

## 102. Xếp lịch hoàn thành công việc

Program	ORDER.*
Input	ORDER.INP
Output	ORDER.OUT
Score	100

Có  $N$  công việc được đánh số từ 1 đến  $N$ . Hoàn thành công việc  $i$  phải chi phí cho thợ số tiền là  $p_i$  (gọi là giá gốc của công việc  $i$ ). Ngoài ra nếu công việc  $j$  được hoàn thành trước công việc  $i$  thì muốn thợ hoàn thành công việc  $i$  phải trả thêm cho thợ là  $s_{ij}$  đồng nữa (để đảm bảo khi tiến hành công việc  $i$ , thợ không phá vỡ hoặc làm hỏng công việc  $j$ ). Ví dụ, khi trang trí nội thất, nếu trong nhà chưa lắp cửa kính thì chỉ phải trả tiền sơn tường là 10 triệu đồng, nhưng khi đã lắp cửa kính rồi mới sơn nhà thì phải trả thêm cho thợ sơn nhà một khoản phí là 2 triệu đồng nữa).

Hãy viết chương trình xếp lịch làm các công việc sao cho chi phí tiền công thợ là nhỏ nhất.

*Input:*

- Dòng đầu tiên chứa số nguyên dương  $N$  ( $0 < N < 16$ ).
- $N$  dòng tiếp theo, mỗi dòng chứa đúng  $N$  số nguyên: số nguyên thứ  $i$  là giá gốc của công việc thứ  $i$ , số nguyên thứ  $j$  trên dòng này ( $i$  khác  $j$ ) là số tiền  $s_{ij}$  phải trả thêm cho công việc  $i$  nếu công việc  $j$  đã hoàn thành trước công việc  $i$ . Các giá tiền nguyên không âm, không vượt quá 100000.

*Output:* Một số nguyên duy nhất là tổng số tiền nhỏ nhất để hoàn thành  $N$  công việc.

*Example:*

Input	Output
2 10 10 9000 10	30
3 14 23 0 5 14 0 1000 9500 14	47

## 103. Thương thức

Program	ENJOY.*
Input	ENJOY.INP
Output	ENJOY.OUT
Score	100

Khi BT bước vào nhà hàng và ngồi xuống bàn, nhân viên phục vụ ngay lập tức mang cho anh ta thực đơn. Có  $n$  món ăn và BT muốn ăn đúng  $m$  món trong số chúng.

BT biết rằng món thứ  $i$  sẽ đem lại cho anh ta độ dễ chịu là  $a_i$ . Có một số món ăn không được ăn cùng nhau và một số món sẽ trở lên tuyệt vời nếu ăn cùng nhau. BT có  $k$  quy tắc tăng độ dễ chịu có dạng như sau: Nếu ăn món  $x$  chính xác trước món  $y$  (tức là không ăn món  $x$  nào khác giữa  $x$  và  $y$ ) thì độ dễ chịu tăng lên một lượng là  $c$ .

Hãy giúp BT chọn các món ăn và trình tự ăn sao cho tổng độ dễ chịu thu hoạch được là lớn nhất

*Input:*

- Dòng đầu tiên chứa ba số nguyên  $n, m, k$  ( $1 \leq m \leq n \leq 18, 0 \leq k \leq n \cdot (n - 1)$ ) - số lượng món ăn trong thực đơn, số món ăn cần ăn và số quy tắc tăng độ dễ chịu.
- Dòng thứ hai chứa  $n$  số nguyên  $a_i$  ( $0 \leq a_i \leq 10^9$ ) - độ dễ chịu của món ăn thứ  $i$
- $k$  dòng tiếp theo mỗi dòng chứa một qui tắc. Dòng thứ  $i$  ghi ba số nguyên  $x_i, y_i$  và  $c_i$  ( $1 \leq x_i, y_i \leq n; 0 \leq c_i \leq 10^9$ ) có nghĩa rằng nếu ăn món  $x_i$  ngay trước món  $y_i$  thì độ dễ chịu

tăng lên một lượng là  $c_i$ . Chú ý rằng không tồn tại hai cặp chỉ số  $i, j$  thỏa mãn  $1 \leq i < j \leq k$  mà  $x_i = x_j$  và  $y_i = y_j$ .

*Output:* Một số nguyên duy nhất là tổng độ dễ chịu lớn nhất có được.

*Example:*

Input	Output
4 3 2 1 2 3 4 2 1 5 3 4 2	12

## 104. Hôn nhân

Program MARRIAGE.\*  
Input MARRIAGE.INP  
Output MARRIAGE.OUT  
Score 100

Ở một làng nọ có  $n$  cô gái chưa chồng xinh đẹp. Vào một ngày đẹp trời có  $n$  chàng trai tuấn tú đến chỗ già bản xin lấy  $n$  cô gái làm vợ (mỗi chàng trai lấy một cô gái). Già bản vốn là một người thông thái nhìn xa trông rộng nên dự tính được nếu chàng trai thứ  $i$  lấy cô gái thứ  $j$  làm vợ thì làng sẽ nhận được số tiền hồi môn là  $a_{ij}$ .

Hỏi rằng già bản bố trí  $n$  đám cưới như thế nào (mỗi chàng trai lấy một cô gái và mỗi cô gái chỉ lấy một chàng trai) sao cho tổng số tiền hồi môn mà dân làng nhận được là lớn nhất. Bạn cũng chỉ cần tính số tiền này. Ngoài ra cũng cần phải biết có bao nhiêu cách xếp khác nhau để được số tiền trên (hai cách xếp là khác nhau nếu như có ít nhất một cặp xuất hiện trong cách này nhưng không xuất hiện trong cách kia)

*Input:*

- Dòng đầu tiên ghi số nguyên dương  $n \leq 20$  là số cô gái và số chàng trai.
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số, số thứ  $j$  là  $a_{ij}$  ( $0 \leq a_{ij} \leq 10^7$ )

*Output:* In ra hai số nguyên là tổng số tiền hồi môn lớn nhất và số cách để có được số tiền như vậy

*Example:*

Input	Output
4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	34 24

## 105. Robot lau nhà

Program MCLEAN.\*  
Input MCLEAN.INP  
Output MCLEAN.OUT  
Score 100

Sàn nhà là một hình chữ nhật được chia thành các ô. Các ô của sàn nhà có thể là ô sạch, ô bẩn hoặc có vật cản. Robot có thể dọn một ô bẩn thành ô sạch nếu như nó ở trên ô đó. Mỗi bước, robot có thể từ một ô di chuyển đến một trong 4 ô chung cạnh nếu như ô này không có vật cản.

Xác định số bước ít nhất để một robot có thể lau sạch các ô bẩn trên sàn nhà (nếu có thể).

*Input:*

- Dòng đầu tiên ghi  $n, m$  là số cột và số hàng của sàn nhà ( $1 \leq n, m \leq 20$ )
- $m$  dòng tiếp theo, mỗi dòng ghi  $n$  ký tự mô tả tình trạng của các ô vuông trên sàn nhà với:
  - o ' ': ô sạch
  - o '\*': ô bẩn

- o 'x': ô có vật cản
- o 'o': ô xuất phát của robot (1 con)

Có không quá 10 ô bẩn

**Output:** In ra số bước di chuyển ít nhất để robot có thể lau sạch các ô bẩn. In ra -1 nếu như nó không thể làm được điều này

**Example:**

Input	Output
<pre> 7 5 ..... .o...*. ..... .*...*. ..... </pre>	8

## 106. T-Shirts

Program TSHIRTS.\*  
 Input TSHIRTS.INP  
 Output TSHIRTS.OUT  
 Score 100

Cinderella cùng với Hoàng tử tổ chức một bữa tiệc nhỏ. Mỗi người đến dự tiệc đều mang theo một số chiếc áo thun. Có tất cả 100 loại áo thun khác nhau và được đánh số từ 1 đến 100. Tất nhiên không có người nào có hai chiếc áo thun cùng loại.

Cinderella muốn biết có bao nhiêu cách khác nhau để những người dự tiệc mặc chiếc áo thun của mình sao cho không có hai người mặc cùng một loại áo thun. Hai cách mặc được gọi là khác nhau nếu như có ít nhất một người mặc hai loại áo thun khác nhau.

**Input:** Dòng đầu tiên ghi số nguyên dương  $T \leq 10$  là số bộ dữ liệu. Tiếp theo là  $T$  nhóm dòng mô tả các bộ dữ liệu với cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên dương  $n \leq 10$  là số người có mặt trong bữa tiệc.
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa ít nhất một số nguyên và nhiều nhất 100 số nguyên khác nhau mô tả các loại áo thun mà người thứ  $i$  mang theo.

**Output:** Với tất cả các bộ dữ liệu in ra số lượng cách mặc áo. Con số này có thể rất lớn nên bạn chỉ cần lấy phần dư của nó khi chia cho  $10^9+7$

**Example:**

input	output
2	4
2	4
3 5	
8 100	
3	
5 100 1	
2	
5 100	

## 107. Cắt bánh

Program PARTITION.\*  
 Input PARTITION.INP  
 Output PARTITION.OUT  
 Score 100

Chiếc bánh sinh nhật của Mai có dạng hình vuông được chia thành  $N \times N$  ô vuông. Trên mỗi ô vuông có gắn một số lượng sô-cô-la khác nhau.

Theo truyền thống Mai thực hiện cắt bánh không quá  $K$  nhát cắt song song với các cạnh của bánh và chỉ đi qua cạnh các ô vuông ( $1 \leq K \leq 2N-2$ )

Mai muốn rằng sau khi cắt bánh thành các phần thì lượng sô-cô-la ở phần có nhiều sô-cô-la nhất là ít nhất. Viết chương trình giúp cô bé thực hiện điều này.

*Input:*

- Dòng đầu tiên ghi hai số nguyên dương  $N, K$  ( $2 \leq N \leq 15$ )
- $N$  dòng tiếp theo, dòng thứ  $i$  ghi  $N$  số nguyên nằm trong khoảng từ 0 đến 1000 mô tả lượng sô-cô-la ở các ô tương ứng trên hàng thứ  $i$  (tính từ trái qua phải)

*Output:* Một số nguyên là lượng sô-cô-la nhỏ nhất ở phần có nhiều sô-cô-la nhất.

*Example:*

input	output
3 2 1 1 2 1 1 2 2 2 4	4

## 108. Remembering Strings

Program REMEMBER.\*  
Input REMEMBER.INP  
Output REMEMBER.OUT  
Score 100

Bạn có  $n$  xâu ký tự có cùng độ dài chỉ chứa các chữ cái tiếng Anh thường. Chúng ta nói rằng các xâu ký tự này là dễ nhớ nếu như với mỗi xâu ký tự đều tồn tại một vị trí  $i$  và chữ cái  $c$  sao cho chỉ dãy ký tự này có chữ cái  $c$  ở vị trí  $i$ .

Ví dụ các dãy ký tự {"abc", "aba", "adc", "ada"} không phải là dễ nhớ. Tuy nhiên tập các dãy {"abc", "ada", "ssa"} là dễ nhớ bởi vì:

- Dãy đầu tiên là dãy duy nhất ở vị trí 3 có ký tự 'c'
- Dãy thứ hai là dãy duy nhất ở vị trí 2 có ký tự 'd'
- Dãy thứ ba là dãy duy nhất ở vị trí 2 có ký tự 's'

Bạn có thể thay đổi tập các dãy ký tự một chút cho dễ nhớ. Với số tiền là  $a_{ij}$  bạn có thể đổi ký tự ở vị trí thứ  $j$  của xâu ký tự thứ  $i$  thành một ký tự in thường tiếng Anh bất kỳ.

Hãy tính số tiền tối thiểu bạn cần phải có để có thể biến đổi tập các dãy ký tự đã cho sao cho dễ nhớ

*Input:*

- Dòng đầu tiên chứa hai số nguyên dương  $n, m$  ( $1 \leq n, m \leq 20$ ) lần lượt là số lượng dãy ký tự và độ dài của mỗi dãy ký tự.
- $n$  dòng tiếp theo, mỗi dòng chứa một dãy ký tự của tập. Các dãy này có cùng độ dài  $m$
- $n$  dòng cuối cùng, dòng thứ  $i$  chứa  $m$  số nguyên  $a_{i1}, a_{i2}, \dots, a_{im}$  ( $0 \leq a_{ij} \leq 10^6$ )

*Output:* In ra một số nguyên duy nhất là số tiền ít nhất phải có

*Example:*

Input	Output
4 5 abcde abcde abcde abcde 1	3

## 109. Ném đĩa

Program MGUARD.\*  
Input MGUARD.INP

Output MGUARD.OUT  
Score 100

Nông dân John(FJ) và đàn bò của ông ta đang chơi môn ném đĩa. Bessie ném cái đĩa xuống sân, nhưng nó lại đi thẳng tới vị trí của Mark ở bên đối thủ. Mark có chiều cao  $H$  ( $1 \leq H \leq 10^9$ ), nhưng có  $N$  con bò ở bên đội của Bessie đang đứng quanh Mark ( $2 \leq N \leq 20$ ). Những con bò này chỉ có thể bắt được cái đĩa nếu chúng có thể đứng chồng lên nhau với độ cao ít nhất phải bằng độ cao của Mark. Mỗi con bò  $N$  có một độ cao, cân nặng, và sức khỏe khác nhau. Sức khỏe của một con bò được xác định bằng số lượng cân nặng tối đa mà nó có thể chất những con bò khác lên trên nó. Được cho trước những thông tin trên, Bessie muốn biết rằng đội của cô ta có thể bắt được cái đĩa hay không, và nếu bắt được thì Bessie muốn biết hệ số an toàn tối đa của các con bò chất lên nhau đó. Hệ số an toàn tối đa của một đồng con bò chồng lên nhau là số lượng cân nặng có thể chồng lên trên các con bò này, với điều kiện rằng không có con bò nào phải giữ các con bò ở trên vượt quá sức chịu đựng của mình.

*Input:*

- Dòng thứ nhất gồm hai số  $N$  và  $H$ .
- $N$  dòng tiếp theo, mỗi dòng mô tả một con bò: chiều cao, cân nặng, và sức khỏe. Tất cả các số này đều là số nguyên dương và không vượt quá 1 tỉ.

*Output:* Nếu đội của Bessie có thể chất thành một đồng đủ cao để bắt được cái đĩa, hãy in ra hệ số an toàn tối đa có thể đạt được. Nếu không, hãy in ra “Mark is too tall” (không bao gồm dấu mở/đóng ngoặc kép).

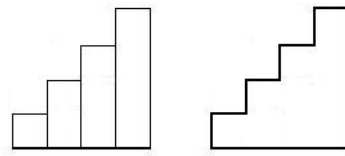
*Example:*

input	output
4 10 9 4 1 3 3 5 5 5 10 4 4 5	2

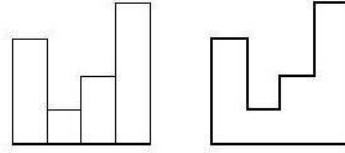
## 110. Biểu đồ

Program HISTOGRAM.\*  
Input HISTOGRAM.INP  
Output HISTOGRAM.OUT  
Score 100

Trong thống kê người ta hay sử dụng biểu đồ cột đứng gồm những hình chữ nhật độ rộng đơn vị đứng liền nhau. Ta có một biểu đồ như vậy với chiều cao của các hình chữ nhật khác nhau từng đôi một. Một vài biểu đồ sẽ cho chu vi bao quanh các hình chữ nhật lớn nhất. Nhiệm vụ của bạn là tìm hoán vị các cột để cho chu vi của biểu đồ là lớn nhất và đếm xem có bao nhiêu hoán vị như vậy?



(a)



(b)

Trong hình trên, hình (a) ứng với hoán vị (1,2,3,4) (bộ dữ liệu 1 trong ví dụ) có chu vi là 16 còn hình (b) ứng với hoán vị (3,1,2,4) có chu vi là 20 (đây cũng là giá trị lớn nhất).

**Input:** Gồm nhiều bộ dữ liệu, mỗi bộ dữ liệu bắt đầu bằng một dòng chứa số nguyên dương  $n$  ( $2 \leq n \leq 15$ ). Dòng thứ hai gồm  $n$  số nguyên dương phân biệt mô tả độ cao các cột (độ cao có giá trị không quá 100). Giá trị  $n = 0$  báo hiệu kết thúc các bộ dữ liệu. Trong mỗi file input có không quá 50 bộ dữ liệu.

**Output:** Ứng với mỗi bộ dữ liệu in ra một dòng chứa hai số nguyên, số thứ nhất là chu vi lớn nhất và số thứ hai là số lượng hoán vị cho kết quả là chu vi này.

**Example:**

input	output
4	20 8
1 2 3 4	24 2
3	
2 6 5	
0	

**Ghi chú:**

- Subtask 1:  $n \leq 10$
- Subtask 2:  $n \leq 15$