao lộ có cửa hàng chơi điện tử và các nơi có cửa hàng, a_i – số tiền trong ATM đặt ở giao lộ i , s – giao lộ trung tâm. Hãy xác định tổng số lượng tiền bị trộm ($n, m \leq 500\,000, 0 \leq a_i \leq 4\,000$).

Input:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Mỗi dòng trong m dòng tiếp theo chứa 2 số nguyên u và v xác định đường đi từ giao lộ u tới giao lộ v ,
- Dòng thứ i trong n dòng tiếp theo chứa số nguyên a_i ,
- Dòng thứ $n+m+2$ chứa 2 số nguyên s và p ,
- Dòng cuối cùng chứa p số nguyên xác định các giao lộ có cửa hàng chơi điện tử.

Output: Một số nguyên – số tiền bị trộm.

Example:

ATM.INP	ATM.OUT
6 7	47
0 2	
2 3	
3 5	
2 4	
4 1	
2 6	
6 5	
10	
12	
8	
16	
1	
5	
0 4	
4 3 5 6	

Solution:

Dùng thuật toán Tarjan tìm các thành phần liên thông mạnh. Coi mỗi thành phần liên thông mạnh là một đỉnh của đồ thị mới $G'(V', E')$. Ta thấy G' là một đồ thị mà mỗi thành phần liên thông của nó là một đồ thị vô hướng không chu trình.

Dùng thuật toán sắp xếp Topo để đánh số lại đồ thị G' . Gọi $\text{Num}[u]=i$ là số thứ tự của đỉnh u sau khi sắp xếp Topo khi đó nếu tồn tại cung (u, v) thì $\text{Num}[u] < \text{Num}[v]$, $Q[i]=u$.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Gọi $f[u]$ là tổng tiền lớn nhất khi đến đỉnh u . $Kt[u]=0/1$ nếu u là đỉnh không/có chứa quán game.

$Sum[u]$ là tổng số tiền có ở đỉnh u , x_p là đỉnh chứa đỉnh s

Quy hoạch động trên G' :

for $u:=1$ to n do

if $u=x_p$ then $f[u]=sum[u]$ else

for v kề u $f[v]=Max(f[v], f[u]+sum[v])$.

$Kq=max(f[u])$ mà $kt[u]=1$.

Bài 3: THU HOẠCH

Mã bài: KCOLLECT.???

Công việc buôn bán dừa của Pirate không mấy khả quan cho lắm, khiến anh đêm ăn không ngon ngày ngủ không yên, chỉ biết chúm đầu vào xem "Rôbô trái cây". Một ngày nọ, đang nằm ngủ dưới gốc dừa, bỗng một trái dừa rơi vào đầu anh ấy. Cũng giống như Newton, Pirate cũng cầm trái dừa lên, ngắm nghía và... rủa: "Khỉ thật, sao xứ này toàn là dừa thế này!". Tức điên lên, Pirate quyết trồng thêm các loại trái cây khác vào hòn đảo của mình.

Đến mùa thu hoạch, Pirate đặt hàng một "Rôbô trái cây" để giúp mình hái quả. Khu vườn của Pirate có hình chữ nhật, và được chia thành $M \times N$ ô vuông bằng nhau. Trong mỗi ô vuông có một cây thuộc một loại quả khác nhau, đánh số từ 0 đến 9. Không phải vô tình mà chúng được đánh số như vậy, con số đó thể hiện giá trị kinh tế của các loại cây.

Tuy nhiên, nhìn mặt con Rôbô trái cây này có vẻ ngu ngu nên trong lần đầu tiên thử việc, Pirate muốn test AI của nó. Cụ thể là Rôbô phải tuân theo các quy định sau:

- Tại mỗi ô, Rôbô chỉ có thể đi sang hướng đông hoặc hướng nam sang ô kề cạnh.
- Có một số ô đặc biệt mà tại đó Rôbô có thể đi được thêm hướng tây hoặc hướng bắc sang ô kề cạnh (chỉ một trong hai).
- Rôbô không được đi vào những ô có cây dừa (Pirate cấm thù dừa).
- Rôbô được đi qua một ô nhiều lần. Khi đi qua một ô, Rôbô phải hái hết quả ở cây trong ô đó. Lợi nhuận thu được sẽ bằng chỉ số của loại cây vừa được thu hái. Và sau này, không thể đạt thêm lợi nhuận gì từ ô đó nữa.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Xuất phát từ ô ở góc tây bắc của khu vườn, hãy giúp Rôbô trái cây xác định hành trình để đạt được lợi nhuận tối đa.



Input

- Dòng thứ nhất: ghi hai số nguyên M và N - kích thước của khu vườn.
- M dòng tiếp theo: mỗi dòng ghi N kí tự liên tiếp nhau mô tả khu vườn:
 - + '0' - '9': các loại trái cây;
 - + '#': cây dừa;
 - + 'W': được quyền đi theo hướng tây;
 - + 'N': được quyền đi theo hướng bắc.

Output

- Ghi một số nguyên duy nhất là lợi nhuận tối đa đạt được.

Giới hạn

- Trong mọi test, $1 \leq M, N \leq 100$.
- 60% số test có $1 \leq M, N \leq 20$.

Example

KCOLLECT.INP	KCOLLECT.OUT
2 3 264 3WW	15

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Giải thích: Rô-bô sẽ đi theo hành trình như sau $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (2, 2) \rightarrow (2, 1)$ (ô (i, j) là ô ở dòng i và cột j). Tổng lợi nhuận sẽ là $2 + 6 + 4 + 3 = 15$.

Solution:

Thuật toán tương tự như 2 bài trên. Ta xây dựng đồ thị rồi QHĐ.

Bài 4: CHÈO THUYỀN

Mã bài: ROWING.???

Người dân nước GeoLand say mê các môn thể thao mạo hiểm đòi hỏi tư duy hình học chuyên nghiệp. Một trong những môn thể thao đó là bơi thuyền vượt bãi đá trên sông Rect River – con sông dài nhất GeoLand. Bản đồ con sông được vẽ trên mặt phẳng tọa độ với hệ tọa độ Descartes vuông góc, hai bờ sông là hai đường thẳng song song $y = 0$ và $y = h$. Bãi đá trên sông gồm n tảng đá đánh số từ 1 tới n , tảng đá thứ i có tọa độ (x_i, y_i) trên bản đồ. Mỗi vận động viên tham gia bài thi với một thuyền thúng hình tròn. Anh ta được đặt thuyền của mình ở vị trí tùy chọn nằm hoàn toàn bên trái bãi đá và cần bơi thuyền tới một vị trí tùy chọn nằm hoàn toàn bên phải bãi đá. Thuyền được di chuyển theo hướng tùy ý nhưng không được chạm vào bờ sông hay chạm vào một tảng đá nào của bãi đá (kể cả đường biên của thuyền).

Yêu cầu: Tìm số nguyên d lớn nhất để mọi thuyền có đường kính $< d$ đều có thể thực hiện được bài thi.

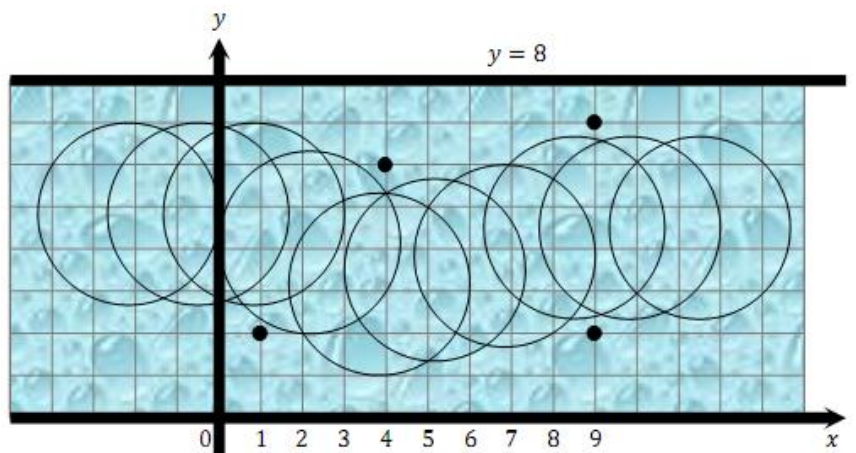
Dữ liệu: Vào từ file văn bản ROWING.INP

- Dòng 1 chứa hai số nguyên dương n, h ($n \leq 4000; 2 \leq h \leq 109$)
- Dòng tiếp theo, mỗi dòng chứa hai số nguyên dương $x_i \leq 10^9, y_i < h$.

Kết quả: Ghi ra file văn bản ROWING.OUT một số nguyên duy nhất là số d tìm được.

Ví dụ

ROWING.INP	ROWING.OUT



CHUYÊN ĐỀ TỔNG HỢP 2013-2014

48	5
12	
46	
92	
97	

Solution:

Cách 1: Đối ngẫu:

Chặt nhị phân đường kính con thuyền.

Xây dựng hàm kiểm tra chiếc thuyền đường kính K có đi qua được sông hay không.

- Kéo bờ trên xuống đoạn $k/2$, bờ dưới lên đoạn $k/2$; Mỗi hòn đá to lên $k/2$.
- Bây giờ chiếc thuyền trở thành 1 điểm.
- Kiểm tra điểm có đi qua được sông không bằng Dfs/Bfs.
- Độ phức tạp ($O(n^2, \log_2 H)$)

Cách 2: KrusKal:

Ta thấy có $\begin{cases} n + 2 \text{ chương ngại vật.} \\ \frac{(n+2)*(n+1)}{2} \text{ khoảng cách giữa 2 chương ngại vật.} \end{cases}$

- Sort lại khoảng cách.
- Kruskal.
- Làm cho đến khi Findset(bờ trên)=Findset(bờ dưới).
- Độ phức tạp ($O(n^2, \log_2 n)$)

Cách 3: Distr:

Tim đường đi từ bờ trên xuống bờ dưới mà có cạnh lớn nhất trên đường đi là nhỏ nhất. Đó là kết quả của bài toán.

- Độ phức tạp ($O(n^2)$)

Cách 4: Euclidean Minimum Spanning Tree:

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Tương tự như cách 3 nhưng ta tìm đường đi bằng cây khung với độ phức tạp ($O(n \cdot \log_2 n)$). Do thuật toán khá phức tạp nên ta sẽ chọn thuật toán thứ 2 hoặc 3 là đủ full test với yêu cầu của bài toán.

Bài 5: DI CHUYỂN ROBOT

Mã bài: ROBOT.???

Cuộc thi ROBOCON năm 2013 có chủ đề “Gặp gỡ”. Các robot sẽ thi đấu trên một bản đồ gồm n điểm dừng được đánh số từ 1 đến n và m đoạn đường một chiều nối các cặp điểm dừng. Mỗi điểm dừng được tô bằng một trong ba màu 1, 2, 3. Mỗi đoạn đường cũng được tô bằng một trong ba màu 1, 2, 3 nhưng sẽ bị biến đổi màu theo thời gian. Cụ thể, nếu ở thời điểm t , đoạn đường có màu là c thì ở thời điểm $t + 1$ đoạn đường sẽ có màu là $(c \bmod 3) + 1$. Mỗi đội chơi phải điều khiển hai robot cùng di chuyển trên bản đồ để hai robot gặp nhau sớm nhất tại một điểm dừng. Ban đầu (tại thời điểm 0), robot thứ nhất đứng ở điểm dừng 1, robot thứ hai đứng ở điểm dừng 2, quy tắc di chuyển của hai robot là:

- Mỗi đơn vị thời gian, cả hai robot cùng phải di chuyển qua một đoạn đường;
- Giả sử ở thời điểm t robot thứ nhất đang đứng ở điểm dừng i , robot thứ hai đang đứng ở điểm dừng j , khi đó robot thứ nhất buộc phải di chuyển theo một trong các đoạn đường có màu ở thời điểm t cùng màu với màu ở điểm dừng j , robot thứ hai buộc phải di chuyển theo một trong các đoạn đường có màu ở thời điểm t cùng màu với màu ở điểm dừng i .

Yêu cầu: Cho thông tin về bản đồ với màu ban đầu của các đoạn đường, hãy xác định thời điểm hai robot gặp nhau sớm nhất.

Input

Dòng đầu tiên chứa Q là số lượng bộ dữ liệu. Tiếp đến là Q nhóm dòng, mỗi nhóm cho thông tin về một bộ dữ liệu theo khuôn dạng sau đây:

- Dòng thứ nhất chứa hai số nguyên dương n, m ;
- Dòng thứ hai ghi n số nguyên, số thứ i là màu của điểm dừng thứ i ;
- m dòng sau, mỗi dòng mô tả một đoạn đường gồm ba số nguyên u, v ($u \neq v$) và c chỉ đoạn đường từ điểm dừng u đến điểm dừng v có màu tại thời điểm 0 là c .

Output

Gồm Q dòng, mỗi dòng chứa một số nguyên là thời điểm gặp nhau sớm nhất của hai robot (ghi -1 nếu không tồn tại cách di chuyển để hai robot gặp nhau) tương ứng với bộ dữ liệu vào.

ROBOT.INP	ROBOT.OUT
1	2

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

3 4 1 2 3 1 2 2 2 1 2 2 3 1 3 1 1	
--	--

Subtask 1: $2 \leq n \leq 10$;

Subtask 2: $2 \leq n \leq 50$;

Subtask 3: $2 \leq n \leq 200$.

Solution:

Bài này đơn giản là dùng thuật toán BFS với đỉnh là trạng thái $[i,j,c]$ tức tại ô $[i,j]$ có đang có màu c .

Note là màu ô (i,j) ở thời gian t hơi đặc biệt một chút.

Thủ tục đẩy đỉnh v kề u vào Queue.

```
66 Procedure Next(u,v,t:longint;var a:Arr;var na:longint);
67 var i:longint;
68     vx:longint;
69 begin
70     i:=head[u];
71     While i<>0 do
72     begin
73         vx:=e[i].y;
74         if (e[i].w+t-1) mod 3+1=mau[v] then
75         begin
76             inc(na);
77             a[na]:=vx;
78         end;
79         i:=e[i].link;
80     end;
81 end;
```

Bài 6: TRÒ CHƠI

Mã bài: GAME.???

Jian-Jia là cậu bé yêu thích các trò chơi. Khi được yêu cầu trả lời một câu hỏi, cậu thích chơi một trò chơi hơn thay vì trả lời trực tiếp câu hỏi đó. Jian-Jia gặp Mei-Yu và kể cho cô bạn về mạng lưới chuyển bay ở Đài Loan. Có n thành phố ở Đài Loan (được đánh số từ 1 đến n), một số thành phố được nối với nhau bởi các chuyển bay. Mỗi chuyển bay nối hai thành phố và có thể đi theo cả hai chiều. Mei-Yu hỏi Jian-Jia liệu có thể di chuyển giữa hai thành phố bất kỳ bằng máy bay (hoặc là trực tiếp hoặc gián tiếp) hay không? Jian-Jia không muốn

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

tiết lộ câu trả lời mà thay vì điều đó đề nghị chơi một trò chơi. Mei-Yu có thể hỏi cậu các câu hỏi có dạng: "Giữa hai thành phố x và y có chuyến bay *trực tiếp* hay không?" và Jian-Jia sẽ trả lời các câu hỏi ngay lập tức. Mei-Yu sẽ hỏi tất cả $\frac{n(n-1)}{2}$ câu hỏi, Mei-Yu sẽ giành phần thắng nếu sau khi nhận được câu trả lời cho k câu hỏi đầu tiên ($k < \frac{n(n-1)}{2}$), cô có thể khẳng định được mạng có liên thông hay không, mạng liên thông là có thể di chuyển giữa hai thành phố bất kỳ bằng các chuyến bay (hoặc là trực tiếp hoặc gián tiếp). Ngược lại, nếu cô cần hỏi tất cả $\frac{n(n-1)}{2}$ câu hỏi thì người giành phần thắng là Jian-Jia.

Để trò chơi được thú vị hơn, hai bạn thỏa thuận với nhau rằng Jian-Jia có thể không sử dụng mạng lưới chuyến bay thực tế của Đài Loan, thay vào đó mạng sẽ tạo ra theo ý của Jian-Jia.

Yêu cầu: Cho n và thứ tự $\frac{n(n-1)}{2}$ câu hỏi của Mei-Yu, bạn hãy giúp Jian-Jia giành phần thắng của trò chơi bằng cách đưa ra những trả lời cho các câu hỏi của Mei-Yu.

Input

Dòng đầu tiên ghi số nguyên dương K ($K \leq 10$) là số lượng bộ dữ liệu. Tiếp đến là K nhóm dòng, mỗi nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng đầu chứa số nguyên n .
- Dòng thứ k trong $\frac{n(n-1)}{2}$ dòng sau chứa hai số nguyên dương i, j mô tả câu hỏi thứ k của Mei-Yu.

Output

Gồm K dòng (mỗi dòng tương ứng với bộ dữ liệu vào), dòng chứa $\frac{n(n-1)}{2}$ số 0 hoặc 1, số thứ k bằng 0 nếu câu hỏi thứ k được trả lời là không có cạnh, bằng 1 trong trường hợp ngược lại.

GAME.INP	GAME.OUT
1 4 1 4 2 1 1 3 4 2 2 3 3 4	0 1 0 1 0 1

Subtask 1: $n \leq 5$

[10 tests]

Subtask 2: $n \leq 50$

[10 tests]

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Subtask 3: $n \leq 500$

[10 tests]

Solution:

Sub1: Ta duyệt nhị phân 0/1 kết hợp với disjoint-set và cập nhật kết quả với phương án tối ưu.

Sub 3:

Nhận xét:

- Ta thấy nếu câu hỏi (i, j) trả lời không có mà không ảnh hưởng tới tính liên thông của đồ thị thì ta trả lời là không, mất tính liên thông ta trả lời là có.
- Để Jian-Jia có thể thắng thì ta cố gắng kéo các câu trả lời có càng về sau càng tốt.
- Giải thuật disjoint-set: Đi từ cuối về đầu, nếu 2 đỉnh chưa thuộc cùng một thành phần liên thông ta trả lời có ngược lại trả lời không.

Mở rộng: Các bạn có thể cập nhật Online cho bài toán trên, tức là đưa ra câu hỏi nào chúng ta phải trả lời ngay cho câu hỏi đó.

Bài 7: ĐẾ CHẾ DỪA

Mã bài: VMAOCE.???

Vào năm 2410, sau khi Pirate vĩ đại của vương quốc Dừa đã không còn, cảnh chọi dừa đổ máu xảy ra như cơm bữa. Dân chúng ngày càng lầm than vì dừa, là lương thực chính của mọi người, đã bị mang đi chọi gần hết. Tuy nhiên, một số kẻ khôn ngoan đi nhặt dừa về tích trữ. Họ trở nên giàu có và hùng mạnh đến mức mỗi người chiếm lấy một hòn đảo để thành lập vương quốc riêng. Sử gọi đây là thời đại của các đế chế dừa (Age of Coconut Empires).

Quần đảo dừa gồm N hòn đảo. Vì xung quanh đảo (dĩ nhiên) toàn nước nên từ xa xưa Pirate đã cho xây các cầu dừa để nối các cặp hòn đảo lại với nhau. Nhắc đến đây phải kể đến sự thiên tài lỗi lạc của Pirate trong việc tính toán xây cầu: hệ thống cầu được thiết kế sao cho hai hòn đảo bất kì đến đi đến được nhau, nhưng nếu bất kì một cây cầu nào đó bị hỏng thì lập tức điều đó sẽ không được đảm bảo.

Các quốc gia sau nhiều năm chọi dừa đã trở nên kiệt quệ. Lý do là vì mỗi đảo cùng lúc phải chọi và hứng dừa của $N - 1$ đảo khác. Thời thế tạo anh hùng, Pirakute (cháu 10 đời của Pirate) du thuyết N đảo về kế sách "tam phân thiên hạ" với ước mong giúp giảm bớt nạn binh đao chọi dừa.

Pirakute giảng giải: "Tam phân thiên hạ thật chẳng có gì khó, chỉ cần tam phân một hòn đảo là đủ. Có nghĩa ta sẽ phân chia hòn đảo đó làm ba lãnh địa, là ranh giới của ba đế chế mới. *Không có ai thể đi lại tự do từ đế chế này sang đế chế khác.* Tiếp theo, các nước có cầu dừa nối đến hòn đảo bị phân chia phải lựa chọn đế chế nào để liên minh, rồi phá bỏ cây cầu cũ nối đến hòn đảo bị phân chia để xây dựng cây cầu mới nối đến lãnh địa của đế chế mình

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

trên hòn đảo này. Chuyện còn lại là tất yếu, các hòn đảo còn lại buộc lòng phải gia nhập vào đế chế mà mình có thể tự do đi đến được (tức là *không phải băng qua ranh giới* nằm trên hòn đảo bị phân chia). Tuy nhiên, hòn đảo bị phân chia sẽ bị tiêu diệt, sau này chỉ đóng vai trò làm ranh giới và *không thuộc vào vương quốc nào cả*."

"Chỉ có điều, phân chia hòn đảo nào, việc này tôi cũng không rõ. Nhưng ngu kiến của tôi là sau khi hoàn thành việc phân chia thiên hạ, mỗi đế chế mới sẽ gồm ít nhất một đảo và không quá một nửa số đảo trong quần đảo ($N/2$). Như vậy cục diện mới cân bằng, thiên hạ mới thái bình dài lâu."

Bạn hãy giúp Pirakute tìm một phương án để tam phân thiên hạ nhé.

Input

- Dòng thứ nhất chứa một số nguyên N , số đảo trong quần đảo.
- Các dòng tiếp theo, mỗi dòng chứa hai số nguyên, mô tả một cặp đảo được nối bởi một cây cầu dừa. Các đảo được đánh số từ 1 đến N .

Output

- Gồm 1 dòng chứa N số, số thứ i là 0 nếu i là hòn đảo bị phân chia, ngược lại là 1, 2 hoặc 3 tương ứng với vương quốc của hòn đảo đó. Nếu không thể tam phân thiên hạ, xuất ra duy nhất một số -1.

Giới hạn

- $1 \leq N \leq 100000$.
- 30% số test có $1 \leq N \leq 1000$.
- Hệ thống đường mô tả trong input đảo bảo tính chất được nêu ra ở đề bài.

Example

Input	Output
6 1 2 1 3 3 4 3 5 5 6	1 1 0 2 3 3

Solution:

Ta sẽ tìm một đỉnh mà sau khi bỏ nó ra khỏi đồ thị, số lượng thành phần liên thông (tplt) còn lại ≥ 3 và các tplt này có trọng lượng $\leq N/2$. Đỉnh đó chính là đỉnh cần tìm của bài toán.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Chứng minh:

- Giả sử trọng lượng của các thành phần khi tách ra là :
 - $d_1, d_2, d_3 \dots d_m$. (tất cả đều $\leq N/2$)
- Ta có 3 tập rỗng, ta sẽ cố gắng xếp các tplt trên vào 3 tập sao cho trọng lượng mỗi tập $\leq N/2$.
- Ta bỏ lần lượt TỐI ĐA có thể vào tập 1 và được trọng lượng c_1 ($c_1 \leq N/2$).
- Nếu thêm một trọng lượng c nữa thì có : $c_1 + c > n/2$.
- Nên phần còn lại là : $(n-1) - (c_1 + c) < n/2$.
- Vậy tập 1 sẽ có c_1 cái, tập 2 chứa ít nhất là c cũng $\leq n/2$ và tập cuối (là phần còn lại) cũng $\leq n/2$.

=> Ta tìm được 3 tập thỏa mãn.

Bài 8: AN TOÀN

Mã bài: SAFE.???

Hệ thống điện trong thành phố được cho bởi n trạm biến thế và các đường dây điện nối giữa các cặp trạm biến thế. Mỗi đường dây điện e có độ an toàn là $p(e) \in (0,1]$. Độ an toàn của cả lưới điện là tích độ an toàn trên các đường dây. Hãy tìm cách bỏ đi một số dây điện để cho các trạm biến thế vẫn liên thông và độ an toàn của mạng là lớn nhất.

Input:

- Dòng đầu 2 số nguyên dương n, m là số thành phố và số đường dây nối trực tiếp 2 thành phố. ($n, m \leq 100000$).
- M dòng tiếp theo 3 số u, v, w là chỉ số 2 thành phố và độ an toàn tương ứng.

Output:

- Số thực duy nhất là độ an toàn lớn nhất của mạng, chính xác tới 3 chữ số sau dấu thập phân.

INPUT	OUTPUT
3 3 1 2 0.5 2 3 0.1 3 1 0.2	0.1

Solution:

- Ta thấy $\ln(A * B) = \ln A + \ln B$.

Do vậy độ an toàn trên lưới điện trở thành tổng độ an toàn trên các đường dây, khi đó ta có thể dùng thuật toán Kruskal để giải bài toán trên.

Bài 9: CÂY

Mã bài: QTREEV.???

Cho một đồ thị cây có N đỉnh $N-1$ cạnh, gốc của cây là đỉnh 1, mỗi đỉnh có một trọng số không âm là A_i . Dễ dàng nhận thấy, ngoại trừ đỉnh 1, các đỉnh còn lại đều có một đỉnh cha và nhận nó làm đỉnh con. Từ mảng A người ta tiến hành xây dựng mảng P như sau:

$P_u = A_u$ nếu như đỉnh u đó không có đỉnh con, ngược lại $P_u = A_u * \max(P_{v1}, P_{v2} \dots P_{vm})$ với $v1, v2 \dots vm$ lần lượt là các đỉnh con trực tiếp có cạnh nối với u . Nhiệm vụ của bạn là tính P_1 .

Input

Dòng 1: Gồm một số nguyên N và M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^{18}$).

Dòng 2: Gồm N số nguyên $A_1 \dots A_N$ với A_i là trọng số của đỉnh thứ i . ($A_i \leq 10^{18}$).

$N-1$ dòng tiếp theo, mỗi dòng gồm hai số nguyên u và v , thể hiện cạnh nối giữa u và v ($1 \leq u, v \leq N$).

Output

Một số nguyên duy nhất là P_1 , do kết quả có thể rất lớn nên chỉ cần in ra phần dư cho một số nguyên M

Example

Input	Output
3 1000	3
1 2 3	
1 2	
1 3	

Solution:

Đây là QHĐ trên cây cơ bản, song cần sử dụng một số mẹo sau:

- $A > B \Leftrightarrow \log A > \log B$.
- $\log A * B = \log A + \log B$.
- Kết hợp với phép Mod số lớn đã được nói ở bài SUM2N (Dùng cách 1 nhé!).

Đối với riêng bài này ta có thể dùng \ln cho nhanh.

Bài 10: SỐ MDS

Mã bài: MDS.???

Một số tự nhiên n được gọi là số MDS nếu n chia hết cho tổng các chữ số của n .

Yêu cầu:

Cho số s , tìm số tự nhiên nhỏ nhất là số MDS mà có tổng các chữ số bằng s .

Input

- Gồm nhiều test, một dòng một test chứa một số s ($s \leq 1000$)

Output

- Ghi số MDS mà có tổng các chữ số bằng s .

MDS.INP	MDS.OUT
5	5
10	190

Solution:

Với dữ liệu nhỏ ta có thể for trâu để tìm kết quả.

Tư tưởng chính của bài toán là dùng thuật toán BFS. Chi tiết Code:

```

20 Procedure Sol(s:longint);
21 var i,j,x,y,u,v,d,c:longint;
22 begin
23     for i:=0 to s do
24         for j:=0 to s do Free[i,j]:=true;
25     d:=1; c:=1; Qx[d]:=0; Qy[d]:=0; Id[d]:=0; Free[0,0]:=false;
26     While d<=c do
27     begin
28         x:=Qx[d]; y:=Qy[d]; inc(d);
29         for i:=0 to 9 do
30         begin
31             u:=x+i; v:=(y*10+i) mod s;
32             if Free[u,v] then
33             begin
34                 inc(c); Qx[c]:=u; Qy[c]:=v; Free[u,v]:=false;
35                 Id[c]:=i; Trx[c]:=d-1;
36                 if (u=s) and (v=0) then begin Save:=c; Exit; end;
37             end;
38         end;
39     end;
40 end;
41
42 Procedure Trace(i,s:longint);
43 var sum:longint;
44 begin
45     sum:=0; nres:=0;
46     While not ((Id[i]=0) and (sum=s)) do
47     begin
48         inc(nres); res[nres]:=Id[i]; Sum:=Sum+Id[i]; i:=Trx[i];
49     end;
50 end;

```


CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Bài 11: VIỆN TRỢ

Mã bài: ODAROAD.???

Tỉnh miền núi zZz có N làng, đánh số 1,2,...,N. Có M cặp làng có thể tiến hành ủi đất làm đường.

Tỉnh zZz muốn làm đường sao cho làng 1 và làng N trở nên liên thông. Hệ thống đường sẽ được hoàn thành dựa vào viện trợ ODA, vốn nhà nước và đóng góp của người dân như sau:

- Giả sử có C con đường được chọn để xây dựng
- Viện trợ ODA dùng để thực hiện không quá K đoạn đường
- Người dân đóng góp xây dựng đoạn đường dài nhất trong C-K đoạn đường còn lại
- Vốn nhà nước dùng để thực hiện các đoạn đường còn lại (C-K+1 đoạn đường cuối cùng)

Hãy xác định số tiền ít nhất người dân phải đóng góp để hoàn thành hệ thống đường đảm bảo liên thông giữa làng 1 và làng N.

Dữ liệu (odaroad.inp)

- Dòng 1: ba số nguyên N, M, K ($1 \leq N \leq 1000; 1 \leq M \leq 10000; 0 \leq K < N$)
- Dòng 2...M+1: dòng i+1 chứa ba số nguyên A_i, B_i, L_i ($1 \leq A_i, B_i \leq N; 1 \leq L_i \leq 10^6$) chỉ hai làng A_i, B_i có thể xây dựng đường nối trực tiếp với chi phí L_i

Kết quả (odaroad.out)

- Dòng 1: số nguyên kết quả, số này bằng -1 nếu không có cách xây dựng đường.

ODAROAD.INP	ODAROAD.OUT
5 7 1	4
1 2 5	
3 1 4	
2 4 8	
3 2 3	
5 2 9	
3 4 7	
4 5 6	

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Solution:

Tư tưởng chính của thuật toán là Tìm kiếm nhị phân + DFS.

Bài toán của ta sẽ trở thành : Hỏi có thể đi được từ 1 đến N mà không có quá K con đường có độ dài vượt quá L?

Chặt nhị phân độ dài L. Để kiểm tra xem có thể đi được từ 1 -> N mà không có quá K con đường có độ dài vượt quá L hay không ta dùng thủ tục DFS(u, kx, l) có nghĩa là có đến được đỉnh u mà không có quá kx con đường mà độ dài vượt quá l ?. Ở đây trạng thái đánh dấu là Free[u,kx]=True/False. (u=1...N, kx=0...K).

Bài 12: STRENGTH OR INTELLECT

Mã bài: SOL.???

Nhân vật trong trò chơi của bạn gồm 2 thông số : sức mạnh và độ thông minh. Ban đầu, cả hai chỉ số này đều bằng 1. Để nâng cao giá trị này, bạn có thể thực hiện một số nhiệm vụ có trong trò chơi. Mỗi nhiệm vụ được biểu diễn bằng 3 chỉ số (x, y, z) thể hiện rằng: để hoàn thành nhiệm vụ này, sức mạnh cần có ít nhất là x **hoặc** độ thông minh cần có ít nhất là y, và bạn sẽ được z điểm kinh nghiệm sau khi hoàn thành nhiệm vụ. 1 điểm kinh nghiệm có thể nâng giá trị của sức mạnh hoặc độ thông minh lên 1 tùy theo lựa chọn của bạn.

Việc thực hiện các nhiệm vụ có thể diễn ra theo tùy ý, tuy nhiên mỗi nhiệm vụ chỉ được làm không quá 1 lần. Tính số nhiệm vụ tối đa nhân vật của bạn có thể làm được.

Input:

- Dòng đầu tiên chứa số N là số lượng nhiệm vụ ($0 < N \leq 50$).
- 3 dòng tiếp theo, mỗi dòng chứa N số nguyên dương không vượt quá 1000 cách nhau bởi dấu cách. Số thứ i trong các dòng lần lượt mô tả các yêu cầu sức mạnh, yêu cầu độ thông minh và điểm kinh nghiệm đạt được của nhiệm vụ thứ i.

Output:

- 1 dòng duy nhất là số nhiệm vụ có thể đạt được.

Example:

SOLINP	SOLOUT
2 1 2 1 2 1 2	2
5 1 3 1 10 3 1 1 3 20 3 2 1 1 5 1	4

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Solution:

Tư tưởng chính là thuật toán DFS. Trạng thái của các đỉnh (st, il) là độ sức mạnh và độ thông minh hiện đang có. Trong quá trình duyệt DFS việc cộng kinh nghiệm sẽ cộng từng đơn vị tức là từ đỉnh $(u,v) \rightarrow (u+1,v)$ hoặc $(u,v+1)$. Trạng thái đánh dấu $Free[st, il] = T/F$ tức là đã từng có độ sức mạnh st và độ thông minh il hay chưa.

Bài 13: TÔ MÀU

Mã bài: COLOUR.???

Cho N số từ 1 đến N được viết vào các ô của bảng kích thước $2 \times N$ sao cho mỗi số xuất hiện đúng 2 lần trong bảng và mỗi cột chứa 2 số khác nhau: Ví dụ:

1	5	3	1	5
4	2	2	4	3

Chúng ta sẽ tô màu mỗi ô bằng màu trắng hoặc đen sao cho 2 ô ở cùng một cột sẽ tô màu khác nhau và các ô chứa các số bằng nhau cũng sẽ tô màu khác nhau: Ví dụ trên chúng ta có thể tô màu nhau sau:

1	5	3	1	5
4	2	2	4	3

Yêu cầu: Có bao nhiêu cách tô màu như vậy?

Dữ liệu: Vào từ file văn bản COLOUR.INP

- Dòng 1: Ghi số nguyên dương N ($2 \leq N \leq 100$)
- 2 dòng tiếp theo, mỗi dòng ghi N số nguyên trong đoạn $[1, N]$, các số này mô tả các số trên bảng $2 \times N$

Kết quả: Ghi ra file văn bản COLOUR.OUT

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

- Một số duy nhất là số cách sử dụng hai màu để tô bảng $2 \times N$.

Ví dụ:

COLOUR.INP	COLOUR.OUT
5 1 5 3 1 5 4 2 2 4 3	4

Solution:

Đây là bài toán cơ bản về tô màu đồ thị.

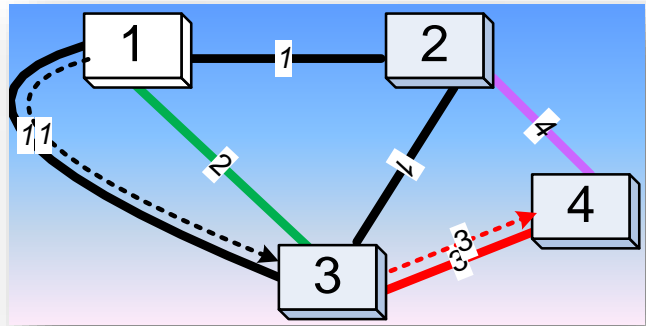
Chúng ta sẽ xây dựng đồ thị 2 chiều cột i nối được với cột $j \Leftrightarrow$ có 1 trong 2 giá trị trong cột i bằng 1 trong 2 giá trị trong cột j .

Kết quả = $2^{(\text{số thành phần liên thông})}$.

Bài 14: ĐƯỜNG ĐI LÝ TƯỞNG

Mã bài: IDEAL.???

Mê cung có n phòng đánh số từ 1 đến n , được nối với nhau bởi m đoạn đường đi 2 chiều nối trực tiếp 2 phòng, đường đi thứ i có màu c_i . Giữa 2 phòng có thể có nhiều đường đi nối trực tiếp. Đường đi trực tiếp có thể nối một phòng với chính nó. Người chơi được máy bay lên thẳng thả xuống phòng 1 và phải tìm đường đi tới phòng n . Độ dài của đường đi được tính bằng số đoạn đường nối giữa 2 phòng đã đi qua. Ai có đường đi ngắn nhất sẽ thắng cuộc. Nếu có nhiều người cùng có đường đi ngắn nhất thì so sánh theo thứ tự từ điển các màu đã đi qua. Người có đường đi ngắn nhất với thứ tự từ điển nhỏ nhất sẽ thắng. Đường đi lý tưởng là đường ngắn nhất và có thứ tự từ điển nhỏ nhất.



Nếu có nhiều người cùng có đường đi ngắn nhất thì so sánh theo thứ tự từ điển các màu đã đi qua. Người có đường đi ngắn nhất với thứ tự từ điển nhỏ nhất sẽ thắng. Đường đi lý tưởng là đường ngắn nhất và có thứ tự từ điển nhỏ nhất.

Yêu cầu: Cho m , n và các đường đi (a_i, b_i, c_i) xác định đường màu c_i nối từ phòng a_i tới phòng b_i ($2 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5, 1 \leq a_i, b_i \leq n, 1 \leq c_i \leq 10^9$). Dữ liệu đảm bảo có đường đi từ phòng 1 đến phòng n . Hãy xác định đường đi lý tưởng: số đoạn đường và màu của các đoạn đó theo trình tự đi.

Dữ liệu: Vào từ file văn bản IDEAL.INP:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Dòng thứ i trong m dòng sau chứa 3 số nguyên a_i, b_i và c_i .

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Kết quả: Đưa ra file văn bản IDEAL.OUT:

- Dòng đầu tiên chứa số nguyên k ,
- Dòng thứ 2 chứa k số nguyên – màu của các đoạn theo trình tự đi.

Ví dụ:

IDEAL.INP	IDEAL.OUT
4 6	2
1 2 1	1 3
1 3 2	
3 4 3	
2 3 1	
2 4 4	
3 1 1	

Solution:

Gọi $d1[u]$ là đường đi ngắn nhất từ 1 $\rightarrow u$, $d2[u]$ là đường đi ngắn nhất từ $n \rightarrow u$.

- Loại bỏ những đỉnh mà $d1[u] + d2[u] < d1[n]$.
- Tạo ra một đồ thị mới là đồ thị vô hướng không chu trình.

Xây dựng thuật toán BFS từ đỉnh 1.

Khi đang ở những đỉnh u ta nhét vào Queue những đỉnh v có $d1[v] - 1 = d1[u]$ và màu của con đường là nhỏ nhất. Ta cần phải duyệt 2 lần, lần đầu là để tìm ra màu nhỏ nhất còn lần 2 là để nhét các đỉnh v nối với u mà màu trùng với màu nhỏ nhất.

PHẦN 2: SỐ HỌC

Bài 1: ĐẾM TAM GIÁC

Mã bài: TRIANGLE.???

Cho 3 dãy số nguyên dương.

Dãy A: a_1, a_2, \dots, a_n

Dãy B: b_1, b_2, \dots, b_n

Dãy C: c_1, c_2, \dots, c_n

Xét 3 loại câu hỏi sau :

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

1. Có bao nhiêu bộ 3 chỉ số (i,j,k) mà a_i, b_j, c_k tương ứng là 3 cạnh của một tam giác đều
2. Có bao nhiêu bộ 3 chỉ số (i,j,k) mà a_i, b_j, c_k tương ứng là 3 cạnh của một tam giác cân (tam giác đều cũng tính là tam giác cân)
3. Có bao nhiêu bộ 3 chỉ số (i,j,k) mà a_i, b_j, c_k tương ứng là 3 cạnh của một tam giác.

Yêu cầu : Cho 3 dãy A, B, C và loại câu hỏi w (w bằng 1,2 hoặc 3 tương ứng là với câu hỏi loại 1,2 hoặc 3), tính số lượng số lượng bộ 3 chỉ số (i,j,k) cho câu hỏi loại w.

Input

Dòng đầu tiên chứa Q ($Q \leq 10$) là số bộ dữ liệu. Tiếp đến là Q nhóm dòng, mỗi nhóm cho thông tin về một bộ dữ liệu theo khuôn dạng sau đây:

- Dòng 1: chứa 2 số nguyên dương n, w;
- Dòng 2: ghi n số nguyên dương : a_1, a_2, \dots, a_n ($a_i \leq 10^9$);
- Dòng 3: ghi n số nguyên dương : b_1, b_2, \dots, b_n ($b_i \leq 10^9$);
- Dòng 4: ghi n số nguyên dương : c_1, c_2, \dots, c_n ($c_i \leq 10^9$);

Output

Gồm Q dòng, mỗi dòng chứa một số nguyên là số lượng bộ 3 chỉ số (i,j,k) cho câu hỏi w tương ứng với bộ dữ liệu vào.

TRIANGLE.INP	TRIANGLE.OUT
2	3
2 2	4
10 30	
30 10	
10 20	
2 3	
20 30	
30 10	
10 20	

Subtask 1: $n \leq 100$

[20 test]

Subtask 2: $n \leq 1000$; w=1;

[20 test]

Subtask 3: $n \leq 1000$; w=2;

[20 test]

Subtask 4: $n \leq 1000$; w=3

[20 test]

Subtask 5: $n \leq 1000$

[20 test]

Solution: Tìm kiếm nhị phân

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

TH1: Δ đều: Với mỗi i gọi x số lượng j thỏa mãn $a_i = b_j$, y là số lượng k thỏa mãn $a_i = b_k$.

Do đó với mỗi i số lượng Δ đều là $x*y$.

TH2: Δ cân: Gọi $x = \begin{cases} a_i = b_j \\ c_k < 2a_i \end{cases}$; $y = \begin{cases} a_i = c_k \\ b_j < 2a_i \end{cases}$; $z = \begin{cases} c_k = b_j \\ a_i < 2b_j \end{cases}$. Do đó số Δ cân là $x+y+z-2*$ số Δ đều.

TH3: Δ thường: $a_i + b_j > c_k > |a_i - b_j|$.

Bài 2: TAM GIÁC MÀU

Mã bài: COLTRI.???

Cho n điểm trên mặt phẳng, không có ba điểm nào thẳng hàng, các điểm được đánh số từ 1 đến n . Người ta nối tất cả các cặp điểm (i, j) bằng sợi dây màu xanh hoặc màu vàng theo nguyên tắc: Nếu $i+j$ là số nguyên tố thì điểm i nối với điểm j bằng sợi dây màu xanh, ngược lại nếu $i+j$ không phải số nguyên tố thì nối bằng sợi dây màu vàng. Sau đó người ta muốn khảo sát xem có bao nhiêu hình tam giác mà ba đỉnh là 3 điểm trong n điểm được nối với nhau bằng các sợi dây cùng màu.

Yêu cầu: Cho n , hãy đếm số hình tam giác mà ba đỉnh là 3 điểm trong n điểm được nối với nhau bằng các sợi dây cùng màu.

Input

- Dòng đầu tiên ghi số nguyên dương T ($T \leq 10$) là số lượng bộ dữ liệu. Tiếp đến là T dòng, mỗi dòng tương ứng với một bộ dữ liệu chứa một số nguyên.

Output

- Gồm T dòng, mỗi dòng chứa một số nguyên là số tam giác đếm được tương ứng với bộ dữ liệu vào.

COLTRI.INP	COLTRI.OUT
2	0
3	1
5	

Subtask 1: $n \leq 100$; [30 tests]

Subtask 2: $n \leq 1000$; [30 tests]

Subtask 3: $n \leq 10^6$; [40 tests]

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Solution:

Sub1: For 3 vòng.

Sub 2: Nhìn theo một hướng khác, thay vì đi tìm số Δ cùng màu ta đi đếm số Δ khác màu.

Với 1 đỉnh i , x là số lượng j thỏa mãn (i,j) là cạnh màu xanh, y là số lượng j thỏa mãn (i,j) là cạnh màu đỏ.

= > Số Δ khác màu là $(\sum x * y) \div 2$, Vì ta thấy một Δ khác màu được tính 2 lần.

- $Kq = C_n^3 - \text{Số } \Delta \text{ khác màu.}$

Sub 3: Trên cơ sở ý tưởng của Sub2, với mỗi i số lượng j thỏa mãn $i+j$ là số nguyên tố là $x=s[i+n]-s[i]$ với $s[i]$ là số lượng số nguyên tố từ 1 $\rightarrow i$, do đó $y=n-x-1$.

Note: TH $i = 1$.

Bài 3: CẮT BÁNH

Mã bài: CUT.???

Trong ngày sinh nhật của mình, Pet đã làm một chiếc bánh hình hộp với kích thước $A \times B \times C$. Có D bạn tới dự sinh nhật, Pet quyết định sẽ cắt chiếc bánh D lần, mỗi lần cắt sẽ cắt chiếc bánh song song theo một trong các cạnh của chiếc bánh và đưa cho một bạn. Sau D lần cắt, phần còn lại sẽ là phần của Pet, do đó Pet muốn tìm cách cắt chiếc bánh sao cho phần còn lại là lớn nhất (giả sử A', B', C' lần lượt là các cạnh của chiếc bánh sau khi cắt thì $A' \times B' \times C'$ đạt giá trị lớn nhất).

Yêu cầu: Cho A, B, C, D , đưa ra giá trị $A' \times B' \times C'$.

Input

Dòng đầu tiên ghi số nguyên dương T là số lượng bộ dữ liệu. Tiếp đến là T dòng, mỗi dòng chứa 4 số nguyên A, B, C, D ($0 \leq D \leq A + B + C - 3$) tương ứng với một bộ dữ liệu.

Output

Gồm T dòng, mỗi dòng là kết quả tìm được (kết quả không vượt quá $2^{64} - 1$)

CUT.INP	CUT.OUT
3	120
4 5 6 0	64
4 5 6 3	1
1 1 10 9	

Subtask 1: $A, B, C, D \leq 100$;

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Subtask 2: $A, B, C \leq 10^{18}; D \leq 1000$;

Subtask 3: $A, B, C, D \leq 10^{18}$.

Solution: Dĩ nhiên mỗi nhát cắt ta chỉ cắt 1 đơn vị.

Sub1 : Giả sử cắt ở các cạnh A,B,C lần lượt là X, Y, Z.

Duyệt X, Y, Z sao cho thỏa mãn $\begin{cases} X + Y + Z = D \\ (A - X) * (B - Y) * (C - Z) \text{ Max} \end{cases}$

Sub2 : Duyệt X, Y $\Rightarrow Z = D - X - Y$

Sub3 : Ta thấy để $A' * B' * C'$ Max thì A', B', C' càng gần nhau càng tốt.

Giả sử $A \leq B \leq C$.

Cắt lần 1 \Rightarrow cắt C.

1. $D \leq C - B \Rightarrow$ cắt được A, B, C - D

2. $D > C - B$. Giả sử $A \leq B \leq C \Rightarrow D' = D - (B - C)$.

a) $D' \leq 2 * (B - A) \Rightarrow$ 3 cạnh $A, B - \frac{D'+1}{2}, B - \frac{D'}{2}$

b) $D > 2 * (B - A) \Rightarrow$ 3 cạnh A, A, A . $D'' = D' - 2 * (B - A)$.

- 3 cạnh còn lại là $A - \frac{D''+2}{3}; A - \frac{D''+1}{3}; A - \frac{D''}{3}$

Bài 4: PHÂN SỐ

Mã bài: FRAC.???

Trong toán học, số hữu tỉ là các số x có thể biểu diễn dưới dạng phân số $\frac{a}{b}$ (trong đó a và b là các số nguyên), cũng có thể biểu diễn dưới dạng thập phân.

Ví dụ: $\frac{2}{25} = 0.08000 \dots; \frac{5}{7} = 0.714285714 \dots$

Yêu cầu: Cho một số hữu tỉ biểu diễn dưới dạng phân số $\frac{P}{Q}$ và hai số s, t. Hãy tính tổng các chữ số từ vị trí thứ s đến vị trí thứ t (tính từ chữ số đầu tiên sau dấu chấm thập phân) trong biểu diễn thập phân chia dư cho Q.

Input

Dòng đầu tiên ghi số nguyên dương T ($T \leq 1000$) là số lượng bộ dữ liệu. Tiếp đến là T dòng, mỗi

dòng tương ứng với một bộ dữ liệu, dòng chứa 4 số nguyên dương P, Q, s, t.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Output

Gồm T dòng, mỗi dòng là kết quả tính được.

FRAC.INP	FRAC.OUT
2	8
2 25 1 100	5
5 7 1 3	

Subtask 1: $P, Q \leq 10^9, s \leq t \leq 10$; [20 tests]

Subtask 2: $P, Q \leq 10^9, s \leq t \leq 100$; [10 tests]

Subtask 3: $P, Q \leq 10^9, s \leq t \leq 10^9$; [10 tests]

Subtask 4: $P, Q \leq 10^9, s = t \leq 10^9$; [10 tests]

Solution:

Sub2 : Mô tả phép chia số thực

Sub3 : Tồn tại chu trình. Tìm chu trình tính tổng.

Sub4 : Tìm 1 chữ số.

Mô tả phép chia :

$$\left(\begin{array}{l} \text{For } i := 1 \text{ to } t \text{ do} \\ \quad \{p := p * 10; p \text{ div } q; \\ \text{if } i \geq s \text{ then } sum := sum + P \text{ div } Q \\ \quad p := p \text{ mod } q; \} \end{array} \right)$$

✚ Nhận xét 1: phép tính $p := p * 10$ (t-1 lần) $\Rightarrow p := p * 10^{t-1} \bmod q$

✚ Nhận xét 2: $p := p * 10 \text{ div } q \Rightarrow$ chữ số cần tìm.

Bài 5: BẢNG SỐ NGUYÊN

Mã bài: MSQR.???

Cho bảng số nguyên không âm gồm $m \times n$, các hàng của bảng được đánh số từ 1 đến m, các cột

của bảng được đánh số từ 1 đến n. Ô nằm giao giữa hàng i và cột j là ô (i, j) ghi số nguyên không âm a_{ij} ($a_{ij} \leq 10^{15}$).

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Yêu cầu: Cho bảng số và số nguyên w , tìm một bảng hình vuông có kích thước lớn nhất mà tổng

các số nằm trên các ô ở biên không vượt quá w .

Input

- Dòng đầu chứa 3 số nguyên dương n, m, w ($w \leq 10^{15}$);
- m dòng sau, mỗi dòng chứa n số nguyên không âm mô tả bảng số.

Output

- Gồm một dòng chứa một số nguyên là kích thước cạnh hình vuông tìm được.

MSQR.INP	MSQR.OUT
4 5 10 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1	4

Subtask 1: $m, n \leq 100$;

Subtask 2: $m, n \leq 1000$.

Solution:

Sub1: Trâu thoi. For 3 vòng, ô (i, j) , cạnh độ dài k .

Sub2: Trâu + Cận tham. Xét ô (i, j) , giả sử đến độ dài k từ $best+1$ trở đi mà tổng các cạnh $\leq w$ thì tăng k ngược lại break.

Chạy cả 4 góc cho chắc.

Note : Bắt buộc phải cài cả 2 sub vì thuật toán cho Sub2 không phải là thuật toán chuẩn.

Bài 6: DUYỆT ĐỒNG DIỄN

Mã bài: DDD.???

Có $m \cdot n$ bạn học sinh gồm cả nam và nữ tham gia duyệt đồng diễn kỷ niệm 1000 năm Thăng Long Hà Nội. Trong quá trình biểu diễn, tất cả $m \cdot n$ bạn xếp thành m hàng, mỗi hàng n bạn. Kết thúc màn trình diễn, tất cả xếp thành một hàng để chào khán giả theo nguyên tắc, n bạn hàng 1 xếp đầu hàng (giữ nguyên thứ tự trong hàng), sau đó bạn hàng 2,..., và cuối cùng là n bạn hàng m . Trong buổi biểu diễn chính thức, Ban tổ chức muốn các bạn đứng vào đúng

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

vị trí như buổi duyệt đồng diễn. Tuy nhiên, thông tin mà Ban tổ chức ghi nhận lại được chỉ là các thông tin thống kê về các hàng và cột, cụ thể hàng thứ i gồm a_i bạn nam, cột thứ j gồm b_j bạn nam. **Yêu cầu:** Cho $m, n, a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n$. Hãy tìm cách xếp $m \times n$ học sinh m hàng, n cột sao cho tổng số các bạn nam ở hàng i là a_i , tổng số các bạn nam ở cột j là b_j , nếu có nhiều phương án thì đưa ra phương án mà khi $m \times n$ học sinh xếp thành một hàng sẽ tạo thành một dãy bit có thứ tự từ điển nhỏ nhất (coi nam là bit 1 và nữ là bit 0).

Input

Gồm nhiều bộ dữ liệu, mỗi bộ gồm một số dòng có khuôn dạng như sau:

- Dòng đầu chứa hai số nguyên m, n ;
- Dòng thứ hai gồm m số a_1, a_2, \dots, a_m
- Dòng thứ ba gồm n số b_1, b_2, \dots, b_n .

Kết thúc file là 2 số 0.

Output

Với mỗi bộ dữ liệu, ghi dãy bit có thứ tự từ điển nhỏ nhất tìm được hoặc ghi -1 nếu không tồn tại cách xếp thỏa mãn.

DDD.INP	DDD.OUT
3 3	011001110
2 1 2	-1
1 2 2	
2 2	
1 2	
1 1	
0 0	

Subtask 1: $m, n \leq 5$;

Subtask 2: $m \leq 10; n \leq 5$;

Subtask 3: $m, n \leq 50$.

Solution:

Xét bài toán gốc sau:

Cho danh sách bậc của các đỉnh trong một đồ thị vô hướng n đỉnh. Xây dựng lại đồ thị.

Cách làm bài toán trên như sau.

1. Sort lại danh sách bậc theo thứ tự giảm dần.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

2. Nối 2 đỉnh có thứ tự 1 với 2, sau đó giảm bậc của 2 đỉnh đó.
3. Làm 2 bước 1 và 2 cho đến khi tất cả bậc của các đỉnh đều =0.

Bây giờ trở về bài toán của chúng ta:

Sub 1: Duyệt trâu.

Try ra dãy nhị phân rồi nhét vào bảng, sau đó ta đi kiểm tra xem có thỏa mãn đề bài hay không

Sub 2: Duyệt + Cận

Vẫn Try nhị phân tương tự như Sub1, ta khi Try ra số 0 or 1 thì ta nhét nó luôn vào bảng và kiểm tra xem nó có thỏa mãn đề bài hay không, do Try 0 trước 1, từ trái qua phải, từ trên xuống dưới nên khi tìm ra dãy thỏa mãn thì ta exit luôn.

Sub3:

- Trước tiên chưa xét gì đến thứ tự từ điển, chỉ cần tìm ra bảng thỏa mãn mảng a và b:
 1. Xét hàng 1 của bảng. Sort lại b[1] bằng đếm phân phối hoặc Sort. Đặt a[1] số 1 vào a[1] cột mà có b[j] lớn nhất sau đó giảm các b[j] đó đi 1 đơn vị.
 2. Làm từ 2->m tương tự như hàng 1, ta được bảng thỏa mãn các giá trị a, b.
- Xét theo thứ tự từ điển nhỏ nhất:
 1. Giả sử giá trị ô [i,j]=0 ($i=1..m, j=1..n$), sau đó đặt các ô còn lại như trên (tức không theo thứ tự từ điển), nếu đặt được thì ghi nhận giá trị ô [i,j] là 0 ngược lại ghi nhận giá trị ô [i,j] là 1 và giảm a[i], b[j] của hàng và cột đó.

Bài 7: Dãy Số

Mã bài: SEQ.???

Cho dãy số gồm n số nguyên $1, 2, \dots, n$. Một đoạn con của dãy đã cho là dãy a_i, \dots, a_j ($1 \leq i \leq j \leq n$), dãy có độ dài $(j - i + 1)$ và có trọng số bằng tổng $(a_i + \dots + a_j)$.

Yêu cầu: Tìm hai đoạn con không có phần tử chung, mỗi đoạn có độ dài là một số chia hết cho 3 và tổng trọng số của hai đoạn con là lớn nhất.

Dữ liệu: Vào từ file văn bản SEQ.INP có định dạng như sau:

- Dòng đầu ghi số nguyên ($n \geq 6$);
- Dòng thứ hai ghi n số nguyên $a_1, a_2, \dots, a_n (|a_i| \leq 10^9)$.

Kết quả: Ghi ra file văn bản SEQ.OUT một số là tổng trọng số của hai đoạn con tìm được.

Ví dụ:

SEQ.INP	SEQ.OUT
---------	---------

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

11 -1 3 -1 -9 -1 1 1 1 1 1 -9	5
----------------------------------	---

Subtask 1: Có 30% số test có $n \leq 20$;

Subtask 2: Có 30% số test có $n \leq 200$;

Subtask 3: Có 20% số test khác có $n \leq 2000$;

Subtask 4: Có 20% số test còn lại có $n \leq 200000$.

Solution:

Sub1 : For 4 vòng chọn các vị trí i, j, x, y , có sử dụng mảng tổng dồn $sum[i]$ là tổng các số từ $1 \rightarrow i$.

Kiểm tra :

- $j-i+1$ có chia hết cho 3 ko.
- $y-x+1$ có chia hết cho 3 ko.

Cập nhật kết quả max.

Sub 2: Gọi $f[i]$ là giá trị SEQ lớn nhất mà kết thúc tại i . For 3 vòng j, x, y sau đó kiểm tra:

- $y-x+1$ có chia hết cho 3 không.
- Cập nhật kết quả max cho $f[j]+sum(x,y)$

Sub 3: Duyệt vị trí i, j cùng với mảng tính trước trên dãy số từ $j+1 \rightarrow n$ có đoạn thỏa mãn max là bao nhiêu.

Sub 4: QHĐ.

- Gọi $l[i]$ là trọng số đoạn $[x,y]$ mà $1 \leq x \leq y \leq i$, thỏa mãn $\begin{cases} y-x+1 \bmod 3 = 0 \\ sum(x,y) \text{ max} \end{cases}$
- Gọi $r[i]$ là trọng số đoạn $[x,y]$ mà $i \leq x \leq y \leq n$, thỏa mãn $\begin{cases} y-x+1 \bmod 3 = 0 \\ sum(x,y) \text{ max} \end{cases}$
- Gọi $s[i]$ là tổng từ $1 \rightarrow i$, $smin[i]$ là giá trị min của $s[j]$ ($1 \leq j \leq i-3$) thỏa mãn $i-j \bmod 3 = 0$.

Cách xây dựng mảng l như sau:

$$\left[\begin{array}{l} \text{for } i := 3 \text{ to } n \text{ do} \\ \quad \text{begin} \\ \quad \quad s[i] := s[i-1] + a[i]; \\ \quad \quad smin[i] := \text{Min}(s[i], smin[i-3]); \\ \quad \quad l[i] := \text{Max}(l[i-1], s[i] - smin[i-3]); \\ \quad \text{end} \end{array} \right]$$

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Mảng r được xây dựng tương tự như trên.

Chúng ta có thể có cách QHĐ như sau:

Gọi $f[i,0/1]$ là giá trị cần tìm thỏa mãn đầu bài khi xét trên dãy từ $1 \rightarrow i$, không hoặc có chọn i .

Bài 8: PHÂN ĐOẠN DÃY SỐ

Mã bài: SEQUENCE.???

Bạn được tham gia trò chơi với dãy gồm n số nguyên không âm. Trong trò chơi này bạn phải chia dãy đã cho ra thành $k+1$ đoạn khác rỗng. Để thu được $k+1$ đoạn bạn cần lặp lại các bước sau đây k lần:

1. Chọn một đoạn tùy ý với nhiều hơn một phần tử (đầu tiên bạn chỉ có một đoạn - đó chính là cả dãy ban đầu).
2. Chọn một vị trí nào đó giữa hai phần tử của đoạn đã chọn để chia nó ra làm hai đoạn mới khác rỗng. Mỗi lần thực hiện xong hai bước này bạn nhận được một điểm số bằng tích của hai tổng các số trong hai đoạn mới chia ra. Bạn mong muốn đạt được tổng điểm lớn nhất.

Input

Dòng đầu tiên của file dữ liệu vào chứa hai số nguyên n và k ($k+1 \leq n$). Dòng thứ hai của dữ liệu chứa n số nguyên không âm a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^4$) là các phần tử của dãy số.

Output

Dòng đầu tiên ghi tổng điểm lớn nhất mà bạn đạt được.

SEQUENCE.INP	SEQUENCE.OUT
7 3 4 1 3 4 0 2 3	108

Note

Trong ví dụ bạn có thể giành được 108 điểm theo cách sau:

- Đầu tiên bạn có dãy số (4, 1, 3, 4, 0, 2, 3) gồm 1 đoạn. Bạn chia dãy ra thành hai đoạn sử dụng điểm chia sau phần tử thứ nhất và nhận được $4 \times (1 + 3 + 4 + 0 + 2 + 3) = 52$ điểm.
- Bạn đang có hai đoạn (4), (1, 3, 4, 0, 2, 3). Bạn chia dãy sau phần tử thứ ba và nhận được $(1 + 3) \times (4 + 0 + 2 + 3) = 36$ điểm.
- Bạn đang có ba đoạn (4), (1,3), (4, 0, 2, 3). Bạn chia dãy sau phần tử thứ năm và nhận được $(4 + 0) \times (2 + 3) = 20$ điểm.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Như vậy, sau các bước thực hiện nói trên bạn chia dãy số thành 4 đoạn (4), (1,3), (4,0), (2, 3) và nhận được $52 + 36 + 20 = 108$ điểm.

Subtask 1: $n \leq 30$.

Subtask 2: $n \leq 300$.

Solution:

- Sub1: $F[i,j,k]$ là điểm nhận được lớn nhất khi chia đoạn (i,j) thành $k+1$ đoạn.
 - $F[i,j,k] = F[i,m,k'] + F[m+1,j,k-k'] + \sum(i,m) * \sum(m+1,j)$.
- Sub2: Ta thấy thứ tự tách không quan trọng nên có $F[i,k]$ là điểm nhận được lớn nhất khi chia đoạn $(1 \rightarrow i)$ thành $k+1$ đoạn.
 - $F[i,k] = F[i',k-1] + \sum 1 \rightarrow i' * \sum i' + 1 \rightarrow i$.

Bài 9: SỐ TRUNG BÌNH

Mã bài: AVG.???

Cho dãy số nguyên không âm a_1, a_2, \dots, a_n , hãy đếm số bộ ba chỉ số $i < k < j$ sao cho $a_i + a_j = 2 * a_k$.

Input

Dòng đầu tiên ghi số nguyên dương T ($T \leq 10$) là số lượng bộ dữ liệu. Tiếp đến là T nhóm dòng, mỗi nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai gồm n số a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$).

Output

- Gồm T dòng, mỗi dòng chứa một số là kết quả tương ứng với bộ dữ liệu vào.

AVG.INP	AVG.OUT
2	0
3	1
1 2 5	
3	
1 1 1	

Subtask 1: $n \leq 100$;

Subtask 2: $n \leq 1000$.

Solution:

Sub 1: Dĩ nhiên là duyệt 3 vòng for i, j, k rồi, sau đó kiểm tra xem thỏa mãn đề bài hay không, nếu thỏa mãn ta tăng đếm nên.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Sub 2:

- Sort lại mảng a có kèm theo chỉ số.
- Duyệt 2 vòng for i,j sau đó tìm kiếm nhị phân để được số lượng k thỏa mãn $a[k] = (a[i] + a[j]) \div 2$. Với mỗi cặp (i,j), để đếm được số lượng k thỏa mãn. Gọi u, v lần lượt là vị trí k nhỏ nhất, lớn nhất thỏa mãn
$$\begin{cases} cs[k] \leq j \\ cs[k] \geq i \\ (a[i] + a[j]) \div 2 = a[k] \end{cases}$$
- Số lượng k thỏa mãn với mỗi cặp (i,j) là v-u+1.

Bài 10: NGÂN HÀNG

Mã bài: BANK.???

Có n người đến ngân hàng để rút tiền, người thứ i cần rút đúng a_i . Hiện tại ngân hàng chỉ còn đúng m tờ tiền, tờ thứ i có giá trị b_i .

Yêu cầu: Hãy kiểm tra xem ngân hàng có thể đáp ứng yêu cầu của n người không?

Input

Dòng đầu tiên ghi số nguyên dương T ($T \leq 5$) là số lượng bộ dữ liệu. Tiếp đến là T nhóm dòng, mỗi nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng đầu chứa hai số nguyên n, m;
- Dòng thứ hai gồm n số a_1, a_2, \dots, a_n ;
- Dòng thứ ba gồm m số b_1, b_2, \dots, b_m ;

Output

- Gồm T dòng, mỗi dòng ghi phương án tương ứng với bộ dữ liệu trong file dữ liệu vào. Cụ thể, mỗi dòng là một dãy số gồm m số, số thứ i bằng j, nếu tờ thứ i trả cho người thứ j (j=0 nếu tờ thứ i không dùng). Trong trường hợp không có cách nào đáp ứng yêu cầu của n người chỉ ghi một số -1.

BANK.INP	BANK.OUT
2	0 0 1 0 1
1 5	-1
8	
4 2 5 1 3	
2 6	
9 10	
5 4 8 6 3 11	

Subtask 1: $n = 1, m \leq 20; a_i, b_i \leq 10^9$

[25 tests]

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Subtask 2: $n \leq 2, m \leq 30; a_i, b_j \leq 1000$ [25 tests]

Subtask 3: $n \leq 10, m \leq 10; a_i, b_j \leq 10^9$ [25 tests]

Subtask 4: $n \leq 20, m \leq 20; a_i, b_j \leq 1000$ [25 tests]

Solution:

Sub 1: Duyệt nhị phân .

Sub 2: Try(i,x,y) với ý nghĩa là duyệt đến tờ thứ i và người 1 đang nhận được x tiền , người 2 đang nhận được y tiền. Trạng thái đánh dấu là F[i,x,y].

Chi tiết đoạn code:

```
41 Procedure Try(i,x,y:longint);
42 begin
43     if ok then exit;
44     if i>m then
45     begin
46         if (x=a[1]) and (y=a[2]) then begin ok:=true; xb:=xx; end;
47         exit;
48     end;
49     xx[i]:=0;
50     if cx[i+1,x,y] then Try(i+1,x,y);
51     if (x+b[i]<=a[1]) and cx[i+1,x+b[i],y] then
52     begin
53         xx[i]:=1;
54         cx[i+1,x+b[i],y]:=false;
55         Try(i+1,x+b[i],y);
56     end;
57     if (y+b[i]<=a[2]) and cx[i+1,x,y+b[i]] then
58     begin
59         xx[i]:=2;
60         cx[i+1,x,y+b[i]]:=false;
61         Try(i+1,x,y+b[i]);
62     end;
63 end;
```

Sub 3: Try(t,p,s) với ý nghĩa đang ở trạng thái t (những tờ tiền đã/chưa được sử dụng, do là trạng thái bit 0/1 lên chuyển về hệ 10), đang trả cho người thứ p, và số tiền của người p trước khi trả là s.

Dùng mảng đánh dấu f[t,s,p] kiểu Logic để tránh việc tính toán lại và tăng tốc độ thuật toán.
Chi tiết đoạn code:

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

```
65 Procedure Try2(t,p:longint; s:int64);
66 var i:longint;
67 begin
68     if dx[t] then exit;
69     if ok then exit;
70     if (p>n) and (s=0) then
71     begin
72         xb:=x;
73         ok:=true;
74         exit;
75     end;
76     for i:=1 to m do
77         if ((t and (int64(1) shl (i-1)))=0) and (s+b[i]<=a[p]) then
78         begin
79             x[i]:=p;
80             if s+b[i]=a[p] then Try2(t or (int64(1) shl (i-1)),p+1,0)
81             ) else
82             Try2(t or (int64(1) shl (i-1)),p,s+b[i]);
83             x[i]:=0;
84         end;
85     dx[t]:=true;
86 end;
```

Note : Do các sub không bao quát nhau lên cẩn thận trong việc chia sub.

Bài 11: GẶP GỠ

Mã bài: MEET.???

Có n thành phố tại Đài Loan, tất cả nằm dọc theo một con đường cao tốc. Các thành phố được đánh số liên tiếp bắt đầu từ 1 đến n . Thành phố thứ i có khoảng cách đến vị trí bắt đầu con đường cao tốc là $d_i (i = 1, 2, \dots, n)$. An đang sống ở thành phố x , Bình đang sống ở thành phố y . Hai bạn muốn tìm một thành phố z để gặp nhau sao cho $\max\{|d_x - d_z|, |d_y - d_z|\}$ là nhỏ nhất.

Yêu cầu: Cho d_1, d_2, \dots, d_n và x, y , hãy tìm z để $\max\{|d_x - d_z|, |d_y - d_z|\}$ là nhỏ nhất.

Input

- Dòng đầu chứa số nguyên n, Q .
- Dòng thứ hai chứa n số nguyên dương $d_1, d_2, \dots, d_n (d_1, d_2, \dots, d_n \leq 10^9)$;
- Q dòng sau, mỗi dòng chứa 2 số x, y mô tả cho một câu hỏi.

Output

Gồm Q dòng (mỗi dòng tương ứng với một câu hỏi), dòng thứ i là giá trị $\max\{|d_x - d_z|, |d_y - d_z|\}$ nhỏ nhất tìm được.

MEET.INP	MEET.OUT
5 2	2

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

1 2 3 4 5 1 5 2 3	1
-------------------------	---

Subtask 1: $n, Q \leq 100$

[20 tests]

Subtask 2: $n, Q \leq 10^5$ và $d_1 < d_2 < \dots < d_n$

[2 tests]

Subtask 3: $n, Q \leq 10^5$

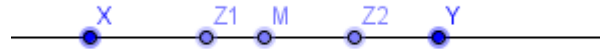
[3 tests]

Solution:

Sub 1: For Q , với mỗi cặp x, y ta for hết n thành phố z và cập nhật min cho kết quả. Độ phức tạp $O(n*Q)$

Sub 2: Ta thấy d được Sort tăng dần, lên làm tương tự Sub 3 nhưng bỏ qua quá trình Sort và do không liên quan đến chỉ số lên việc tìm kiếm là khá dễ dàng.

Sub 3:



- Sort lại mảng d kèm theo chỉ số.
- Hình dung các điểm như trên với M là trung điểm $[X, Y]$. Ta cần tìm $Z1$ lớn nhất mà nhỏ hơn M , $Z2$ là điểm nhỏ nhất mà lớn hơn M . Điểm Z cần tìm là $Z1$ hoặc $Z2$. Tìm $Z1, Z2$ bằng tìm kiếm nhị phân (nhớ là tìm bằng chỉ số).

Bài 12: CẮT DÂY

Mã bài: C11DEL.???

Songuku95 đố hgminh95 một bài toán như sau:

Xếp n củ Carrot trên một hình tròn (các củ carrot được đánh số từ 1 -> n). Khánh lấy n đoạn dây buộc 2 củ carrot liên tiếp lại (Củ 1 buộc với củ 2, củ 2 buộc với củ 3, ..., củ $n-1$ buộc với củ n , củ n buộc với củ 1). Bây giờ songuku95 đố hgminh95 bỏ một vài đoạn dây nối đi sao cho mỗi củ carrot còn ít nhất 1 đoạn dây nối. hgminh95 thông minh vốn sẵn tính trời nên việc này hoàn toàn là đơn giản với hgminh95 (Thực chất chả bỏ dây nào cũng đã đúng rồi :3). Cơ mà sau đấy, songuku95 lại thách hgminh95 đếm số cách bỏ một vài đoạn dây để vẫn thỏa mãn “mỗi củ carrot còn ít nhất 1 đoạn dây nối”. Bài toán này làm hgminh95 trầm trồ suy nghĩ suốt nhiều ngày liền. Vì quá khó nên hgminh95 đã nhờ đến sự cứu trợ từ người không quen (hoặc quen :))

Các bạn hãy giúp hgminh95 bài toán này nhé.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Input

Cho số test là T ($t \leq 5$)

T dòng tiếp theo chứa T số nguyên N ($3 \leq N \leq 50000$)

Output

Gồm T dòng

Dòng thứ i in ra số R = số cách bỏ một vài đoạn dây sao cho mỗi củ carrot vẫn còn ít nhất 1 dây nối đến nó.

Sample

C11DEL.INP	C11DEL.OUT
2	4
3	18
6	

Solution:

Giải bài toán trên trong trường hợp N củ carrot xếp thẳng hàng

Gọi $F[i]$ là số cách bỏ một số đoạn dây để trong i củ carrot ban đầu, củ nào cũng còn được nối (không được xóa 2 dây cùng chung củ carrot)

TH1: Để dây $(i-1) - (i) \Rightarrow F[i] = F[i-1]$

TH2: Xóa dây $(i-1) - (i) \Rightarrow F[i] = F[i-2]$

$\Rightarrow F[i] = F[i-1] + F[i-2]$

Quay lại bài toán hình tròn. Gọi $G[i]$ là số cách bỏ một số đoạn dây để trong hình tròn có i củ carrot, củ nào cũng còn được nối (không được xóa 2 dây cùng chung củ carrot)

TH1: Nếu không bỏ dây nối $(n) - (1): G[n] = F[n]$

TH2: Nếu bỏ dây $(n) - (1): G[n] = F[n-2]$ (số cách bỏ dây trong đoạn $[2..(n-1)]$)

$\Rightarrow G[n] = F[n] + F[n-2]$

$= (F[n-1] + F[n-2]) + (F[n-3] + F[n-4])$

$= (F[n-1] + F[n-3]) + (F[n-2] + F[n-4])$

$= G[n-1] + G[n-2]$

Bài 13: XÂU TƯƠNG ĐỒNG

Mã bài: HSTR.???

Người ta định nghĩa chuỗi X và chuỗi Y là tương đồng bậc k nếu:

- 1) Hai chuỗi có độ dài bằng nhau ($|X| = |Y|$);
- 2) Ký tự $X[i]$ và $Y[i]$ cách nhau không quá k vị trí trong bảng mã ASCII ($i = 1, 2, \dots, |X|$).

Yêu cầu: Cho số nguyên k và 2 chuỗi S_1, S_2 có độ dài bằng nhau (chuỗi chỉ gồm ký tự 'a' đến 'z'). Hãy xác định số cách cắt S_1 thành 3 chuỗi khác rỗng, sau đó ghép tạo thành chuỗi S_3 mà chuỗi S_3 tương đồng bậc k với chuỗi S_2 . Hai cách cắt được gọi là khác nhau nếu tồn tại một vị trí cắt khác nhau.

Input

- Dòng 1: chứa số nguyên k ;
- Dòng 2: chứa chuỗi S_1 ;
- Dòng 3: chứa chuỗi S_2 .

Output

- Số cách cắt thỏa mãn.

HSTR.INP	HSTR.OUT
0 beast betas	1
1 aaaaa bbbbbb	6

Subtask 1: $|S_1|=|S_2| \leq 300$; $k \leq 25$; [40 tests]

Subtask 2: $|S_1|=|S_2| \leq 3000$; $k = 0$; [30 tests]

Subtask 3: $|S_1|=|S_2| \leq 3000$; $k \leq 25$; [30 tests]

Sub 1 : Ta for 2 vòng i, j là vị trí cắt trên chuỗi S_1 , sau đó xây dựng thủ tục ghép 3 chuỗi con thành chuỗi S_3 , so sánh 2 chuỗi có tương đồng không bằng việc so sánh bằng từng vị trí nếu $\text{ord}(S_3[i]) = \text{ord}(S_2[i])$ thì tiếp tục nếu không thì exit, ta tăng kết quả lên với mỗi vị trí cắt i, j .

Chú ý : Với mỗi cặp i, j ta có 6 hoán vị, tức là có 6 chuỗi S_3 tương ứng.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Sub 2 : Hash để trả lời nhanh cho câu hỏi $S_1(i, j) = S_2(x, y)$.

Sub 3 : Qhđ dựa trên dãy con chung.

Gọi $f[i, j]$ là độ dài đoạn tương đồng dài nhất.

- $a[i]$ và $b[j]$ tương đồng thì $f[i, j] := f[i-1, j-1] + 1$.
- $a[i]$ và $b[j]$ không tương đồng thì $f[i, j] := 0$.

Xây dựng hàm kiểm tra $\text{Check}(i, j)$ tức là cắt tại i và j , có 6 xâu tương ứng được tạo thành với 1 cách cắt.

Bài 14: BẢNG TƯƠNG ĐỒNG

Mã bài: HTAB.???

Người ta định nghĩa bảng số A và bảng số B là tương đồng bậc k nếu:

- 1) Hai bảng có cùng kích thước (cùng số dòng, cùng số cột);
- 2) Hai phần tử tương ứng chênh lệch không quá k ($|A[i, j] - B[i, j]| \leq k$).

Yêu cầu: Cho số nguyên k và 2 bảng số P, Q có cùng kích thước $m \times n$, hãy tìm bảng số C và D tương ứng là bảng con của bảng số P và Q mà C và D tương đồng có cùng kích thước $r \times c$ và $r \times c$ đạt giá trị lớn nhất.

Input

- Dòng đầu chứa ba số nguyên m, n, k;
 - m dòng sau, mỗi dòng n số nguyên không âm mô tả bảng số P;
 - m dòng sau, mỗi dòng n số nguyên không âm mô tả bảng số Q;
- Các số không vượt quá 10^9 .

Output

- Giá trị $r \times c$ đạt giá trị lớn nhất tìm được.

HTAB.INP	HTAB.OUT
3 3 0 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 2 1 1 3 3 1	4 9

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 2 1 1	
--	--

Subtask 1: $m, n \leq 16; k \leq 10^9$; [40 tests]

Subtask 2: $m, n \leq 64; k = 0$; [30 tests]

Subtask 3: $m, n \leq 64; k \leq 10^9$; [30 tests]

Solution:

Sub 1 : Trâu với độ phức tạp $m * n * m * m^2 * n^2$

Sub 2 : So sánh 2 bảng giống nhau ta dùng Hash.

Sub 3 : Xây dựng mảng C (0/1) bằng cách đè 2 bảng lên nhau đè vào cả 4 góc.

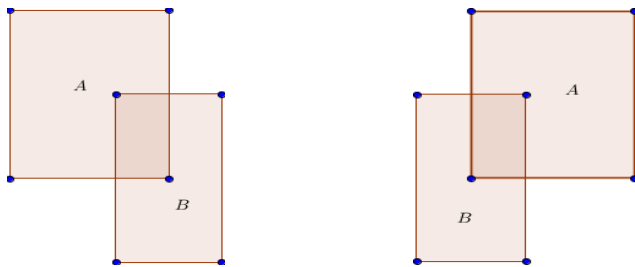
- $|A[i, j] - B[i, j]| \leq k$ thì giá trị 1.
- $|A[i, j] - B[i, j]| > k$ thì giá trị 0.

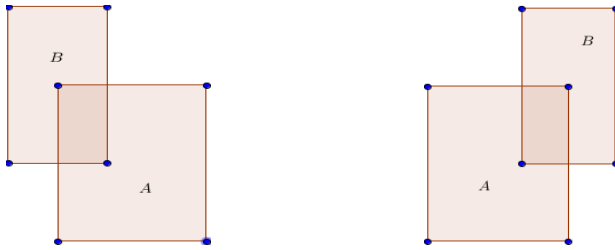
Với mỗi mảng C ta đưa bài toán về bài toán cơ sở:

Cho mảng a kích thước $m*n$ gồm giá trị 0/1 tìm hình chữ nhật lớn nhất của bảng mà gồm toàn số 1.

Hình chữ nhật lớn nhất tìm được là kết quả.

Sau đây là 4 vị trí trên bảng:





Bài 15: CHIA KẸO

Mã bài: CANDY.???

An và Bình có n gói kẹo, gói thứ i có a_i cái kẹo. An đề xuất thuật toán chia kẹo như sau: Ban đầu sắp xếp n gói kẹo theo một thứ tự, sau đó lần lượt lấy từng gói kẹo theo thứ tự để chia n gói kẹo thành 2 phần. Mỗi lượt, xem phần nào ít hơn thì cho gói kẹo vào phần đó, nếu trong trường hợp hai phần bằng nhau thì cho vào phần thứ nhất. Bình nhận ra ngay việc chia kẹo ảnh hưởng rất nhiều từ thứ tự sắp xếp n gói kẹo, do đó Bình đã chọn một thứ tự sắp xếp n gói kẹo và sẽ lấy phần thứ nhất.

Yêu cầu: Cho a_1, a_2, \dots, a_n là số lượng kẹo của n gói. Hãy tìm thứ tự các gói kẹo để Bình nhận được nhiều kẹo nhất.

Input

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai chứa n số nguyên dương $a_1, a_2, \dots, a_n (a_i \leq 1000)$.

Output

- Gồm một dòng chứa số nguyên là một hoán vị của $1, 2, \dots, n$, mô tả thứ tự các gói kẹo mà Bình sắp xếp.

CANDY.INP	CANDY.OUT
3 1 2 3	2 1 3

Subtask 1: $n \leq 8$; [40 tests]

Subtask 2: $n \leq 15$; [30 tests]

Subtask 3: $n \leq 1000$; $a_i = i$ [30 tests]

Solution:

Sub 1 : Duyệt hoán vị.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Sub 2 : Xây dựng thủ tục Try(t, ax, ay) với ý nghĩa là duyệt cấu hình đang ở trạng thái t (t là trạng thái 0/1/2 thỏa mãn

$\left\{ \begin{array}{l} 0 : \text{gói kẹo ở bit thứ } i \text{ của } t \text{ ở hệ tam phân chưa được chọn.} \\ 1 : \text{gói kẹo ở bit thứ } i \text{ của } t \text{ ở hệ tam phân được chọn cho thăng } 1; \text{ ax là thăng } 1 \text{ đang} \\ 2 : \text{gói kẹo ở bit thứ } i \text{ của } t \text{ ở hệ tam phân được chọn cho thăng } 2 \end{array} \right.$
có ax cái kẹo, thăng 2 có ay cái kẹo.

Ta thấy từ trạng thái của t ta có thể xây dựng được cách chọn lên ta dùng mảng đánh dấu $cx[t]$ (do $t \leq 3^{15}$) kiểu logic.

Sub 3 : Là mọi trường hợp đặc biệt.

Nhận xét:

- \sum số kẹo là chẵn \Rightarrow ta có thể chia sao cho chênh lệch lớn nhất là n .
- \sum số kẹo là lẻ \Rightarrow ta có thể chia sao cho chênh lệch lớn nhất là $n-1$.

TH 1: $n=4*k$ hoặc $n=4*k+1$, ta xếp lần lượt từ $n-1$ gói kẹo từ $n-1 \rightarrow 1$, sau đó xếp gói n vào cuối.

TH 2: $n=4*k+2$, Xếp $n-1, n-2, \dots, 2, n, 1$.

TH 3: $n=4*k+3$, Xếp $n-1, n-2, \dots, 3, 1, 2, n$.

Note: Khi chuyển từ hệ 3 sang hệ 10 và ngược lại ta cần thủ tục nhanh chóng không sẽ bị timelimit.

Bài 16: TRÒ CHƠI CHẴN LẺ VER1

Mã bài: PARIGAME_1.???

Trò chơi chẵn lẻ là trò chơi hai đối thủ được mô tả như sau: Xuất phát từ bảng trò chơi là một bảng vuông kích thước $n \times n$ gồm n dòng và n cột. Các dòng của bảng được đánh số từ 1 đến n , từ trên xuống dưới. Các cột của bảng được đánh số từ 1 đến n , từ trái sang phải. Trên mỗi ô của bảng ghi một số nguyên. Hai đối thủ luân phiên thực hiện nước đi. Đối thủ đến lượt chơi của mình được phép xóa dòng cuối cùng nếu tổng các số trên dòng đó là số chẵn hoặc là cột cuối cùng nếu tổng các số trên cột đó là số chẵn.

Đối thủ thắng cuộc là người xóa được ô cuối cùng của bảng hoặc sau khi thực hiện nước đi của mình thì tổng các số trên dòng cuối cùng và tổng các số trên cột cuối cùng của bảng đều là số lẻ.

Yêu cầu: Cho biết bảng số của trò chơi, hãy xác định xem người đi trước có cách chơi giành phần thắng hay không?

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Dữ liệu:

- Dòng thứ nhất chứa số nguyên dương k là số lượng bộ dữ liệu.
- Tiếp theo là k nhóm dòng, mỗi nhóm dòng tương ứng với một bộ dữ liệu có dạng:
- Dòng thứ nhất chứa số nguyên dương n ($n \leq 500$).
- Dòng thứ i trong số n dòng tiếp theo chứa n số nguyên dương (mỗi số không vượt quá 10^9) là các số trên dòng thứ i của bảng trò chơi, $i = 1, 2, \dots, n$.
- Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra k dòng, mỗi dòng là kết quả tương ứng với một bộ dữ liệu theo thứ tự xuất hiện trong file dữ liệu vào: ghi thông báo **“YES”** nếu người đi trước có cách chơi giành chiến thắng và **“NO”** trong trường hợp ngược lại.

Ví dụ:

Dữ liệu			Kết quả
2			YES
3			NO
1	2	2	
1	2	3	
2	3	1	
4			
2	2	2	2
2	2	2	2
2	2	2	2
2 2 2 2			

Ràng buộc: 50% số test ứng với 50% số điểm của bài có $n \leq 50$.

Solution:

- Gọi $a[i,j]=\text{true/false}$ là thắng hay thua nếu nước đi tiếp theo là người 1 và bảng đang chơi là bảng $[1,1 \rightarrow i,j]$.
- Gọi $b[i,j]=\text{true/false}$ là thắng hay thua nếu nước đi tiếp theo là người 2 và bảng đang chơi là bảng $[1,1 \rightarrow i,j]$.

Dễ thấy công thức Qhđ : $\begin{cases} a[i,j] = \text{not } b[i,j-1] \text{ or } \text{not } b[i-1,j] \\ b[i,j] = \text{not } a[i,j-1] \text{ or } \text{not } a[i-1,j] \end{cases}$. Thực tế mảng a, b là như nhau, chỉ trình bày thế cho dễ hiểu, nên khi code chỉ cần dùng một mảng a thôi.

Bài 17: TRÒ CHƠI CHẮN LỀ VER2

Mã bài: PARIGAME.???

Trò chơi chắn lẻ là trò chơi hai đối thủ được mô tả như sau: Cho một bảng số kích thước $m \times n$ gồm m dòng và n cột. Các dòng của bảng được đánh số từ 1 đến m , từ trên xuống dưới. Các cột của bảng được đánh số từ 1 đến n , từ trái qua phải. Trên mỗi ô của bảng ghi một số nguyên. Hai đối thủ luân phiên thực hiện nước đi. Đối thủ đến lượt chơi của mình được phép xóa đi một số dòng cuối cùng nếu tổng các số trên các dòng đó là số chẵn hoặc là xóa đi một số cột cuối cùng nếu tổng các số trên các cột đó là số chẵn. Đối thủ thắng cuộc là người xóa được ô cuối cùng của bảng hoặc sau khi thực hiện nước đi của mình thì đối phương không thể thực hiện được nước đi.

Yêu cầu: Cho biết bảng số của trò chơi, hãy xác định xem người đi trước có cách chơi giành phần thắng hay không?

Input

- Dòng thứ nhất chứa số nguyên dương T là số lượng bộ dữ liệu;
- Tiếp theo là T nhóm dòng, mỗi nhóm dòng tương ứng với một bộ dữ liệu có dạng:
 - Dòng thứ nhất chứa ba số nguyên dương m, n ;
 - Dòng thứ i trong số m dòng tiếp theo chứa n số nguyên dương (mỗi số không vượt quá 10^9) là các số trên dòng thứ i của bảng trò chơi, $i = 1, 2, \dots, m$.

Output

- Gồm T dòng, mỗi dòng là kết quả tương ứng với một bộ dữ liệu theo thứ tự xuất hiện trong file dữ liệu vào: ghi thông báo 'YES' nếu người đi trước có cách chơi giành phần thắng và 'NO' trong trường hợp ngược lại.

PARIGAME.INP	PARIGAME.OUT
2	NO
2 2	YES
5 2	
4 2	
2 2	
1 1	
1 1	

Subtask 1: $m, n \leq 50$;

Subtask 2: $m, n \leq 500$.

Solution:

Sub1: Làm tương tự bài PARIGAME VER1, thay vì xóa 1 dòng hoặc một cột ta đi for từ 1-> k để xóa k dòng hoặc k cột.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Sub 2:

Để tính nhanh tổng ta dùng mảng $s[i,j] = \sum \text{bảng } 1,1 \rightarrow i,j$.

- Gọi $a[i,j] = \text{true/false}$ là thắng hay thua nếu nước đi tiếp theo là người 1 và bảng đang chơi là bảng $[1,1 \rightarrow i,j]$.
- Gọi $b[i,j] = \text{true/false}$ là thắng hay thua nếu nước đi tiếp theo là người 2 và bảng đang chơi là bảng $[1,1 \rightarrow i,j]$.

Để thấy công thức Qhđ: $\begin{cases} a[i,j] = \text{not } b[i,j-k] \text{ or } \text{not } b[i-k,j] \\ b[i,j] = \text{not } a[i,j-k] \text{ or } \text{not } a[i-k,j] \end{cases}$. Ở đây ta nhận thấy để tổng chẵn thì phải là hiệu 2 số lẻ hoặc hiệu 2 số chẵn, do vậy ta sẽ lưu lại vị trí chẵn, lẻ mà tạo false đối với người đối thủ.

Bài 18: CON SỐ

Mã bài: NUMBER.???

Cho số nguyên dương n , hãy tìm số nguyên dương m nhỏ nhất để $n \times m$ là một số mà trong biểu diễn của nó chỉ gồm một loại chữ số.

Ví dụ: $n = 37$ thì $m = 3$ vì $37 \times 3 = 111$.

Input

- Gồm nhiều dòng, mỗi dòng chứa một số nguyên dương n ($n \leq 10^6$).

Output

- Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào, chứa một số nguyên dương m tìm được, nếu không tồn tại m ghi -1.

NUMBER.INP	NUMBER.OUT
37	3

Solution:

Ta có $n \times m = ddd\dots dddd = d \times 111\dots 1111$ (k số 1). Ta thấy d thuộc $[1,9] \Rightarrow$ có thể chạy d .

```
for d := 1 to 9 do
begin
for k := 1 to ??? do
begin
s := s * 10 + d;
if s mod n = 0 thì ghi nhận kết quả cho m.
end;
end;
```

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Nhận xét:

- Để s không vượt quá giới hạn ta mod liên tục cho n , tức $s := (s \cdot 10 + d) \bmod n$.
- Để m là nhỏ nhất thì $k_1 > k_2 > \dots > k_9$ (với k_i ứng với $d=i$). do vậy dấu ??? sẽ là k cũ tức là k nhỏ nhất tồn tại m .

Bài 19: ĐA GIÁC

Mã bài: POLY.???

Trên mặt phẳng tọa độ, xét đa giác lồi n đỉnh, các đỉnh của đa giác được liệt kê theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ.

Yêu cầu: Cho đoạn thẳng xác định bởi hai điểm có tọa độ là (x_1, y_1) và (x_2, y_2) trong đó x_1, x_2, y_1, y_2 là các số nguyên và có giá trị tuyệt đối không vượt quá 1000. Hãy xác định độ dài L là phần của đoạn thẳng nằm trong đa giác hay trên cạnh của đa giác.

Input

- Dòng đầu tiên chứa số nguyên ($3 \leq n \leq 100$);
- Dòng thứ i trong n dòng sau chứa 2 số nguyên xác định tọa độ đỉnh i của đa giác;
- Dòng cuối cùng chứa bốn số nguyên x_1, y_1, x_2, y_2 .

Output

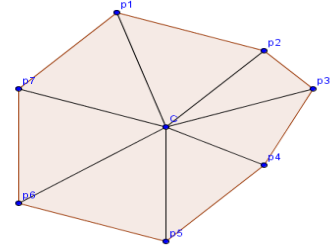
- Ghi một số nguyên của tích $L \cdot 100$.

POLY.INP	POLY.OUT
4 0 1 1 0 0 -1 -1 0 -2 0 0 0	100

Solution:

Ta thấy bài trên có khá nhiều cách làm, nhưng đối với bài này do cần độ chính xác không cao lên ta có giải thuật sau.

- Gọi C là trung điểm đoạn AB . P là đa giác.
- $AB \cap P = AC \cap P + CB \cap P$. \Rightarrow Xây dựng hàm đệ quy để tính kq .
- Cách kiểm tra điểm C có thuộc P hay không ta kiểm tra bằng diện tích.
- $\sum \Delta nhỏ = S_{đa\ giac}$



Bài 20: KHU ĐẤT

Mã bài: LAND.???

Một khu đất kích thước $m \times n$. Theo quy hoạch ban đầu, trên khu đất đó sẽ xây dựng k khu vui chơi, khu vui chơi thứ i sẽ là một khu vực với kích thước $a_i \times b_i$. Tuy nhiên, một vấn đề mới phát sinh, người ta muốn quy hoạch lại để có thêm một khu vui chơi mới có dạng hình vuông và hình vuông có diện tích càng lớn càng tốt. Các khu vui chơi được xây dựng có thể tiếp xúc với nhau nhưng không được giao nhau.

Yêu cầu: Cho m, n và k khu vui chơi, hãy tính diện tích khu đất hình vuông lớn nhất tìm được.

Input

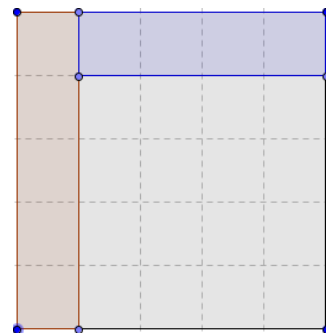
- Dòng đầu ghi 3 số nguyên m, n, k ($m, n \leq 10; k \leq 5$);
- k dòng sau, mỗi dòng ghi hai số nguyên a_i, b_i .

Output

Gồm một dòng ghi một số nguyên là diện tích khu đất hình vuông lớn nhất tìm được.

LAND.INP	LAND.OUT
5 5 2 1 5 1 4	16

Solution:



+ Từ bài toán tối ưu ta đưa về bài toán thỏa được.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

+ Chọn luôn hình vuông và coi là đó là hình chữ nhật thứ $k+1$ cần xếp sao cho không có 2 hình nào giao nhau \Rightarrow cạnh hình vuông $l_{\max} = \sqrt{(m \cdot n - \sum s_i)}$.

+ Thử cạnh hình vuông từ l_{\max} về 0.

- $a[k+1] := l$;
- $b[k+1] := l$;
- Xếp (1);

Tổng quan về thủ tục Xếp như sau:

```
1  Procedure Xếp(i:longint);
2  var      x,y:longint;
3  begin
4      if i>k+1 then => KQ
5      for x=1 -> m do
6          for y=1 -> n do
7              begin
8                  if Đặt được(a[i],b[i],x,y) then
9                      begin
10                         Đặt(a[i],b[i],x,y);
11                         Xếp(i+1);
12                         Đặt lại(a[i],b[i],x,y);
13                     end;
14                  if Đặt được(b[i],a[i],x,y) then
15                      begin
16                         Đặt(b[i],a[i],x,y);
17                         Xếp(i+1);
18                         Đặt lại(b[i],a[i],x,y);
19                     end;
20              end;
21  end;
```

Để thủ tục xếp trên nhanh chóng được thực hiện thì ta sẽ tạo ra xung đột càng sớm càng tốt.

- Sort lại giảm dần theo diện tích trước khi gọi thủ tục Xếp(i).

Note: Nên cẩn thận trong việc sắp xếp các diện tích, khi thêm một hình chữ nhật thì ta nhớ bỏ hình chữ nhật cũ ra, thêm hcn mới vào và sắp xếp lại.

Bài 21: TẠO TEST

Mã bài: TEST.???

Bài toán “Số trung bình” được phát biểu như sau:

Cho dãy số nguyên không âm a_1, a_2, \dots, a_n hãy đếm số bộ ba chỉ số $i < k < j$ sao cho $a_i + a_k = 2 * a_j$.

An đã tạo ra được dãy số nguyên không âm a_1, a_2, \dots, a_n , nhưng Bình muốn sắp xếp lại dãy số để dãy mới nhận được có số bộ ba chỉ số $i < k < j$ sao cho $a_i + a_k = 2 * a_j$ là ít nhất.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Yêu cầu: Cho dãy số nguyên không âm a_1, a_2, \dots, a_n , hãy sắp xếp lại dãy số để dãy mới nhận được có số bộ ba chỉ số $i < k < j$ sao cho $a_i + a_k = 2 * a_j$ là ít nhất.

Input

Dòng đầu tiên ghi số nguyên dương T ($T \leq 10$) là số lượng bộ dữ liệu. Tiếp đến là T nhóm dòng, mỗi nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai gồm n số a_1, a_2, \dots, a_n ($a_i \leq 1000$).

Output

- Gồm T dòng, mỗi dòng chứa số là hoán vị của $1, 2, \dots, n$, mô tả thứ tự các phần tử của dãy số sau khi sắp xếp tương ứng với bộ dữ liệu vào.

TEST.INP	TEST.OUT
2	1 3 2
3	1 2 3
1 2 3	
3	
1 1 1	

Subtask 1: $n \leq 8$; [25 tests]

Subtask 2: $n \leq 10$; [25 tests]

Subtask 3: $n \leq 1000$; $a_i = i$; [25 tests]

Subtask 4: $n \leq 1000$ [25 tests]

Solution:

Note: Các $0 \leq a[i] \leq 1000$.

Sub 1+2 : Duyệt hoán vị kết hợp mảng đánh dấu.

Thủ tục duyệt sẽ như sau:

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

```
28 Procedure Sinh(k,sl:longint);
29 var i,j,slc,ii:longint;
30 begin
31     slc:=sl;
32     for i:=1 to n do
33         if cx[i] then
34             begin
35                 cx[i]:=false;
36                 x[k]:=i;
37                 dec(dem[a[i]]);
38                 sl:=slc;
39                 for j:=1 to k-1 do sl:=sl+dem[2*a[i]-a[x[j]]];
40                 if (k<n) and (sl<res) then Sinh(k+1,sl) else
41                     if sl<res then
42                         begin
43                             res:=sl;
44                             for ii:=1 to n do xb[ii]:=x[ii];
45                         end;
46                 cx[i]:=true;
47                 sl:=slc;
48                 inc(dem[a[i]]);
49             end;
50 end;
```

Kết quả là mảng xb.

Sub 3: Các số là số thứ tự từ 1->n.

Ta thấy nếu $a_i + a_j$ lẻ => không tồn tại a_k

- Ta sẽ bóc chẵn sang một bên, bóc lẻ sang một bên chẵn lẻ là vị trí chứ không phải là giá trị.

Để ví dụ cho đơn giản.

Giả sử $n=7$ => Dãy số là 1, 2, 3, 4, 5, 6, 7.

1. 1, 3, 5, 7 | 2, 4, 6.
2. 1, 5 | 3, 7 || 2, 6 | 4.

Dãy số thu được sau khi bóc là kết quả cần tìm.

Sub 4:

+ Các số đôi một khác nhau ta có thể làm tương tự như Sub 3 bằng việc đánh số lại theo thứ tự, nếu như bằng nhau ta gom chúng vào làm 1 và cho chúng đứng cạnh nhau sau khi đã sắp xếp.

Do $n \leq 1000$ nên ta sẽ chia luôn dãy các số từ 0->1000. Nên xây dựng thủ tục đệ quy chia(i,j) tức là chia đoạn từ i -> j trên dãy số.

Bài 22: TÍNH TỔNG

Mã bài: SUM2N.???

Yêu cầu: Cho số nguyên dương n, K , tính $S = (1 \cdot 2^0 + 2 \cdot 2^1 + \dots + n \cdot 2^{n-1}) \bmod K$

Input

- Gồm nhiều bộ dữ liệu, mỗi bộ trên một dòng, mỗi dòng chứa hai số nguyên dương n, K .

Output

- Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào.

SUM2N.INP	SUM2N.OUT
1 10	1
2 2	1

Subtask 1: $n, k \leq 10^9$

Subtask 2: $n, k \leq 10^{18}$

Solution:

Công thức rút gọn: $S = (n - 1) * 2^n \bmod K$.

Có thể sử dụng một trong 3 thuật toán sau để tính $a * b \bmod p$ với $a, b, p \in \text{int64}$.

1. Nhanh nhất, song khó code nhất.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

```
37. Function Mul_mod(a,b,p:int64):int64;
38. var i,j:longint;
39. t,jj,c32:Qword;
40. c0,c1,b0,b1,t1:Qword;
41. begin
42. c32:=Qword(1) shl 32;
43. c1:=a shr 32;
44. c0:=a mod c32;
45. b0:=b mod c32;
46. b1:=b shr 32;
47. t:=c0*b0; t1:=t shr 32; d[0]:=t mod c32;
48. t:=(c0*b1+c1*b0)+t1;
49. t1:=t shr 32; d[1]:=t mod c32;
50. t:=c1*b1+t1;
51. d[2]:=t mod c32; d[3]:=t shr 32;
52. t:=(d[3] shl 32 or d[2]) mod p;
53. for i:=1 downto 0 do
54. for j:=7 downto 0 do
55. begin
56. jj:=j*4;
57. t1:=(d[i] shr jj and 15);
58. t:=(t shl 4) or t1;
59. t:=t mod p;
60. end;
61. exit(t);
62. end;
```

2. Ngắn nhất, hiệu quả nhất, song chậm hơn cách 1 một ít time.

```
37. Function Mul_mod(a,b,p:int64):int64;
38. var tg:int64;
39. begin
40. if b=0 then exit(0);
41. if b=1 then exit(a mod p);
42. tg:=Mul_mod(a,b shr 1,p);
43. tg:=tg+tg;
44. if (b mod 2=0) then exit(tg mod p) else exit((tg+a) mod p);
45. end;
```

3. Cách dễ hiểu nhất. Cài số lớn. hì hì.

Bài 23: TÍNH TỔNG BÌNH PHƯƠNG

Mã bài: SUM.???

Viết chương trình đọc vào hai số thực dương A và B và tính tổng bình phương tất cả các số nguyên không nhỏ hơn A và không lớn hơn B.

Dữ liệu: Vào từ file văn bản SUM.INP gồm một dòng chứa hai số thực dương A, B.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Kết quả: Đưa ra file văn bản SUM.OUT gồm một dòng chứa một số nguyên là phần dư của số S chia cho 10^9+7 , trong đó S là tổng bình phương tất cả các số nguyên không nhỏ hơn A và không lớn hơn B.

Ví dụ:

SUM.INP	SUM.OUT
0.3 2.89	5

Note:

- Có 50% số test có $0 < A \leq B \leq 1000$;
- Có 50% số test còn lại có $0 < A \leq B \leq 10^9$.

Solution:

Ta có $S = 1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$. Để tránh cho việc cài div số lớn ta thấy trong 3 số n, n+1, 2n+1 sẽ có ít nhất 1 số chia hết cho 2, một số chia hết cho 3, do vậy ta chỉ việc rút gọn và vừa nhân vừa mod.

Bài 24: TÍNH TỔNG 2

Mã bài: SUM2.???

Viết chương trình đọc vào số nguyên dương n và tính tổng sau:

$S = \left\lfloor \frac{n}{1} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + \dots + \left\lfloor \frac{n}{n} \right\rfloor$, trong đó $\lfloor x \rfloor$ là số nguyên lớn nhất không vượt quá x.

Input

- Dòng đầu chứa số nguyên T ($T \leq 30$) là số lượng bộ test;
- T dòng sau, mỗi dòng chứa một số nguyên dương n ($n \leq 10^{12}$).

Output

- Gồm T dòng, mỗi dòng chứa một số nguyên là phần dư của số S chia 10^6 tương ứng với bộ dữ liệu vào.

SUM2.INP	SUM2.OUT
2	1
1	10
5	

Solution:

Hàm tính kết quả.

```

9  Function Calc(n:int64):int64;
10  var      i,j:int64;
11           v:int64;
12           sum:int64;
13  begin
14           i:=1;
15           sum:=0;
16           while i<=n do
17           begin
18               v:=n div i;
19               j:=n div v;
20               if j>n then j:=n;
21               sum:=(sum+(j-i+1)*v) mod e;
22               i:=j+1;
23           end;
24           Calc:=sum;
25  end;

```

Bài 25: NHÂN MA TRẬN

Mã bài: MMC.???

Phép nhân hai ma trận chỉ thực hiện được khi số cột của ma trận bên trái bằng số dòng của ma trận bên phải. Nếu ma trận A có kích thước $m \times n$ và ma trận B có kích thước $n \times p$, thì ma trận tích $C = A \cdot B$ có kích thước $m \times p$, phần tử đứng ở hàng thứ i , cột thứ j xác định bởi:

$$c_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \dots + a_{i,n}b_{n,j}$$

Phép nhân ma trận có các tính chất kết hợp: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$;

Ví dụ,

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; \quad A^2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}; \quad A^3 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

Yêu cầu: Cho ma trận A kích thước $n \times n$ và ma trận B, hãy kiểm tra xem A^3 có bằng B hay không?

Input

- Dòng thứ nhất chứa số nguyên dương T ($T \leq 20$) là số lượng bộ dữ liệu;
- Tiếp theo là T nhóm dòng, mỗi nhóm dòng tương ứng với một bộ dữ liệu có dạng:
 - Dòng đầu chứa số nguyên ;
 - n dòng tiếp theo, mỗi dòng chứa số nguyên mô tả ma trận, các số có giá trị tuyệt đối không vượt quá 1000;
 - n dòng tiếp theo, mỗi dòng chứa số nguyên mô tả ma trận, các số có giá trị tuyệt đối không vượt quá 10^{18} ;

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Output

- Gồm T dòng, mỗi dòng là kết quả tương ứng với một bộ dữ liệu theo thứ tự xuất hiện trong file dữ liệu vào: ghi thông báo 'YES' nếu $A^3 = B$ và ghi 'NO' trong trường hợp ngược lại.

MMC.INP	MMC.OUT
2	NO
2	YES
0 1	
1 1	
1 2	
2 2	
2	
0 1	
1 1	
1 2	
2 3	

Subtask 1: $n \leq 10$

Subtask 2: $n \leq 500$

Solution:

Sub 1: Trâu theo đề.

Sub 2: Sử dụng tính chất sau $A=B \Rightarrow A*C=B*C$. Do đó $A^3 = B \Rightarrow A * (A * (A * C)) = B * C$ với C là ma trận $n \times 1$ được sinh bằng cách random.

Bài 26: Dãy FIBO BỤ

Mã bài: FIBO.???

Số Fibonacci được định nghĩa như sau:

$$Fibo_i = \begin{cases} a & \text{nếu } i = 1 \\ b & \text{nếu } i = 2 \\ Fibo_{i-2} + Fibo_{i-1} & \text{nếu } i = 3 \end{cases}$$

Yêu cầu: Cho 4 số nguyên n, F, L, R ($0 \leq L \leq R \leq 100, n \geq 3$) hãy kiểm tra xem có tồn tại a, b mà $L \leq a, b \leq R$ để $Fibo_n = F$.

Input

- Dòng thứ nhất chứa số nguyên dương T là số lượng bộ dữ liệu;
- Tiếp theo là T nhóm dòng, mỗi nhóm gồm 4 dòng: dòng thứ nhất chứa số n, L, R, dòng sau chứa số F ($F \leq 10^{30000}$).

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Output

- Gồm T dòng, mỗi dòng là kết quả tương ứng với một bộ dữ liệu theo thứ tự xuất hiện trong file dữ liệu vào: ghi thông báo 'YES' tồn tại a,b mà $L \leq a, b \leq R$ để $Fibo_n = F$. và ghi 'NO' trong trường hợp ngược lại.

LRF.INP	LRF.OUT
2	YES
3 0 10	NO
1	
3 0 10	
21	

Subtask 1: $n \leq 100$;

Subtask 2: $n \leq 1000$;

Subtask 3: $n \leq 100000$.

Solution:

Sử dụng tính chất sau. $A=B \Rightarrow A \bmod C = B \bmod C$.

Đây là bài toán cơ bản của nhân ma trận với ma trận cơ sở $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$.

Bài 27: ĐẢO DỪA

Mã bài: VMYT.???

yenthanh132 vừa trúng vé số, anh ta định mua một hòn đảo để xây nhà nghỉ mát. Thế là anh ta để ý ngay đến vương quốc đảo dừa trù phú của Pirate vĩ đại. Sau cuộc nội chiến dừa và vụ "tam phân thiên hạ" lộn xộn vừa rồi thì hiện vương quốc dừa đã được bình yên với sự cai trị của vị vua mới - Pirakute (Xem bài [VMAOCE](#) để biết thêm chi tiết). Do rất yêu quê hương đảo nước của mình nên Pirakute đã một mực không chịu bán bất cứ hòn đảo nào cho *yenthanh132*, tuy nhiên nhờ sự đàm phán lâu dài cùng với mức giá kết xù của *yenthanh132* đưa ra để mua đảo nên Pirakute sau khi suy nghĩ một thời gian đã đưa ra quyết định. Pirakute sẽ đồng ý bán cho *yenthanh132* một hòn đảo với điều kiện anh ta phải tính được số hòn đảo hiện có trong vương quốc dừa.

Do sau một thời gian dài nội chiến nhiều thông tin đã bị mất, không ai thống kê lại xem hiện thời vương quốc dừa có tổng cộng bao nhiêu đảo. Thay vào đó chỉ còn lại một tài liệu cổ nói về lịch sử của vương quốc dừa từ thuở khai thiên lập địa. Tài liệu nói rằng từ thời xa xưa, cụ tổ Pirate đã xây dựng từng hòn đảo bằng cách chở dừa đến và chất thành đồng ở giữa biển, các trái dừa này sẽ mọc ra những cây dừa và thế là một hòn đảo dừa được tạo ra. Lịch sử nói rằng K năm đầu tiên là thời kì khó khăn nhất của vương quốc dừa. Do có một số hòn đảo chất lượng không tốt nên bị mất đi và việc Pirate kiên trì tạo thêm các hòn đảo mới nên số

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

lượng các hòn đảo qua K năm đầu tiên lên xuống thất thường nhưng luôn có ít nhất một đảo. Số lượng hòn đảo trong K năm đầu được lịch sử ghi lại trong dãy số $a[1], a[2], \dots, a[K]$. Sau K năm đó, tức là kể từ năm $K+1$, các cây dừa trên hòn đảo đã trưởng thành và cho trái, các trái dừa rớt xuống biển và tự động tạo nên các hòn đảo dừa mới một cách tự nhiên. Số lượng đảo dừa từ đó tăng một cách chóng mặt và vương quốc dừa trở thành một đại cường quốc. Lịch sử cũng nói lại rằng số lượng đảo dừa trong năm thứ i sẽ bằng tích số lượng đảo dừa của K năm trước. Tức là số lượng đảo dừa của vương quốc trong năm i bằng $a[i] = a[i-1] * a[i-2] * \dots * a[i-K]$. Mặc dù vương quốc đã trải qua nhiều thăng trầm, chiến tranh liên miên nhưng số lượng đảo dừa qua từng năm vẫn tăng đều theo công thức đó.

Thế là có đủ thông tin để *yenthanh132* có thể tính được số lượng đảo dừa của vương quốc dừa. Nhưng tính tới thời điểm hiện tại đã là năm tồn tại thứ N của vương quốc và N là một số rất lớn nên không ai có thể tính được. *yenthanh132* định dùng siêu máy tính mà anh ta đã mua hôm nọ ([TNHTEST](#)) nhưng khổ nỗi vương quốc dừa dùng điện năng lượng mặt trời nên không có đủ điện để chạy siêu máy tính được. Thế là giờ chỉ còn mỗi cái Laptop Pentium 3 800Mhz là có thể dùng được. Bạn là một nhà lập trình viên siêu hạng, hãy giúp *yenthanh132* và chứng tỏ rằng Laptop cùi tới mấy mà biết sử dụng thì cũng có thể làm nên chuyện :)

Yêu cầu:

- Cho số N, K và dãy số $a[1], a[2], \dots, a[K]$ - số đảo dừa trong vương quốc qua K năm đầu
- Với $i > K$ ta có $a[i] = a[i-1] * a[i-2] * a[i-3] * \dots * a[i-K]$;
- Tính $a[N]$, do số lượng hòn đảo rất lớn nên *yenthanh132* chỉ yêu cầu bạn tính phần dư của kết quả khi chia cho số 1000000007 ($= 10^9 + 7$). Tức là tính $a[N] \% 1000000007$

Lưu ý: Những đường dẫn đến bài tập bên ngoài chỉ mang tính chất tham khảo cho vui, không liên quan đến bài này.

Input

- Dòng đầu tiên gồm hai số nguyên dương N, K .
- Dòng thứ hai ghi K số nguyên dương, số thứ i của dãy số là A_i

Giới hạn

- $1 \leq N \leq 10^9$
- $1 \leq K \leq 50$ và $K < N$.
- $1 \leq A_i \leq 10^9$
- 30% số test có $N \leq 100000$

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Output

- Ghi ra kết quả theo modulo $1000000007 (10^9 + 7)$.

Chấm bài

Bài của bạn sẽ được chấm trên thang điểm 100. Điểm mà bạn nhận được sẽ tương ứng với % test mà bạn giải đúng.

Trong quá trình thi, bài của bạn sẽ chỉ được chấm với 1 test ví dụ có trong đề bài.

Khi vòng thi kết thúc, bài của bạn sẽ được chấm với bộ test đầy đủ.

Example

Input:

7 3
1 2 3

Output:

139968

Solution:

- Công thức cần biết : $a^{p-1} \bmod p = 1$ với p là số nguyên tố và a không chia hết cho p ;
- Thấy rằng có thể phân tích mọi số trong dãy thành:
 - $A_i = A_1^{E_{i,1}} * A_2^{E_{i,2}} * \dots * A_k^{E_{i,k}}$ (E_1, E_2, \dots, E_k là các số nguyên không âm). Vậy nên thay vì tính A , ta đi tính E .
- Do công thức:
 - $A_i = A_{i-1} * A_{i-2} * \dots * A_{i-k}$.
- Nên ta có được:
 - $E_{i,j} = E_{i-1,j} + E_{i-2,j} + \dots + E_{i-k,j} (1 \leq j \leq k)$.

Ma trận cơ sở c:

```
26. Procedure Init;  
27.   var    i,j:longint;  
28.   begin  
29.       fillchar(a,sizeof(a),0);  
30.       fillchar(c,sizeof(c),0);  
31.       j:=1;  
32.       for i:=1 to k do begin a[i,i]:=1; c[i,k]:=1; end;  
33.       for i:=2 to k do begin c[i,j]:=1; inc(j); end;  
34.   end;
```

Bài 28: GIÁ SÁCH

Mã bài: BSF.???

Bờm có N quyển sách, quyển sách thứ i có chiều cao h_i và chiều rộng w_i . Bờm muốn xây dựng một số giá sách để chứa hết tất cả n quyển sách này. Qua tìm hiểu, Bờm nhận được các thông tin sau: Nhà sản xuất nhận làm m loại giá sách, mỗi loại giá sách gồm các thông tin: H_i, F_i, C_i . Giá sách loại thứ i có thể chứa được các quyển sách có độ cao không vượt quá H_i và nếu muốn dựng giá sách có độ rộng là W thì giá tiền tương ứng là: $F_i + W \times C_i$.

Yêu cầu: Cho thông tin về các quyển sách và các loại giá sách, hãy giúp Bờm tính chi phí ít nhất để dựng một số giá sách chứa tất cả các quyển sách.

Input

- Dòng 1: gồm 2 số n, m .
- Dòng 2 đến dòng $n + 1$, mỗi dòng chứa 2 số nguyên dương mô tả chiều cao h_i và chiều rộng w_i của quyển sách ($h_i, w_i \leq 10^9$).
- Dòng thứ $n + 2$ đến dòng $n + m + 1$, mỗi dòng chứa 3 số nguyên dương H_i, F_i, C_i ($H_i, F_i, C_i \leq 10^9$) mô tả các thông tin về các loại giá sách.

Output

- Gồm một dòng chứa một số là chi phí ít nhất để dựng một số giá sách chứa tất cả các quyển sách.

BFS.INP	BFS.OUT
3 3	1680
20 5	
21 10	
22 5	
20 100 1	
21 150 2	
25 1000 100	

Subtask 1: $N \leq 20; M \leq 2$ [25 tests]

Subtask 2: $N \leq 1000; M \leq 10$ [25 tests]

Subtask 3: $N \leq 100; M \leq 100$ [25 tests]

Subtask 4: $N \leq 1000; M \leq 1000$ [24 tests]

Solution:

Sub 2:

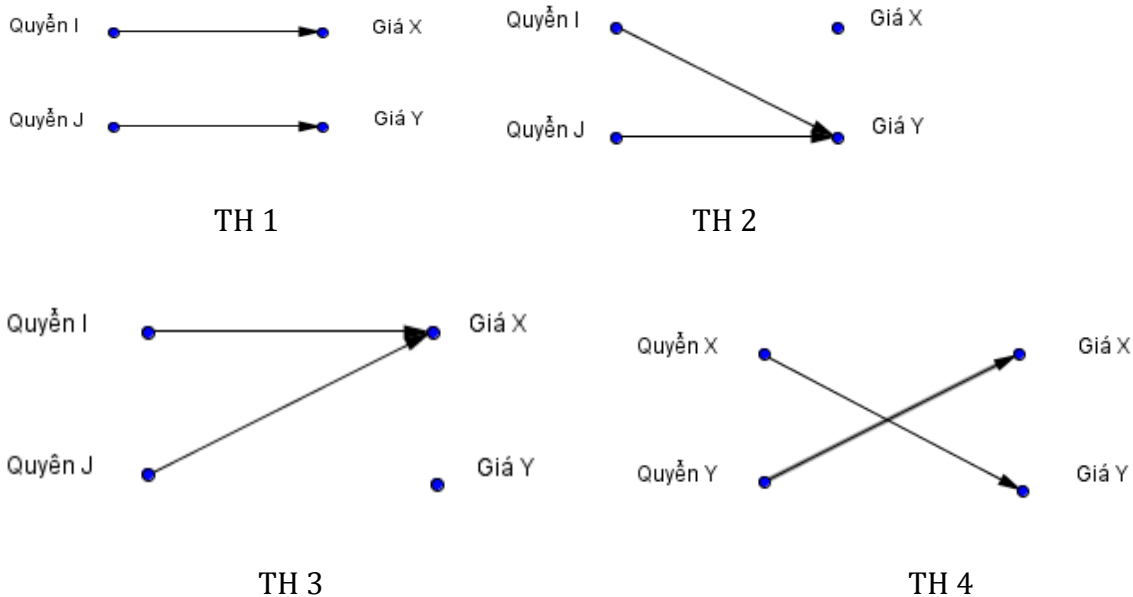
CHUYÊN ĐỀ TỔNG HỢP 2013-2014

- Duyệt nhị phân giá nào được dùng, giá nào không được dùng.
- Nhét sách vào giá tạo chi phí min, do là duyệt nên luôn tìm được phương án tối ưu.

Sub 3:

Nhận xét:

Ta thấy để tạo chi phí min, thì giả sử quyển I > quyển J, giá X > giá Y thì chỉ có thể xảy ra một trong 3 TH sau:



Do TH 4 không xảy ra lên ta có thể QHĐ.

Gọi $F[i, j]$: Từ 1->i chọn ra một số giá sách để nhét hết j quyển với chi phí nhỏ nhất.

$$F[i, j] = \min \begin{cases} F[i-1, j-1] \\ F[i-1, j'] + f[i] + \sum W_{j' \rightarrow j} * C[i] \end{cases}$$

Sub 4: Cải tiến từ Sub 3:

Gọi $F[i, j, k]$ với ý nghĩa như trên. $\begin{cases} k = 0 : \text{Giá thứ } i \text{ chưa được dùng.} \\ k = 1 : \text{Giá thứ } i \text{ đã được dùng.} \end{cases}$

- $F[i, j, 0] = \min(F[i-1, j, 0]; F[i-1, j, 1]);$
- $F[i, j, 1] = \min(F[i-1, j, 1] + w[j] * c[i]; F[i-1, j, 0] + f[i]).$

Bài 29: BOWLING

Mã bài: BOWLING.???

Bowling là một trò chơi giải trí mà người chơi ném một quả bóng nặng cho chạy trên một đường băng dài và phẳng để làm đổ những chai gỗ đứng ở cuối đường. Ngày nay, Bowling được xem là một môn thể thao. Trong bài toán này chúng ta sẽ xét trò chơi Bowling cải biên như sau:

- Cuối đường băng người ta đặt n chai gỗ được xếp thành một hàng ngang, các chai gỗ được đánh số từ 1 đến n từ trái qua phải. Chai gỗ thứ i ghi số nguyên a_i tương ứng là điểm thưởng (nếu $a_i \geq 0$) hoặc phạt (nếu $a_i < 0$) khi ném bóng mà làm đổ chai gỗ này.
- Người chơi phải ném ít nhất một lần và không giới hạn số lần ném bóng. Mỗi lần ném bóng, người chơi sẽ ném bóng hướng vào một trong n vị trí đặt chai gỗ, nếu ném bóng hướng vào vị trí đặt chai gỗ thứ i thì nó sẽ làm đổ những chai đặt ở vị trí có khoảng cách với vị trí chai thứ i không vượt quá r . Khoảng cách giữa vị trí hai chai thứ i và thứ j được tính là $|i - j|$. Tổng điểm mà người chơi đạt được là tổng các số ghi trên các chai gỗ mà người chơi làm đổ được. Muốn đạt được nhiều điểm người chơi không những phải có khả năng thực hiện việc ném bóng chính xác mà còn phải biết lựa chọn hướng ném bóng trong mỗi lượt chơi.

Yêu cầu: Cho r và a_1, a_2, \dots, a_n , hãy tính tổng điểm lớn nhất mà người chơi có thể đạt được với

giả thiết người chơi có khả năng thực hiện chính xác việc ném bóng.

Input

Dòng đầu tiên ghi số nguyên dương K là số lượng bộ dữ liệu. Tiếp đến là K nhóm dòng, mỗi

nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng thứ nhất ghi hai số nguyên dương n và r ($r \leq n$);
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n , số a_i tương ứng là số ghi trên chai gỗ thứ i ($|a_i| \leq 10^9$).

Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Output

Gồm K dòng, mỗi dòng ghi một số nguyên là tổng điểm mà người chơi có thể đạt được tương ứng với bộ dữ liệu trong dữ liệu vào.

BOWLING.INP	BOWLING.OUT
3	2
5 1	5

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

1 0 -10 0 1 5 1 1 1 1 1 1 5 1 -1 -1 -1 -1 -1	-2
--	----

Subtask 1: Giả thiết là $n \leq 20$. [2 tests]

Subtask 2: Giả thiết là $n \leq 2000$. [4 tests]

Subtask 3: Giả thiết là $n \leq 200000$. [4 tests]

Solution:

Note : Nếu đọc đề không cẩn thận sẽ hiểu sai đề, hoặc khó làm.

Bài này ta hiểu đơn giản là chọn một số đoạn con có độ dài $\geq 2*r$ sao cho tổng đạt max.

Sub 1: Duyệt nhị phân 0/1 tương ứng với việc chọn hay không chọn số đó. Có thể thêm cận là độ dài

Sub 2: Gọi $F[i]$ là kết quả khi xét dãy từ 1 \rightarrow i.

- $F[i,j] = \begin{cases} F[i-1] \\ F[i'] + \sum i' + 1 \rightarrow i \end{cases} \quad (i' \rightarrow i \text{ có } \geq 2 * r \text{ phần tử})$

Sub 3: Gọi $F[i,k]$ ý nghĩa tương tự $\begin{cases} k = 0 \text{ chưa chọn } i. \\ k = 1 \text{ chọn } i. \end{cases}$

- $f[i,0] := \text{Max}(f[i-1,0], f[i-1,1]);$
- $f[i,1] := \text{Max}(f[i-1,1] + a[i], \text{Max}(f[i-2*r-1,0] + \sum i - 2r \rightarrow i, \sum i - 2r \rightarrow i));$

Bài 30: QUÂN HẬU

Mã bài: NQUEEN.???

An rất thích thú với trò chơi xếp hậu. Trò chơi như sau: Cho một bàn cờ vua $n \times n$ ô và n quân hậu, tìm cách đặt tất cả n quân hậu lên bàn cờ mà không có 2 quân hậu nào chiếu nhau.

Sau nhiều giờ An đã xếp được n quân hậu lên bàn cờ $n \times n$ mà không có hai quân hậu nào chiếu nhau. Sau đó An nhắc bỏ một số quân ra khỏi bàn cờ rồi đổ Bình xếp các quân mà An vừa nhắc để xếp vào bàn cờ thỏa mãn không có hai quân hậu nào chiếu nhau.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Yêu cầu: Cho kích thước bàn cờ và trạng thái bàn cờ sau khi An đã nhấc bỏ một số quân, hãy tìm cách đặt các quân hậu còn lại lên bàn cờ mà không có hai hậu nào chiếu nhau.

Input

- Gồm nhiều bộ dữ liệu, mỗi bộ trên một dòng chứa số n và tiếp theo là n số mô tả trạng thái bàn cờ - số thứ i là thứ tự cột của quân hậu đặt trên dòng i , nếu số này bằng 0 nghĩa là quân hậu trên dòng đó đã bị nhấc ra.

Output

- Gồm nhiều dòng, mỗi dòng gồm n số - số thứ i là thứ tự cột của quân hậu đặt trên dòng i , mô tả trạng thái sau khi đã đặt các quân còn lại lên bàn cờ mà không có hai quân hậu nào chiếu nhau.

NQUEEN.INP	NQUEEN.OUT
4 0 0 0 0	2 4 1 3
8 2 0 0 0 4 0 0 0	2 6 1 7 4 8 3 5

Subtask 1: $N \leq 10$;

Subtask 2: $N \leq 100$.

Solution:

Sub 1: Try hoán vị là .

Sub 2: Try vị trí đặt , ta sẽ đặt vào dòng có ít khả năng ăn các quân khác nhất để tạo ra nhiều xung đột nhất.

Bài 31: TRÒ CHƠI MARIO

Mã bài:MARIO.???

Trò chơi Mario diễn ra trên lưới ô vuông gồm m hàng và n cột. Các hàng của lưới được đánh số từ trên xuống dưới bắt đầu từ 1, còn các cột – đánh số từ trái sang phải, bắt đầu từ 1. Ô nằm giao của hàng i và cột j có tọa độ (i,j) . Trên mỗi ô vuông, hoặc là ô trống hoặc là có một cây nấm. Ban đầu Mario đang đứng tại ô (x,y) và ô này là ô không có nấm. Mario có thể đứng yên hoặc di chuyển sang các ô chung cạnh, thời gian Mario di chuyển mất 1 đơn vị thời gian. Giả sử ô (i,j) có nấm, nếu Mario đang ở ô này Mario sẽ được ăn cây nấm đó và được thưởng s_{ij} điểm, sau 2 đơn vị thời gian nấm sẽ lại được mọc lại (thời gian Mario ăn nấm là không đáng kể). Bạn có có t đơn vị thời gian điều khiển Mario để nhận được nhiều điểm nhất.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Yêu cầu: Cho bảng số $S_{ij} (i = 1, 2, \dots, m; j = 1, 2, \dots, n)$, trong đó $s_{ij} = 0$ nếu ô (i, j) không có nấm, $S_{ij} > 0$ nếu ô (i, j) có nấm, và x, y là tọa độ Mario đang đứng, t là thời gian hái nấm. Hãy tìm cách di chuyển Mario để nhận được nhiều điểm nhất.

Input

- Dòng đầu chứa số nguyên m, n, x, y và t ;
- m dòng tiếp theo, dòng thứ i chứa n số $s_{i1}, s_{i2}, \dots, s_{in} (s_{ij} \leq 10^6)$

Output

- Gồm một dòng chứa một số là điểm lớn nhất đạt được.

MARIO.INP	MARIO.OUT
1 5 1 3 3 5 2 0 3 1	9

Subtask 1: $m \cdot n \leq 5000$; $t \leq 100$ [40 tests]

Subtask 2: $m=1$; $n \leq 5000$; $t \leq 1000000$; [30 tests]

Subtask 3: $m \cdot n \leq 5000$; $t \leq 1000000$ [30 tests]

Solution:

Thứ nhất do không thể lưu được mảng 5000×5000 kiểu số nguyên lên ta xây dựng hàm

$\text{Index}(i, j) = (i-1) \cdot n + j$.

Sub 1: $F[i, j, t]$ là số điểm max khi ở ô (i, j) tại thời điểm t .

Sub 2: Mảng một chiều. Tìm 2 ô liên tiếp sẽ được lặp đi lặp lại.

Sub 3: Vẫn là 2 ô nào đó được lặp lại.

```
50 For t:=0 to Min(T,m*n) do
51   for i:=1 to m do
52     for j:=1 to n do
53       - f[i-1,j,t+1].
54       - f[i,j-1,t+1].
55       - f[i+1,j,t+1].
56       - f[i,j+1,t+1].
```


Bài 32 : NGHỊCH THỂ

Mã bài:TRANS.???

Gọi Π là tập các dãy số gồm N số nguyên dương, trong đó mỗi số nguyên từ 1 đến N xuất hiện trong dãy đúng một lần. Ta nói một cặp số trong dãy tạo thành một nghịch thể nếu như số xuất hiện trước trong dãy lại nhỏ hơn số xuất hiện sau trong dãy. Số lượng nghịch thể trong dãy được xác định như là số lượng cặp số trong dãy tạo thành nghịch thể.

Ví dụ: Số lượng nghịch thể trong dãy (1, 4, 3, 2) là 3 vì ta có 3 cặp số tạo thành nghịch thể: (4, 3), (4, 2) và (3, 2).

Yêu cầu: Với C là số nguyên không âm cho trước, hãy tính số lượng dãy trong tập Π có số lượng nghịch thể đúng bằng C .

Dữ liệu: Vào từ file văn bản TRANS.INP chứa duy nhất một dòng gồm hai số nguyên N ($1 \leq N \leq 1000$) và C ($1 \leq C \leq 10\,000$) được ghi cách nhau bởi dấu cách.

Kết quả: Ghi ra file văn bản số lượng dãy tìm được trong modun 1 000 000 007.

Ví dụ:

TRANS.INP	TRANS.OUT
9 13	17957

TRANS.INP	TRANS.OUT
4 3	6

TRANS.INP	TRANS.OUT
10 1	9

Solution:

QHD:

Gọi $F[i,j]$ là số lượng dãy độ dài i mà có j nghịch thể.

$$F[i,j] = \sum_k f[i-1, j-k] \text{ với } \forall k \in [0, \min(i-1, j)] \quad i=1..n, j=0..m.$$

Cải tiến công thức trên bằng mảng cộng dồn để được độ phức tạp $N \cdot C$.

Bài 33: CỘT ĐIỆN

Mã bài:TELEWIRE.???

HP có N cột điện đánh số từ 1 đến N , các dây điện được nối giữa các cột liền kề, tức là cột 1 nối với cột 2, cột 2 nối với cột 3,...

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Cột thứ i có chiều cao là $A[i]$. Vì mỹ quan đô thị, HT sẽ phải trả một chi phí $= C * (\text{độ chênh lệch giữa 2 cột liên tiếp})$, tức là $= C * |A[1] - A[2]| + C * |A[2] - A[3]| + \dots$. Tuy nhiên, vì HP có quen biết với Sở Xây dựng nên anh ta có thể nâng cao thêm các cột điện, nếu cột điện i nâng cao thêm X (m) thì anh ta phải trả chi phí là X^2 (VNĐ).

Bạn hãy tính chi phí tối thiểu là HP phải trả?

Input: TELEWIRE.INP

- Dòng 1: N và C ($1 \leq N \leq 100.000$; $1 \leq C \leq 100$)
- $N + 1$ dòng tiếp theo, mỗi dòng ghi 1 số nguyên $A[i]$ ($1 \leq A[i] \leq 100$)

Output: TELEWIRE.OUT

- Một số duy nhất là chi phí tối thiểu phải trả

Ví dụ:

TELEWIRE.INP	TELEWIRE.OUT	
5 2 2 3 5 1 4	15	Nâng cột 1 thêm 1: chi phí 1 VNĐ Nâng cột 4 thêm 2: chi phí 4 VNĐ Khi đó dãy: 3 3 5 3 4 Chi phí: $2 * (0 + 2 + 2 + 1) = 10$ VNĐ Tổng: 15 VNĐ

Subtask 1: time 1s.

Subtask 2: time 0,1 s.

Solution:

Gọi $F[i, h]$ là chi phí nhỏ nhất từ 1-> i khi nâng độ cao cột điện thứ i lên độ cao h .

- $f[i, h] = \min(f[i-1, h'] + (h - a[i])^2 + c * |h - h'| ; f[i, h])$.

Với Sub 2 ta sẽ có gắng phá dấu $| |$.

Đây là chi tiết code cho Sub 2.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

```
58 Function Too(h,i:longint):boolean;
59 begin
60     too:=(a[i]<=h) and (h<=min(a[i]+c,100));
61 end;
62
63 Procedure Sol;
64 var
65     i,j:longint;
66     h,hh:longint;
67     tg:int64;
68     x,y:longint;
69 begin
70     for i:=1 to n do
71         for j:=1 to 100 do f[i,j]:=maxv;
72     for i:=a[1] to a[1]+c do f[1,i]:=sqr(i-a[1]);
73     for i:=2 to n do
74         begin
75             tg:=maxv;
76             x:=min(a[i],a[i-1]);
77             y:=max(a[i]+c,a[i-1]+c);
78             for h:=x to y do
79                 begin
80                     if too(h,i-1) and (tg>f[i-1,h]-c*h) then tg:=f[i-1,h]-c*h;
81                     if too(h,i) then f[i,h]:=min(f[i,h],tg+sqr(h-a[i])+c*h);
82                 end;
83             tg:=maxv;
84             for h:=y downto x do
85                 begin
86                     if too(h,i-1) and (tg>f[i-1,h]+c*h) then tg:=f[i-1,h]+c*h;
87                     if too(h,i) then f[i,h]:=min(f[i,h],tg+sqr(h-a[i])-c*h);
88                 end;
89             res:=maxv;
90             for i:=a[n] to 100 do
91                 if f[n,i]<res then res:=f[n,i];
92         end;
```

Bài 34: VÒNG TRÒN

Mã bài:CIRCLE.???

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ là một hoán vị của dãy $(1, 2, \dots, n)$ viết quanh một vòng tròn, người ta viết vào bên cạnh mỗi số a_i một giá trị b_i là tổng của a_i với hai số đứng cạnh nó trên vòng tròn.

Yêu cầu: Cho biết dãy a_1, a_2, \dots, a_n . Hãy tìm dãy A (nếu có nhiều dãy A tương ứng với dãy B chỉ đưa ra 1 dãy nhỏ nhất theo thứ tự từ điển)

Dữ liệu: Vào từ file văn bản CIRCLE.INP

- Dòng 1 chứa số nguyên dương $n \leq 10^5$
- Dòng 2 chứa n số nguyên a_1, a_2, \dots, a_n

Kết quả: Ghi ra file văn bản CIRCLE.OUT lần lượt các số nguyên a_1, a_2, \dots, a_n trên một dòng

Ví dụ

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

CIRCLE.INP	CIRCLE.OUT
6 11 9 11 10 12 10	1 6 2 3 5 4

Solution:

Ta có : $a_{i+3} - a_i = b_{i+2} - b_{i+1}$.

1. TH 1 : N không chia hết cho 3.

VD: n=10.

Giả sử : $a_1 = 0 \Rightarrow a_4 = a_1 + (b_3 - b_2) = \dots; a_7 = \dots; a_{10}, a_3, a_6, \dots$

Do $a_{min} + \Delta = 1$ nên $a_i = a_i + \Delta$

2. TH 2: N : 3.

VD: n=12.

a. $a_1 = 0 \Rightarrow a_4, a_7, \dots, a_{10} + \Delta_1$

b. $a_2 = 0 \Rightarrow a_5, a_8, \dots, a_{11} + \Delta_2$

c. $a_3 = 0 \Rightarrow a_6, a_9, \dots, a_{12} + \Delta_3$

Đoán $1 \in a. \Rightarrow \Delta_1$, tìm x nhỏ nhất không $\in a$. sau khi tịnh tiến Δ_1 , Giả sử x thuộc dãy

b. $\Rightarrow \Delta_2 \dots \Rightarrow \Delta_3$. Thu được dãy cần tìm. **Note** có 6 Th thử tất cả.

Bài 35: KIẾN

Mã bài: ANTS.???

Cho một đàn kiến gồm N con đang đi trên một sợi dây căng ngang có hai đầu là A và B chiều dài k cm. Trong đàn có số con kiến đi về phía điểm A, những con còn lại đi về phía điểm B, ban đầu không có hai con nào ở cùng vị trí.

Các con kiến đều di chuyển với tốc độ giống nhau: 1 cm/s, khi hai con kiến gặp nhau, chúng chạm râu vào nhau rồi cùng quay lại để di chuyển theo hướng ngược lại. Khi một con kiến chạm vào điểm A hay điểm B, nó sẽ bị rơi xuống đất và không còn trên dây nữa.

Yêu cầu: Biết vị trí và hướng di chuyển của từng con kiến tại thời điểm xuất phát là thời điểm 0, tính thời điểm con kiến cuối cùng bị rơi xuống đất

Dữ liệu: Vào từ file văn bản ANTS.INP

- Dòng 1 chứa hai số nguyên dương $n \leq 10^5$; $k \leq 10^{18}$
- Dòng 2 chứa số nguyên x_1, x_2, \dots, x_n trong đó $|x_k|$ là khoảng cách từ con kiến thứ i tới điểm A, $x_i < 0$ có nghĩa là ban đầu con kiến thứ i di chuyển về phía điểm A, $x_i > 0$ có nghĩa là ban đầu con kiến thứ i di chuyển về phía điểm B. ($0 < |x_i| < k$)

Các số trên một dòng của input file được ghi cách nhau ít nhất một dấu cách

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Kết quả: Ghi ra file văn bản ANTS.OUT một số nguyên duy nhất là phần nguyên của thời điểm con kiến cuối cùng bị rơi xuống đất.

Ví dụ



ANTS.INP	ANTS.OUT
2 6 1 -5	5

Solution:

Do chỉ quan tâm tới thời điểm con kiến cuối cùng rơi lên dữ kiện 2 con chạm nhau sẽ quay lại không có ý nghĩa, dữ kiện đó nhằm đánh lừa. Ta cứ cho chúng đi tiếp, bài toán trở lên rất đơn giản.

Bài 36: VUI ĐUA CÙNG KIẾN

Mã bài: C11ANT2.???

Trịnh là một học sinh giỏi trong lớp. Dù là một học sinh khá giỏi nhưng Trịnh luôn biết cách điều chỉnh thời khóa biểu của mình để cân bằng thời gian học và thời gian chơi thể thao vì Trịnh quan điểm rằng sức khỏe và kiến thức là hai yếu tố quan trọng để trở thành một người có ích cho xã hội. Mặc dù đã lên kế hoạch chơi thể thao nhưng dạo này thời tiết khá xấu, mưa liên miên đã cản trở các dự định rèn luyện thân thể của Trịnh.

Trong một lần bị bế kế hoạch do mưa quá to và gió giật rất mạnh nên Trịnh phải ngồi nhà. Lúc rảnh rỗi, Trịnh nghĩ rằng trời mưa thì sẽ có rất nhiều kiến nên cậu đã nghĩ ra trò chơi mới lạ với những con kiến. Cậu ta lấy một đoạn dây độ dài là X và bôi một lớp đường lên sợi dây và căng ra hai đầu. Sau đó, Trịnh chộp mắt một lát trong lúc chờ đợi lũ kiến bị dụ vào. Lúc tỉnh dậy, Trịnh đã thấy kiến bu đầy đoạn dây. Một số con đang đi về đầu mút bên phải của sợi dây. Các con kiến còn lại thì đang có hướng đi về đầu mút bên trái. Vì sợi dây khá mảnh nên trường hợp hai con kiến đứng cùng một chỗ sẽ không xảy ra. Trịnh đã đánh số các con kiến trên dây từ 1 đến N THEO MỘT THỨ TỰ LỘN XỘN và ghi lại vị trí của nó cũng như hướng di chuyển. Nếu con kiến thứ i đi tới một trong 2 đầu mút của sợi dây thì ngay sau đó nó sẽ bị rút ra khỏi đoạn dây. Trịnh muốn biết thời điểm mà con kiến thứ i đi tới một trong 2 đầu mút của sợi dây. Biết rằng các con kiến đi được 1 đơn vị trong một giây và nếu hai con kiến đụng nhau, chúng sẽ lập tức đổi hướng đi. Thời điểm 0 là thời điểm vị

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

trí và hướng đi của các con kiến giống y như dữ liệu mà Trịnh đã ghi lại. Nói cách khác, thời điểm 0 là thời điểm Trịnh tỉnh dậy.

Dữ liệu vào

- Dòng đầu chứa 2 số nguyên N, X .
- Dòng thứ i trong N dòng tiếp theo chứa 2 số nguyên pos, dir . Trong đó pos là khoảng cách từ con kiến thứ i tới đầu mút bên trái của sợi dây và dir là hướng di chuyển của con kiến thứ i . Biến dir mang 2 giá trị là 0 và 1 tương ứng với hướng đi sang bên trái và hướng đi sang bên phải.

Dữ liệu ra

- Gồm N dòng, dòng thứ i chứa thời điểm mà con kiến thứ i đi ra khỏi sợi dây.

Ràng buộc

- 30% số test $N \leq 1000, X \leq 1000$.
- 30% số test $N \leq 1000, X \leq 10^9$.
- 40% số test $N \leq 10^6, X \leq 10^9$.

Ví dụ

INPUT	OUTPUT
4 10	6
3 0	8
6 0	3
2 1	3
7 1	

Solution:

Bài toán này cần quan tâm đến thời điểm rơi của từng con, vì vậy khi 2 con đụng nhau ta sẽ hoán đổi **chỉ số** và cho chúng đi tiếp.

Code chi tiết:

```
64 Procedure Sol;  
65 var i,j,tt:longint;  
66 begin  
67 fillchar(c,sizeof(c),0);  
68  
69 Sort(1,n,d,pos);  
70 Sort(1,na,a,c);  
71 Sort(1,nb,b,c);  
72 i:=1; j:=1;  
73 a[na+1]:=maxv; b[nb+1]:=maxv;  
74  
75 while (i<=na) or (j<=nb) do  
76 begin  
77 if a[i]<b[j] then  
78 begin  
79 res[pos[i]]:=a[i];  
80 inc(i);  
81 end else  
82 if a[i]>b[j] then  
83 begin  
84 res[pos[n-j+1]]:=b[j];  
85 inc(j);  
86 end else  
87 begin  
88 res[pos[i]]:=a[i];  
89 res[pos[n-j+1]]:=b[j];  
90 inc(i);  
91 inc(j);  
92 end;  
93 end;  
94 end;
```

Bài 37: THẾ GIỚI NĂM 1000003

Mã bài:VMKEY.???

Nếu là một người ưa phát minh, chắc hẳn bạn sẽ cảm thấy thế giới năm 1000003 vô cùng nhàm chán. Lúc này, tất cả mọi phát minh mà con người từng mơ ước tới đều đã trở thành hiện thực. Ô tô bay, nhà bay, thành phố bay... đã xuất hiện nhiều nhan nhản. Tất cả mọi người sống trong những ngôi nhà thông minh được điều khiển bởi giọng nói, ánh mắt hoặc suy nghĩ, với sự hỗ trợ của các hệ thống máy tính hiện đại nhất. Con người có thể di chuyển tức thời giữa không gian (giữa các hành tinh ở cách nhau hàng nghìn năm ánh sáng) và thời gian - điều này đã góp phần khiến cho dân số thế giới tăng với tốc độ chóng mặt, vượt qua con số 1000^{1000} vào năm 1000003.

Lúc này, những chiếc SmartPhone đã trở nên lỗi thời, do chúng không chịu được áp lực quá lớn khi đi qua cổng dịch chuyển tức thời hoặc khi đi vào các cỗ máy thời gian. Tất cả mọi người đều dùng điện thoại Nokia.

Do dân số thế giới đã tăng vượt mốc 1000^{1000} , những dãy số điện thoại cũng đã dài tới hàng trăm nghìn chữ số. Điều này đã khiến cho việc bấm phím số điện thoại trở nên ác mộng. Trước tình trạng này, hãng điện thoại Nokia đã đưa ra tính năng mới, cho phép bạn có thể thay đổi vị trí các phím số trên bàn phím, để bạn có thể bấm số điện thoại một cách nhanh nhất có thể.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Thời gian để bạn bấm số điện thoại bằng tổng thời gian di chuyển của ngón tay bạn giữa các phím số liên tiếp trong dãy số. Thời gian này đúng bằng khoảng cách Manhattan giữa 2 phím số: Nếu phím số **A** ở dòng **i**, cột **j**, và phím số **B** ở dòng **u**, cột **v**, thì khoảng cách Manhattan giữa 2 phím số được tính theo công thức:

$$\text{dist}(A, B) = |i - u| + |j - v|$$

Chẳng hạn, bạn muốn gọi số 357468, nếu sử dụng bàn phím cổ điển:

1	2	3
4	5	6
7	8	9
0		

bạn sẽ mất tổng thời gian là;

$$\text{dist}(3, 5) + \text{dist}(5, 7) + \text{dist}(7, 4) + \text{dist}(4, 6) + \text{dist}(6, 8)$$

$$= (|1 - 2| + |3 - 2|) + (|2 - 3| + |2 - 1|) + (|3 - 2| + |1 - 1|) + (|2 - 2| + |1 - 3|) + (|2 - 3| + |3 - 2|)$$

$$= 2 + 2 + 1 + 2 + 2$$

$$= 9$$

Nhưng nếu bạn thay đổi vị trí các phím số như hình dưới:

3	5	7
8	6	4
1	2	9
0		

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

thì bạn chỉ mất tổng thời gian là:

$$\text{dist}(3, 5) + \text{dist}(5, 7) + \text{dist}(7, 4) + \text{dist}(4, 6) + \text{dist}(6, 8)$$

$$= 1 + 1 + 1 + 1 + 1$$

$$= 5$$

Note: Bạn chỉ được phép đổi chỗ các phím số, nghĩa là sau khi bạn thay đổi các phím số, mỗi phím phải tương ứng với duy nhất 1 số, và mỗi số phải tương ứng với duy nhất 1 phím. Các phím phải nằm đúng ở 10 vị trí chứa các số 0-9 trong bàn phím cổ điển. Các phím trên điện thoại được đánh số hàng từ trên xuống dưới, đánh số cột từ trái sang phải (xem ví dụ).

Input

Gồm 1 dòng duy nhất chứa dãy số điện thoại

Output

Gồm 1 dòng duy nhất ghi thời gian nhỏ nhất để bạn bấm dãy số điện thoại, sau khi đã thay đổi vị trí các phím số một cách hợp lý

Giới hạn

- Độ dài dãy số điện thoại không quá 400,000
- Trong 20% số test, độ dài dãy số điện thoại không quá 625

Example

Input:

357468

Output:

5

Solution:

Đpt $O(10! \cdot 10^2 + N)$

Nhìn vào độ phức tạp, ta có thể hình dung ra thuật toán như sau:

+ Duyệt ra dãy số hoán vị từ 0-9.

+ Tính khoảng cách giữa 2 số x, y.

- Thực sự ta chỉ cần quan tâm các cặp kí tự đứng liên tiếp nhau trong dãy số điện thoại.

Vì thể dữ liệu từ 400000 rút gọn chỉ còn $10^2 = 100$ con số đại diện.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

- Sau đó ta có thể chạy chương trình với độ phức tạp $10! \cdot 10^2$ (thay vì $10! \cdot 400000$)

Note: Khi Try cần một số cận để tránh bị Tle.

Bài 38: DÂY CON CHUNG

Mã bài: VOSLIS.???

Cho 2 dãy $a[1..N]$ và $b[1..M]$. Gọi $c[1..k]$ là 1 dãy con chung (không cần liên tiếp) bất kì của 2 dãy này. Đặt $f(c) = \text{abs}(c[2] - c[1]) + \text{abs}(c[3] - c[2]) + \dots + \text{abs}(c[k] - c[k-1])$. Nếu số phần tử của $c < 2$ thì $f(c) = 0$.

Xác định dãy con chung có giá trị f lớn nhất và in ra giá trị đó.

Input

Dòng 1: 2 số nguyên N và M

Dòng 2: N số nguyên biểu diễn dãy A

Dòng 3: M số nguyên biểu diễn dãy B

Output

Một dòng duy nhất ghi số nguyên kết quả.

Giới hạn:

20% số test có $1 \leq N, M \leq 20$

40% số test có $1 \leq N, M \leq 200$

60% số test có $1 \leq N, M \leq 2000$

Trong tất cả các test $1 \leq N, M \leq 5000$

Trong tất cả các test $-10^9 \leq a[i], b[i] \leq 10^9$

Example

INPUT	OUTPUT
4 4 1 15 8 7	8

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

15 1 7 8	
----------	--

Giải thích:

Dãy 15 7

Solution:

QHĐ: Gọi $fc[j]$ là giá trị lớn nhất của dãy con chung xét trên dãy $1..i$, và $1..j$.

Code chi tiết:

```
26 Procedure Qnd;
27   var    i,j:longint;
28         tmp:int64;
29   begin
30     fillchar(fc,sizeof(fc),255);
31     res:=0;
32     for i:=1 to m do
33       begin
34         tmp:=0;
35         fillchar(fm,sizeof(fm),255);
36         for j:=1 to n do
37           begin
38             if a[i]=b[j] then fm[j]:=Max(tmp,fm[j]);
39             if fc[j]>-1 then tmp:=Max(tmp,fc[j]+abs(a[i]-b[j]));
40             res:=Max(res,fm[j]);
41             fc[j]:=Max(fc[j],fm[j]);
42           end;
43       end;
44   end;
```

Bài 39: ĐỒNG XU

Mã bài: LATXU.???

Cho bảng $N \times N$, mỗi lưới ô vuông chứa 1 đồng xu có thể úp hoặc ngửa. Tại mỗi bước ta có phép biến đổi trạng thái của đồng xu đó vad tất cả các đồng xu nằm ở các ô chung cạnh (úp thành ngửa, ngửa thành úp). Cho trước một trạng thái các đồng xu, hãy lập trình tìm số phép biến đổi ít nhất để đưa về trạng thái tất cả các đồng xu hoặc đều úp hoặc đều ngửa)

Input

Tập COIN.INP có dạng :

- Số nguyên dương n .

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

- Gồm n dòng, mỗi dòng n kí tự 'w' – mô tả trạng thái ngựa hoặc 'b' – mô tả trạng thái úp.

Output

Tệp COIN.OUT có dạng: Nếu có thể biến đổi được ghi số phép biến đổi ít nhất nếu không ghi "Impossible".

LATXU.INP	LATXU.OUT
4 bwbw wwww bbwb bwwb	Impossible
LATXU.INP	LATXU.OUT
4 bwwb bbwb bwwb bwww	4

Subtask 1: $N \leq 4$.

Subtask 2: $N \leq 10$.

Solution:

Sub 1: Nhận thấy thứ tự tác động vào các đồng xu là không quan trọng. Do vậy ta có thể duyệt xâu nhị phân độ dài $n*n$ để xem có hay không tác động vào đồng xu.

Sub 2: Giả sử đã chuyển hết hàng 1 của bảng về 1 trạng thái, để không làm ảnh hưởng đến hàng 1 mà có thể chuyển hết hàng 2 về cùng trạng thái với hàng 1 thì chỉ có cách duy nhất là tác động đến hàng thứ 3 của bảng, cứ như vậy cho đến hết. Nếu hàng cuối thỏa mãn thì đó là một cách chuyển. Do vậy ta chỉ cần duyệt nhị phân hàng thứ nhất của bảng sau đó kiểm tra xem có thỏa mãn bằng việc làm lần lượt các hàng ở dưới cho chúng cùng trạng thái với hàng 1.

Độ phức tạp là $O(2^{n*n})$.

Bài 40: XÂU ĐÚNG

Mã bài: RESTORE.???

Biểu thức ngoặc là xâu chỉ gồm các ký tự '(' hoặc ')'. Biểu thức ngoặc đúng và bậc của biểu thức ngoặc được định nghĩa một cách đệ quy như sau:

- Biểu thức rỗng là biểu thức ngoặc đúng và có bậc bằng 0,

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

- Nếu A là biểu thức ngoặc đúng có bậc bằng k thì (A) cũng là một biểu thức ngoặc đúng có bậc bằng k+1,
- Nếu A và B là hai biểu thức ngoặc đúng và có bậc tương ứng là k_1 và k_2 thì AB cũng là một biểu thức ngoặc đúng có bậc bằng $\max(k_1, k_2)$.

Ví dụ, 'O(O)' là một biểu thức ngoặc đúng có bậc bằng 2 còn '(O(O))' là một biểu thức ngoặc đúng và có bậc bằng 3.

Yêu cầu: Cho số nguyên k và xâu S là một xâu chỉ gồm các ký tự '(', ')' và '?', hãy đếm số cách thay các ký tự '?' trong xâu S thành ký tự '(' hoặc ')' để nhận được xâu T là biểu thức ngoặc đúng có bậc bằng k.

Input

- Dòng đầu chứa số nguyên dương ;
- Dòng thứ hai chứa xâu S (độ dài xâu S không vượt quá 200) chỉ gồm các ký tự '(', ')' và '?'.

Output

- Gồm một dòng là số cách thay các ký tự '?' trong xâu S thành ký tự '(' hoặc ')' để nhận được xâu T là biểu thức ngoặc đúng.

Ví dụ:

RESTORE	RESTORE
2 ????(?)	1

Subtask 1: $\text{Length}(S) \leq 20$

Subtask 2: $\text{Length}(S) \leq 256$.

Solution:

Sub 1: Try nhị phân, sau đó kiểm tra xem bậc của xâu có bằng k không, nếu thỏa mãn thì ta tăng kết quả lên.

Sub 2: Kết quả xâu S có dạng:
$$\begin{cases} (x_1, x_2, \dots, x_n) \\ x \in ['(' \dots ')'] \\ \text{Biểu thức đúng bậc } k \end{cases}.$$

Vậy nên ta sẽ tạo thủ tục đệ quy để xây dựng xâu S.

Vì xâu S chỉ chứa ký tự '(' hoặc ')' nên để kiểm tra xem xâu S có dạng đúng dạng hay không thì ta chỉ cần dùng một biến mo, khi gặp ký tự '(' thì ta $\text{inc}(\text{mo})$ ngược lại ta $\text{dec}(\text{mo})$, sau khi kết thúc xâu nếu $\text{mo}=0$ thì S là một xâu đúng.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Gọi $\text{Calc}(i, \text{mo}, \text{bac})$ là số lượng xâu thỏa mãn khi xây dựng đến ký tự $S[i]$, mo là hiệu số lượng ký tự '(' với số lượng ký tự ')' khi đến vị trí $s[i]$, bac là bậc của xâu $S[1...i]$.

Dùng mảng $f[i, \text{mo}, \text{bac}]$ để lưu trữ kết quả tính toán, đồng thời giảm việc tính toán lại các trạng thái đã tính trước đó.

Sau là chi tiết thủ tục Calc

```
21 Function Calc(i,mo,bac:longint):int64;
22 var
23     Count :int64;
24 begin
25     if f[i,mo,bac] <> -1 then exit(f[i,mo,bac]);
26     if i > n then exit(ord((mo=0) and (bac=k)));
27     Count:=0;
28     if s[i]<>'(' then inc(Count,Calc(i+1,mo+1,max(mo+1,bac)));
29     if (s[i]<>')' and (mo>0) then
30         inc(Count,Calc(i+1,mo-1,bac));
31     f[i,mo,bac]:=Count;
32     exit(Count)
33 end;
```

Bài 41: SỐ 6 VÀ SỐ 8

Mã bài: NUM68.???

Một số nguyên dương N ($N > 1$) luôn có thể biểu diễn dưới dạng tổng hai số nguyên dương A và B ($N = A + B, A \leq B$). Trong bài toán này chúng ta sẽ quan tâm đến các cách biểu diễn một số nguyên N thành tổng hai số nguyên dương A và B thỏa mãn tính chất: trong biểu diễn của A hoặc B phải chứa chữ số 6 hoặc chữ số 8.

Ví dụ: $N=10$, có tất cả 5 cách biểu diễn nhưng chỉ có 2 cách biểu diễn thỏa mãn là: $2+8$; $4+6$.

Yêu cầu: Cho số nguyên dương N ($N > 1$), hãy đếm số cách biểu diễn N thành tổng hai số nguyên dương A và B thỏa mãn tính chất: trong biểu diễn của A hoặc B phải chứa chữ số 6 hoặc chữ số 8.

Input

- Gồm nhiều dòng, mỗi dòng tương ứng với một số nguyên N ($1 < N < 10^{18}$).

Output

- Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với dữ liệu vào.

NUM68.INP	NUM68.OUT
10	2
19	4

Subtask 1: $N \leq 10^6$

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Subtask 2: $N \leq 10^{18}$

Solution:

Sub 1: For 2 số A hoặc số B, A thỏa mãn hoặc N-A thỏa mãn có chứa chữ số 6 hoặc 8, $A \leq B$

Sub 2: Nghiệm:

$A_1 A_2 \dots A_k$

$B_1 B_2 \dots B_k$

$N_1 N \dots N_k$

Vậy nên ta sẽ đi xây dựng thủ tục Calc(i, nho, ok) mô tả phép tính cộng, để tính kết quả của bài toán

- i là vị trí N[i]
- nho là số nhớ cho hàng đơn vị sau
- ok lưu trữ đã có hay chưa có số 6 hoặc 8.

Dùng mảng f[i, nho, ok] để lưu trữ kết quả đã tính.

Sau là chi tiết thủ tục Calc.

```
39 Function Calc(i,nho,ok:longint):int64;  
40 var  
41     ai,bi:longint;  
42     Count:int64;  
43     oo:longint;  
44 begin  
45     if f[i,nho,ok]<>-1 then exit(f[i,nho,ok]);  
46     if i=0 then  
47         begin  
48             if (nho=0) and (ok=0) then exit(1);  
49             exit(0);  
50         end;  
51     count:=0;  
52     for ai:=0 to 9 do  
53         for bi:=0 to 9 do  
54             if (ai+bi+nho) mod 10 =n[i] then  
55                 begin  
56                     if (ai=6) or (ai=8) or (bi=6) or (bi=8) or (ok=0)  
57                     then oo:=0 else oo:=1;  
58                     Count:=Count+Calc(i-1,(ai+bi+nho) div 10,oo);  
59                 end;  
60     f[i,nho,ok]:=Count;  
61     exit(Count);  
end;
```

Bài 42: CHIA KẸO

Mã bài: SWEETS.???

Ba anh em An, Bình, Cường có n chiếc kẹo và muốn chia n chiếc kẹo thành 3 phần, sau đó mỗi người sẽ nhận một phần. An rất yêu thích số a vì đó là ngày sinh của An, do đó An sẽ chỉ nhận phần mà có số kẹo là bội của a , thậm chí An sẽ không nhận kẹo nếu không có

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

phần nào là bội của a . Tương tự, Bình và Cường có số yêu thích tương ứng là b và c và cũng chỉ nhận phần có số kẹo là bội của số mà mình yêu thích. Vì vậy, ba anh em quyết định sẽ tìm cách tách số n thành 3 số nguyên không âm x, y, z thỏa mãn yêu cầu sau:

1. $x + y + z = n$
2. x chia hết cho a ; y chia hết cho b ; z chia hết cho c .

Yêu cầu: Cho 4 số nguyên dương n, a, b, c . Hãy đếm số cách tách n thành 3 số nguyên không âm x, y, z thỏa mãn yêu cầu của ba anh em.

Input

- Dòng đầu chứa số nguyên T là số bộ dữ liệu trong file;
- T dòng sau, mỗi dòng chứa 4 số nguyên dương n, a, b, c ($n \leq 10^9$; $a, b, c \leq 31$).

Output

- Gồm T dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào.

SWEETS.INP	SWEETS.OUT
2	3
10 3 3 4	0
10 6 6 6	

Subtask 1 : $T \leq 3$.

Subtask 2 : $T \leq 1000$.

Solution:

Nghiệm:

$$\begin{array}{r} A \quad a_1 a_2 \dots a_k \\ B \quad b_1 b_2 \dots b_k \\ C \quad c_1 c_2 \dots c_k \\ \hline N \quad n_1 n_2 \dots n_k \end{array}$$

Nhận thấy:

$$\overline{a_1 a_2 \dots a_k} \bmod m = (a_1 * 10^{k-1} \bmod m + a_2 * 10^{k-2} \bmod m + \dots + a_k \bmod m) \bmod m$$

Sub 1:

Gọi $\text{pow}[i] = 10^i$

Xây dựng thủ tục $\text{Calc}(i, \text{nho}, \text{ra}, \text{rb}, \text{rc})$, với:

- i là vị trí $N[i]$.
- nho là số nhớ trong phép tính cộng nếu tổng trong hàng ≥ 10 .
- $\text{ra}, \text{rb}, \text{rc}$ lần lượt là số dư của phép chia A, B, C lần lượt cho a, b, c .

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Dùng mảng $F[i, \text{nho}]$ để lưu trữ kết quả đã tính trước đó.

Sub 2:

Vẫn dựa trên mô hình phép tính cộng như trên, ta thấy for 3 vòng $a[i], b[i], c[i]$ mất đi là 10^3 , để cải tiến đoạn này ta chuyển N về cơ số 2. Sau đó thực hiện phép cộng trên hệ cơ số 2.

Mô hình

```
28 Function Calc(i,nho:longint):int64;
29 var   xi,yi,zi:longint;
30       Count:int64;
31 begin
32     if f[i,nho]<>-1 then exit(f[i,nho]);
33     if i=0 then exit(ord(nho=0));
34     Count:=0;
35     for xi:=0 to 1 do
36         for yi:=0 to 1 do
37             for zi:=0 to 1 do
38                 if (xi*a+yi*b+zi*c+nho) mod 2=x[i] then
39                     Count:=Count+Calc(i-1,(xi*a+yi*b+zi*c+nho) div 2);
40     f[i,nho]:=Count;
41     exit(count);
42 end;
```

Bài 43: SỐ MDS

Mã bài: MDS.???

Một số tự nhiên n được gọi là số MDS nếu n chia hết cho tổng các chữ số của n .

Yêu cầu:

Cho số s , tìm số tự nhiên nhỏ nhất là số MDS mà có tổng các chữ số bằng s .

Input

- Gồm nhiều test, một dòng một test chứa một số s ($s \leq 1000$)

Output

- Ghi số MDS mà có tổng các chữ số bằng s .

MDS.INP	MDS.OUT
5	5
10	190

Solution:

Với dữ liệu nhỏ ta có thể for trâu để tìm kết quả.

Tư tưởng chính của bài toán là dùng thuật toán BFS. Chi tiết Code:

```

20 Procedure Sol(s:longint);
21 var i,j,x,y,u,v,d,c:longint;
22 begin
23   for i:=0 to s do
24     for j:=0 to s do Free[i,j]:=true;
25   d:=1; c:=1; Qx[d]:=0; Qy[d]:=0; Id[d]:=0; Free[0,0]:=false;
26   While d<=c do
27     begin
28       x:=Qx[d]; y:=Qy[d]; inc(d);
29       for i:=0 to 9 do
30         begin
31           u:=x+i; v:=(y*10+i) mod s;
32           if Free[u,v] then
33             begin
34               inc(c); Qx[c]:=u; Qy[c]:=v; Free[u,v]:=false;
35               Id[c]:=i; Trx[c]:=d-1;
36               if (u=s) and (v=0) then begin Save:=c; Exit; end;
37             end;
38         end;
39     end;
40 end;
41
42 Procedure Trace(i,s:longint);
43 var sum:longint;
44 begin
45   sum:=0; nres:=0;
46   While not ((Id[i]=0) and (sum=s)) do
47     begin
48       inc(nres); res[nres]:=Id[i]; Sum:=Sum+Id[i]; i:=Trx[i];
49     end;
50 end;

```

Bài 44: PHÁT GIẤY THI

Mã bài: PAPERS.???

Giáo sư X sắp phải đi họp và ông chuẩn bị một bài tập làm văn cho sinh viên làm trong thời gian ông đi vắng. Giáo sư X có m tờ giấy thi để phát cho n sinh viên. Tùy theo trình độ viết dài, viết hỏng của từng người, ông xác định chính xác được rằng sinh viên thứ i phải được phát không ít hơn a_i tờ giấy thi.

Yêu cầu: Đếm số cách phát m tờ giấy thi cho n sinh viên theo yêu cầu trên. Hai cách phát giấy thi được gọi là khác nhau nếu tồn tại một sinh viên nhận được số tờ giấy thi khác nhau trong hai cách đó.

Dữ liệu: Vào từ file văn bản PAPERS.INP

- Dòng 1 chứa hai số nguyên dương $m \leq 10^9$; $n \leq 10^5$
- Dòng 2 chứa n số nguyên dương a_1, a_2, \dots, a_n ($\forall i: a_i \leq 10^9$)

Các số trên một dòng của input file được ghi cách nhau bởi dấu cách

Kết quả: Ghi ra file văn bản PAPERS.OUT một số nguyên duy nhất là số dư của phép chia kết quả tìm được cho 1000000007 ($10^9 + 7$).

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Ví dụ

PAPERS.INP	PAPERS.OUT
5 3 1 1 2	3

Giải thích, 3 cách chia có thể là

1 | 1 | 3

1 | 2 | 2

2 | 1 | 2

Solution:

Bài toán có thể chuyển về bài: Đếm số cách phát m tờ giấy thi cho n sinh viên sao cho mỗi sinh viên có ít nhất 1 tờ giấy thi.

Gọi $S = \sum (a_i - 1)$; $k = n - 1$; $nn = m - s - 1$.

Kết quả: $C_{nn}^k \bmod (10^9 + 7)$.

Bài toán con: Tính $R = \frac{a}{b} \bmod e$

- $R = a * \text{Ucln2}(b, e) \bmod e$

Thủ tục Ucln2

```

24  Function Ucln2(m,n:int64):int64;
25  var      xm,xn:int64;
26          r,q,xr:int64;
27  begin
28      xm:=1; xn:=0;
29      while n<>0 do
30      begin
31          q:=m div n;
32          r:=m - q*n;
33          xr:=xm-q*xn;
34          m:=n;
35          xm:=xn;
36          n:=r;
37          xn:=xr;
38      end;
39      exit(xm);
40  end;
```

Note: Thủ tục trên chỉ đúng khi e là một số nguyên tố.

Bài 45: ĐẾM BI

Mã bài: VOSCAL.???

Ngoài sở thích sử dụng dầu ăn để nấu cơm giúp mẹ, Khải còn một sở thích khác mà ít người biết tới, đó là sưu tầm bi. Cũng bởi vậy, sau nhiều năm tìm tòi, nghiên cứu, trăn lộn và trao đổi bi với lũ trẻ trong xóm, Khải đã có N viên bi giống hệt nhau, cả về màu sắc lẫn kích thước. Lo sợ mẹ sẽ quẳng hết bi của mình cho ve chai nên cậu lên kế hoạch để N viên bi vào M hộp giống nhau cho mẹ khó tìm, tất nhiên số bi trong các hộp có thể là khác nhau dẫn tới việc có hộp có đủ N viên bi, hoặc có hộp không có viên bi nào. Đến khi chia bi vào hộp, cậu bỗng nảy ra một câu hỏi, liệu sẽ có bao nhiêu cách chia N viên bi giống nhau và M chiếc hộp. Câu hỏi khó đó khiến cho Khải cả ngày bồn chồn, bồn chồn....Do kết quả quá lớn nên Khải chỉ cần bạn in ra phần dư của số cách chia bi cho một số nguyên K cho trước.

Giới hạn bộ test :

Subtask 1: (10 test) $n, m \leq 300, k \leq 10^9$.

Subtask 2: (10 test) $n, m \leq 2000, k \leq 10^9$.

Subtask 3: (10 test) $n, m \leq 10^5, k \leq 10^9$.

Subtask 4: (10 test) $n, m \leq 2 \cdot 10^7, k = 10^9 + 7$

Input

Gồm 3 số nguyên N , M và K .

Output

Một dòng duy nhất là số cách chia bi.

Example

Input:

2 2 1000

Output:

3

Giải thích:

Có 3 cách chia bi là: (2-0), (0-2) và (1-1).

Solution:

Tương tự bài trên, nhưng chú ý 3 Sub đầu K không là số nguyên tố lên ta phải xử lý khác.

Ta thấy:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Do tử số luôn chia hết cho mẫu số lên ta cần rút gọn phân số. Ta phân tích tử và mẫu ra thừa số nguyên tố.

Để tính nhanh trong phân tích $n!$ ra thừa số nguyên tố thì mũ số của số nguyên tố p là ? . Ta dùng thuật toán sau.

Code:

```
45  Function Get(p:longint; n:longint): longint;  
46  var      tmp:longint;  
47  begin  
48      tmp:=0;  
49      while n>=p do  
50      begin  
51          inc(tmp,n div p);  
52          n:=n div p;  
53      end;  
54      get:=tmp;  
55  end;
```

Note: $k=0$ hoặc $k=1$.

Bài 45: ÁO PHÔNG

Mã bài: SHIRTS.???

Theo truyền thống, các thí sinh dự thi ACM ICPC được phát áo phông in biểu tượng của cuộc thi. Áo được đóng trong hòm hình khối lập phương kích thước mỗi cạnh 1m. Hòm được đặt vào một góc của căn phòng hình chữ nhật kích thước $m \times n$ mét. Ban Tổ chức còn cẩn thận lấy sơn viết lên nắp hòm dòng chữ ACM ICPC.

Chỉ vài giờ sau người ta phát hiện ra cái hòm ở góc đó cản trở một số hoạt động phải tổ chức ở đây và cần phải di chuyển hòm sang góc đối diện. Cái hòm quá nặng, không thể khiêng đi được, người ta phải di chuyển bằng các lật hòm (đưa một mặt kề với đáy tiếp xúc với sàn thành mặt đáy mới). Để tránh làm chột sàn số lần lật yêu cầu phải là ít nhất. Nhưng quả là “họa vô đơn chí, phúc bất trùng lai”! Một vấn đề mới lại nảy sinh: dòng chữ trên hòm chưa hoàn toàn khô, vì vậy, khi trở thành mặt đáy nó sẽ để lại vết trên sàn. Do đó, bên cạnh yêu cầu số lần lật ít nhất còn có thêm yêu cầu số lần mặt nắp ban đầu trở thành mặt đáy cũng phải là ít nhất.

CHUYÊN ĐỀ TỔNG HỢP 2013-2014

Hãy xác định số lần mặt nắp ban đầu trở thành mặt đáy trong quá trình lật.

Dữ liệu: Vào từ file văn bản SHIRTS.INP gồm một dòng chứa 2 số nguyên n và m ($1 \leq n, m \leq 10^9$).

Kết quả: Đưa ra file văn bản SHIRTS.OUT một số nguyên – số lần mặt nắp ban đầu trở thành mặt đáy.

Ví dụ:

SHIRTS.INP	SHIRTS.OUT
3 3	1

Solution

Phân biệt các trường hợp:

- $m = 1$ hoặc $n = 1 \rightarrow \text{res} = (m+n+2) \text{ div } 4$,
- $m = 2$ hoặc $n = 2 \rightarrow \text{res} = 0$,
- Còn lại: $\text{res} = 1$!