



DESENV. E IMPLEMENTAÇÃO DE ALGORITMOS – 08/10/2016

Este caderno contém 10 páginas com a descrição de 10 problemas a seguir:

Problemas:

- A - Subindo...(What goes up - Valladolid 481)
- B - Twin Towers (Valladolid 10066)
- C - Nested Dolls (Valladolid 11368)
- D - Prince and Princess (Valladolid 10635)
- E - Jackpot (The Jackpot - Valladolid 10684)
- F - Sistema aberto de créditos (Open Credit System - Valladolid 11078)
- G - Palíndromo Mais Longo (Longest Palindrome - Valladolid 11151)
- H - Subsequência crescente máxima (Longest Increasing Subseq. - Val. 11499)
- I - Garbage Heap (Valladolid 10755)
- J - Area (Valladolid 11951 )

São dados também os seguintes 4 Desafios:

- K - Finding Seats(Archive Valladolid 3932 )
- L - Maximum sum on a torus (Valladolid 10827 )
- M - Virus(Valladolid 12511 )
- N - Minimal Subarray Length (Valladolid 12697 )

Boa sorte.

## Problema A (Valladolid 481)

# Subindo...

Escreva um programa que seleciona a maior subsequência estritamente crescente de uma sequência de números inteiros.

## Entrada

O arquivo de entrada contém uma sequência de inteiros (positivos negativos ou zero). Cada linha conterá apenas um inteiro.

## Saída

A saída deste programa começa com uma linha indicando o tamanho da maior subsequência estritamente crescente, uma nova linha contendo apenas o carácter '-', e a subsequência por si mesma, imprimindo-se um inteiro por linha. Se a entrada contiver mais de uma subsequência máxima, deve-se imprimir a subsequência que ocorre por último na entrada.

## Exemplo de Entrada

```
-7
10
9
2
3
8
8
1
```

## Exemplo de saída

```
4
-
-7
2
3
8
```

## Problema B (Valladolid 10066)

### The Twin Towers

Once upon a time, in an ancient Empire, there were two towers of dissimilar shapes in two different cities. The towers were built by putting circular tiles one upon another. Each of the tiles was of the same height and had integral radius. It is no wonder that though the two towers were of dissimilar shape, they had many tiles in common.

However, more than thousand years after they were built, the Emperor ordered his architects to remove some of the tiles from the two towers so that they have exactly the same shape and size, and at the same time remain as high as possible. The order of the tiles in the new towers must remain the same as they were in the original towers. The Emperor thought that, in this way the two towers might be able to stand as the symbol of harmony and equality between the two cities. He decided to name them the *Twin Towers*.

Now, about two thousand years later, you are challenged with an even simpler problem: given the descriptions of two dissimilar towers you are asked only to find out the number of tiles in the highest twin towers that can be built from them.

### Input

The input file consists of several data blocks. Each data block describes a pair of towers. The first line of a data block contains two integers  $N_1$  and  $N_2$  ( $1 \leq N_1, N_2 \leq 100$ ) indicating the number of tiles respectively in the two towers. The next line contains  $N_1$  positive integers giving the radii of the tiles (from top to bottom) in the first tower. Then follows another line containing  $N_2$  integers giving the radii of the tiles (from top to bottom) in the second tower. The input file terminates with two zeros for  $N_1$  and  $N_2$ .

### Output

For each pair of towers in the input first output the twin tower number followed by the number of tiles (in one tower) in the highest possible twin towers that can be built from them. Print a blank line after the output of each data set.

### Sample Input

```
7 6
20 15 10 15 25 20 15
15 25 10 20 15 20
8 9
10 20 20 10 20 10 20 10
20 10 20 10 10 20 10 10 20
0 0
```

### Sample Output

```
Twin Towers #1
Number of Tiles : 4
```

```
Twin Towers #2
Number of Tiles : 6
```

## Problema C (Valladolid 11368)

### Nested Dolls

Dilworth is the world's most prominent collector of Russian nested dolls: he literally has thousands of them! You know, the wooden hollow dolls of different sizes of which the smallest doll is contained in the second smallest, and this doll is in turn contained in the next one and so forth. One day he wonders if there is another way of nesting them so he will end up with fewer nested dolls? After all, that would make his collection even more magnificent! He unpacks each nested doll and measures the width and height of each contained doll. A doll with width  $w_1$  and height  $h_1$  will fit in another doll of width  $w_2$  and height  $h_2$  if and only if  $w_1 < w_2$  and  $h_1 < h_2$ . Can you help him calculate the smallest number of nested dolls possible to assemble from his massive list of measurements?

#### Input

On the first line of input is a single positive integer  $1 \leq t \leq 20$  specifying the number of test cases to follow. Each test case begins with a positive integer  $1 \leq m \leq 20000$  on a line of itself telling the number of dolls in the test case. Next follow  $2m$  positive integers  $w_1, h_1, w_2, h_2, \dots, w_m, h_m$ , where  $w_i$  is the width and  $h_i$  is the height of doll number  $i$ .  $1 \leq w_i, h_i \leq 10000$  for all  $i$ .



#### Output

For each test case there should be one line of output containing the minimum number of nested dolls possible.

#### Sample Input

```
4
3
20 30 40 50 30 40
4
20 30 10 10 30 20 40 50
3
10 30 20 20 30 10
4
10 10 20 30 40 50 39 51
```

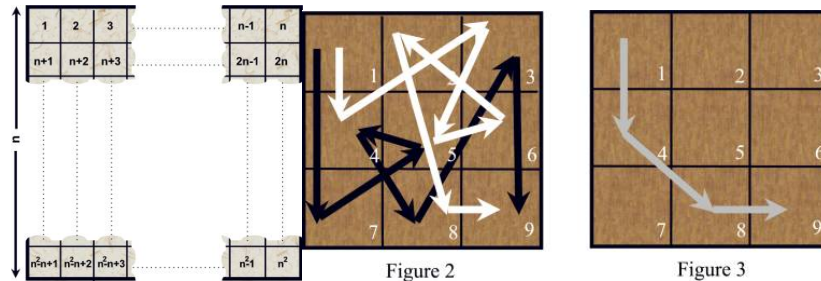
#### Sample Output

```
1
2
3
2
```

## Problema D (Valladolid 10635)

# Prince and Princess

In an  $n \times n$  chessboard, Prince and Princess plays a game. The squares in the chessboard are numbered  $1, 2, 3 \dots n^2$ , as shown below:



Prince stands in square  $1$ , make  $p$  jumps and finally reach square  $n^2$ . He enters a square at most once. So if we use  $x_p$  to denote the  $p$ -th square he enters, then  $x_1, x_2, \dots, x_{p+1}$  are all different. Note that  $x_1 = 1$  and  $x_{p+1} = n^2$ . Princess does the similar thing - stands in square  $1$ , make  $q$  jumps and finally reach square  $n^2$ . We use  $y_1, y_2, \dots, y_{q+1}$  to denote the sequence, and all  $q+1$  numbers are different.

Figure 2 show a  $3 \times 3$  square, a possible route for Prince and a different route for Princess. The Prince moves along the sequence:  $1 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 8 \rightarrow 3 \rightarrow 9$  (Black arrows), while the Princess moves along this sequence:  $1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 8 \rightarrow 9$  (White arrow).

The King -- their father, has just come. "Why move separately? You are brother and sister!" said the King, "Ignore some jumps and make sure that you're always together."

For example, if the Prince ignores his **2nd, 3rd, 6th** jump, he'll follow the route:  $1 \rightarrow 4 \rightarrow 8 \rightarrow 9$ . If the Princess ignores her **3rd, 4th, 5th, 6th** jump, she'll follow the same route:  $1 \rightarrow 4 \rightarrow 8 \rightarrow 9$ , (The common route is shown in figure 3) thus satisfies the King, shown above. The King wants to know the longest route they can move together, could you tell him?

## Input

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 10$ ), the number of test cases followed. For each case, the first line contains three integers  $n, p, q$  ( $2 \leq n \leq 250, 1 \leq p, q < n^2$ ). The second line contains  $p+1$  different integers in the range  $[1..n^2]$ , the sequence of the Prince. The third line contains  $q+1$  different integers in the range  $[1..n^2]$ , the sequence of the Princess.

## Output

For each test case, print the case number and the length of longest route. Look at the output for sample input for details.

## Sample Input

## Output for Sample Input

<pre> 1 3 6 7 1 7 5 4 8 3 9 1 4 3 5 6 2 8 9                     </pre>	<pre> Case 1: 4                     </pre>
--	--

## Problema E (Valladolid 10684)

# Jackpot

Como Manuel quer ficar rico rápido e sem muito trabalho, ele decidiu ser um jogador. . Inicialmente, ele planeja estudar os ganhos e perdas dos jogadores, de tal forma que ele possa identificar padrões de vitórias consecutivas e elaborar uma estratégia ganha-ganha. Mas Manuel, apesar de achar que é muito inteligente, não sabe programar computadores. Então contratou você para escrever programas que vão ajudá-lo a elaborar suas estratégias.

## O Problema

Sua primeira tarefa é escrever um programa que identifica o ganho máximo possível em uma sequência de apostas. Uma aposta é um volume de dinheiro e é ou vencedora (gravada como um valor positivo) ou perdedora (gravada como um valor negativo).

## Entrada

A entrada consiste de um número positivo  $N \leq 10000$ , que dá o tamanho da sequência, seguido por  $N$  inteiros. Cada aposta é um inteiro maior que 0 e menor que 1000.

A entrada termina com  $N = 0$ .

## Saída

Para cada conjunto de entrada, a saída conterá uma linha com a correspondente solução. Se a sequência não possibilita ganhar nenhum dinheiro, a saída é a mensagem "Losing streak."

## Exemplo de entrada

```
5
12 -4
-10 4
9
3
-2 -1 -2
0
```

## Exemplo de Saída

```
The maximum winning streak is 13.
Losing streak.
```

## Problema F (Valladolid 11078)

### Sistema aberto de créditos

Em um sistema aberto de créditos, os estudantes podem escolher quaisquer cursos que quiserem, mas há um problema. Alguns dos estudantes são mais altos do que os outros estudantes. O professor de certo curso tem encontrado um grande número de alunos que vieram de classes superiores (como se fossem freqüentar o curso pré-requisito, depois de passar por um curso avançado). Mas ele quer fazer justiça aos novos alunos. Então, vai fazer um teste de classificação (basicamente um teste de QI) para avaliar o nível de diferença entre os alunos. Ele quer saber o montante máximo da pontuação que o aluno senior é mais alto do que qualquer estudante do primeiro ano. Por exemplo, se um estudante sênior recebe 80 e um estudante júnior recebe 70, então esse valor é 10. Tenha cuidado porque não queremos o valor absoluto. Ajude o professor a descobrir uma solução.

### Entrada

A entrada consiste de  $T$  casos ( $T < 20$ ). Cada caso começa com um inteiro  $n$ , que é o número de estudantes no curso. Esse valor está entre 2 e 100.000. As próximas  $N$  linhas contem  $n$  inteiros, onde o  $i$ -ésimo inteiro é o score do  $i$ -ésimo estudante. Todos esses inteiros têm valores absolutos inferiores a 150000. Se  $i < j$ , então o  $i$ -ésimo estudante é senior em relação ao  $j$ -ésimo estudante.

### Saída

Para cada caso de teste, escreva a solução em uma linha. Siga o formato dos exemplos.

### Exemplo de entrada

```
3
2
100
20
4
4
3
2
1
4
1
2
3
4
```

### Saída para o exemplo

```
80
3
-1
```

## Problema G (Valladolid 11151)

# Palíndromo mais longo

Um **palíndromo** é um string que pode ser lido da mesma forma da direita para a esquerda ou vice-versa. Por exemplo I, GAG e MADAM são palíndromos, mas ADAM não. Aqui, consideramos um *string* *vazio* como um palíndromo.

A partir de um string que não é um palíndromo, você pode sempre tirar algumas letras e obter uma subseqüência palindrômica. Por exemplo, dado o string ADAM, você remove a letra M e obtém o palíndromo ADA.

### Some *longer* palindromic words in English

TATTARRATTAT

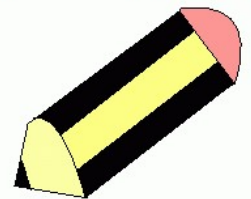
(a knock on the door)

RACECAR

ROTATOR

PULLUP

REDDER ...



(Source: Wikipedia)

**Escreva um programa para determinar o tamanho** do palíndromo mais longo que você pode obter de um string.

## Entrada e Saída

A primeira linha da entrada contém um inteiro  $T$  ( $\leq 60$ ). Cada uma das próximas  $T$  linhas é um string, cujo tamanho é sempre menor que 1000.

Para  $\geq 90\%$  dos casos de teste, o tamanho do string é  $\leq 255$ .

Para cada string de entrada, seu programa deve imprimir o tamanho do mais longo palíndromo que você pode obter, pela remoção de zero ou mais caracteres do mesmo.

## Exemplo de entrada

```
2
ADAM
MADAM
```

## Exemplo de Saída

```
3
5
```



## Problema H (Valladolid 11499)

# Subsequência crescente máxima

O problema de determinar a subsequência crescente máxima em uma lista de inteiros já é clássico em competições de programação. Apesar disso, é o problema que você tem que resolver aqui. Mas, para que você não boceje durante a solução, vai uma mudança: a lista dos números é dada em uma matriz bidimensional, e a subsequência crescente está embutida em uma submatriz da original.

Vamos definir o problema mais precisamente. Você deve escrever um programa que, dada uma matriz de inteiros, determina a maior submatriz tal que, quando linearizada, resulta em uma sequência crescente. A figura abaixo mostra alguns exemplos das submatrizes máximas que contêm sequências crescentes.

1	2	5
4	6	7
10	8	3

1	2	1	2
9	6	7	3
8	7	2	8

22	2	14	22	23
16	21	22	31	31
57	33	43	45	50
46	51	66	83	93

## Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois inteiros  $N$  e  $M$  indicando as dimensões da matriz ( $1 \leq N, M \leq 600$ ). Cada uma das  $N$  próximas linhas contém  $M$  inteiros, separados por um espaço, descrevendo os elementos da matriz. O elemento  $X_{i,j}$  da matriz é o  $j$ -ésimo inteiro da  $i$ -ésima linha da entrada ( $-10^6 \leq X_{i,j} \leq 10^6$ ). O fim da entrada é a linha com 0 0.

## Saída

Para cada caso de teste, seu programa deve imprimir uma única linha contendo o tamanho da maior submatriz que, quando linearizada, resulte em uma sequência crescente.

## Exemplo de entrada

```
3 3
1 2 5
4 6 7
10 8 3
3 4
1 2 1 2
9 6 7 3
8 7 2 8
4 2
-23 -12
0 2
16 15
57 33
```

```
4 4
2 2 2 2
2 2 2 2
2 2 2 2
2 2 2 2
0 0
```

## Exemplo de Saída

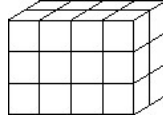
```
4
3
4
1
```

## Problema I (Valladolid 10755)

# Garbage Heap

Farmer John has a heap of garbage formed in a rectangular parallelepiped.

It consists of  $A \times B \times C$  garbage pieces each of which has a value. The value of a piece may be 0, if the piece is neither profitable nor harmful, and may be negative which means that the piece is not just unprofitable, but even harmful (for environment).



The farmer thinks that he has too much harmful garbage, so he wants to decrease the heap size, leaving a rectangular nonempty parallelepiped of smaller size cut of the original heap to maximize the sum of the values of the garbage pieces in it. You have to find the optimal parallelepiped value. (Actually, if any smaller parallelepiped has value less than the original one, the farmer will leave the original parallelepiped).

## Input

The first line of the input contains the number of the test cases, which is at most 15. The descriptions of the test cases follow. The first line of a test case description contains three integers  $A$ ,  $B$ , and  $C$  ( $1 \leq A, B, C \leq 20$ ). The next lines contain  $A \cdot B \cdot C$  numbers, which are the values of garbage pieces. Each number does not exceed  $2^{31}$  by absolute value. If we introduce coordinates in the parallelepiped such that the cell in one corner is  $(1,1,1)$  and the cell in the opposite corner is  $(A,B,C)$ , then the values are listed in the order

$$\begin{aligned} &(1, 1, 1), (1, 1, 2), \dots, (1, 1, C), \\ &(1, 2, 1), \dots, (1, 2, C), \dots, (1, B, C), \\ &(2, 1, 1), \dots, (2, B, C), \dots, (A, B, C). \end{aligned}$$

The test cases are separated by blank lines.

## Output

For each test case in the input, output a single integer denoting the maximal value of the new garbage heap. Print a blank line between test cases.

## Examples

Input

1

2 2 2  
-1 2 0 -3 -2 -1 1 5

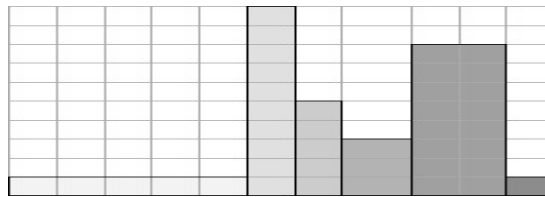
Output

6

Looking at Murcia's skyline from left to right, we have a series of  $N$  buildings. Each building has its own height and width. You have to discover if the skyline is increasing or decreasing.

We say the skyline is **increasing** if the longest increasing subsequence of buildings is bigger or equal than the longest decreasing subsequence of buildings; in other case, we say it is **decreasing**. A **subsequence** is a subset of the original sequence, in the same order. The **length** of a subsequence of buildings is the sum of the widths of its elements.

For example, assuming we have six buildings of heights: 10, 100, 50, 30, 80, 10; and widths: 50, 10, 10, 15, 20, 10; then we have an increasing subsequence of 3 buildings and total length 85, and a decreasing subsequence of 1 building and total length 50 (also, there is a decreasing subsequence of 4 buildings and length 45). So, in this case, we say that the skyline is increasing. You can see this example below.



## The Input

The first line of the input contains an integer indicating the number of test cases. For each test case, the first line contains a single integer,  $N$ , indicating the number of buildings of the skyline. Then, there are two lines, each with  $N$  integers separated by blank spaces. The first line indicates the heights of the buildings, from left to right. The second line indicates the widths of the buildings, also from left to right.

## The Output

For each test case, the output should contain a line. If the skyline is increasing, the format will be:

Case  $i$ . Increasing ( $A$ ). Decreasing ( $B$ ).

If the skyline is decreasing, the format will be:

Case  $i$ . Decreasing ( $B$ ). Increasing ( $A$ ).

where  $i$  is the number of the corresponding test case (starting with 1),  $A$  is the length of the longest increasing subsequence, and  $B$  is the length of the longest decreasing subsequence.

## Sample Input

```
3
6
10 100 50 30 80 10
50 10 10 15 20 10
4
30 20 20 10
20 30 40 50
3
```

```
80 80 80
15 25 20
```

## Sample Output

```
Case 1. Increasing (85).
Decreasing (50).
Case 2. Decreasing (110).
Increasing (50).
Case 3. Increasing (25).
Decreasing (25).
```

## Problema J (Valladolid 11951)

### Area

Steve has a dream to move from dormitory where we live now to a private house. He has already earned some money by solving various problems and now wants to make a first move towards his dream. Steve is going to buy a plot for his future house. However, due to high tax rate, he is going to buy only one plot. Certainly Steve wants to maximize its area for the amount of money he owns (and if there are several plots of same area, Steve certainly chooses a cheaper one). Can you help Steve?

The whole district where Steve wants to reside is divided into  $N * M$  equal strips. Each strip is sold separately, for its own price  $P_{ij}$ . According to the country law, plot is defined as a rectangular collection of strips with sides parallel to the district.

### INPUT

There is a number of tests  $T$  ( $T \leq 110$ ) on the first line. After  $T$  tests follows. Each test case is defined by three numbers  $N M K$  ( $N, M \leq 100; K \leq 10^9$ ). Next  $N$  lines with  $M$  numbers each stands for prices of the strips  $P_{ij}$  ( $P_{ij} \leq 10^6$ ).

### OUTPUT

For each test print a line formatted as "Case #T: S P", where  $T$  stands for test case number (starting from 1),  $S$  for maximal plot area that Steve can afford and  $P$  for plot's total cost.

### SAMPLE INPUT

```
1
5 6 15
10 1 10 20 10 10
30 1 1 5 1 1
50 1 1 20 1 1
10 5 5 10 5 1
40 10 90 1 10 10
```

### SAMPLE OUTPUT

```
Case #1: 6 10
```