

Olimpíada Informàtica  
Comunitat Valenciana

## III Olimpíada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior

Universitat de València - Escola Tècnica Superior d'Enginyeria

### Concert

L'organització d'un festival musical està programant un concert dins del festival. Per a això disposa d'un llistat ordenat de 12 grups musicals (o solistes, però els direm *grup* a tots) que s'han oferit a participar, i ha de triar només alguns d'ells per a configurar un dels concerts del festival.

El llistat ordenat de grups és el següent:

*Aitana*  
*Bad Bunny*  
*Daddy Yankee*  
*Dorian*  
*La Casa Azul*  
*Lori Meyers*  
*Love of Lesbian*  
*Reincidentes*  
*Ska-P*  
*The Killers*  
*Vetusta Morla*  
*Viva Suecia*

L'organització té les següents regles per a organitzar el concert:

- El grup que comence el concert tocarà també al final i, per tant, tancarà el concert. Per consegüent, aquest grup tocarà 2 vegades. I serà l'únic que repetisca.
- D'aqueix llistat ordenat de grups, l'organització ha de triar un **número N de grups (entre 2 i 8)**. Per tant, el concert tindrà N+1 actuacions, ja que, com hem comentat abans, el grup que comença el concert repeteix al final.
- D'aqueix llistat ordenat **no es poden triar** grups que **estiguen distanciats més de S posicions** en la llista, sent **S un número entre 2 i 5**.
- Els grups s'han de triar seguint l'ordre de la llista de grups. És a dir, [*Aitana*, *Love of Lesbian*, *Dorian*, *The Killers*, *Aitana*] no és una configuració vàlida.
- Es considera que la llista és circular, per la qual cosa, si la meua selecció de grups començara per *Daddy Yankee* i acabara en *The Killers*, es consideraria que entre *The Killers* i *Daddy Yankee* (que també actuaria al final) hi hauria un salt de 5 posicions en la llista ordenada de grups.
- En ser circular la llista, la meua selecció pot començar, per exemple, amb *Love of Lesbian* i acabar, per exemple, amb la *Casa Azul*. La llista està ordenada, però no té ni inici ni final, ja que és circular.
- No pot donar-se més d'una volta a la llista circular.

### Exemples de configuracions de concert vàlides:

[*Aitana, Dorian, Love of Lesbian, The Killers, Aitana*]  
seria vàlida amb  $N=4$  i  $S=3$ .

[*La Casa Azul, Love of Lesbian, Ska-P, Viva Suecia, Daddy Yankee, La Casa Azul*]  
seria vàlida amb  $N=5$ ,  $S=3$ .

[*Ska-P, Viva Suecia, Daddy Yankee, La Casa Azul, Love of Lesbian, Ska-P*] seria vàlida amb  
 $N=5$  i  $S=4$

### Format de l'entrada

*Grup que comença el concert*  
 $N$   
 $S$

Exemple:

*The Killers*  
8  
5

Això significa que cerquem un concert amb  $8+1=9$  actuacions que comença amb el grup *The Killers* i acaba amb el mateix grup, de manera que el salt màxim en la llista circular ordenada de grups siga de 5 posicions.

### Format de l'eixida

El programa ha d'imprimir el nombre de possibles configuracions diferents del concert donats les dades d'entrada proporcionats. És possible que no existisca per a alguns casos cap configuració possible. En tal cas, l'eixida seria 0.

### Exemples

Si volem configurar un concert que comence amb *Aitana* amb  $N=4$  grups (i per tant que tinga  $4+1=5$  actuacions) amb un espaiat màxim  $S=3$ , posaríem a l'entrada:

*Aitana*  
4  
3

I l'eixida seria 1, perquè l'única configuració vàlida amb aquestes regles seria aquesta:

*[Aitana, Dorian, Love of Lesbian, The Killers, Aitana]*

Per tant, en aquest cas l'especificació d'entrada/eixida seria:

Entrada:

Aitana 4 3
------------------

Eixida:

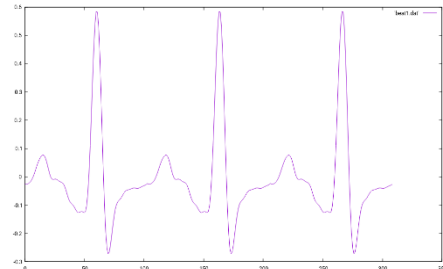
1
---

### Puntuació

- **20 punts:** Si es resol bé amb  $S=2$  i  $N=7$ .
- **30 punts:** Si es resol bé amb  $S=2$ , i qualsevol valor vàlid de  $N$ .
- **50 punts:** Si es resol bé amb qualsevol valor vàlid de  $N$  i de  $S$ .

## Amb cor

Fa uns dies, vas fer una revisió mèdica i, entre altres coses, et van fer un electrocardiograma. Li has demanat al metge que et guarde la informació de l'electrocardiograma en un fitxer. Desafortunadament, el senyal té molt soroll pel que necessites suavitzar-la i per a això utilitzaràs un mètode de mitjana mòbil.



La mitjana mòbil en un punt es calcula com la mitjana aritmètica dels  $k$  dades (amb  $k$  imparella) al voltant del punt actual, inclòs el propi punt. Inclourà per tant,  $(k-1)/2$  dades abans del punt i uns altres  $(k-1)/2$  dades després del punt.

Considerant  $i$  l'índex o posició del punt actual, la fórmula per a la mitjana mòbil en aqueix punt serà:

$$M_k(i) = \frac{1}{k} * \sum_{j=i-\frac{k-1}{2}}^{i+\frac{k-1}{2}} x_j$$

Com que necessitem diverses dades abans i després de  $i$ , en principi la mitjana només la definim per a  $i \geq (k-1)/2$  i  $i \leq n - 1 - (k-1)/2$ , considerant que els índexs per a les dades van de 0 a  $n-1$ , la resta de valors es queden sense modificar. Per exemple, si les dades són:  $[0 \ 2 \ 1 \ 4]$  i  $k=3$ , llavors la mitjana mòbil per a la 2<sup>a</sup> posició (índex 1) serà  $(0+2+1)/3 = 1$  i per a la 3<sup>a</sup> posició (índex 2) serà  $(2+1+4)/3 = 2.3333$  que ardonem a l'enter més pròxim, és a dir, **2**, amb el que el resultat seria:  $[0 \ 1 \ 2 \ 4]$ . El primer element (índex 0) es queda igual, ja que no hi ha element a la seua esquerra per a poder calcular la mitjana i el mateix li succeeix a l'últim element. Aquest problema es correspondrà amb l'opció 0 de suavitzat. Les dades de l'electrocardiograma seran sempre dades **senceres** entre  $-1000$  i  $+1000$  i les dades d'eixida hauran de ser també **sencers**, però el resultat de la divisió en realitzar la mitjana haurà d'arredonar-se a l'enter més pròxim.

En l'opció 1, intentarem generalitzar la mitjana perquè incloga també els primers i últims punts. Per a això, considerarem que el primer i últim punt s'estenen indefinidament. D'aquesta forma, es podrà calcular la mitjana des de  $i \geq 0$  fins a  $i \leq n-1$ . En l'exemple anterior, per a la primera posició (índex 0), tindriem  $(0+0+2)/3 = 0.6666$  que ardonit dona 1, i per a l'última:  $(1+4+4)/3 = 3$ , amb el que el resultat quedaria  $[1 \ 1 \ 2 \ 3]$ .

Escriu un programa que reba com a entrada l'opció de mitjana mòbil a usar i el senyal de l'electrocardiograma i retorne aquest senyal suavitzat.

### Entrada

L'entrada constarà de 2 línies. En la primera línia estarà l'opció triada per al suavitzat (Op), la grandària del vector que conté l'electrocardiograma (N) i el nombre d'elements

a utilitzar en total per a realitzar la mitjana de cada punt (K) i en la segona línia, els N nombres enters representant l'electrocardiograma.

$$0 \leq Op \leq 1$$

$$1 \leq N \leq 1000$$

$$1 \leq K \leq 21 \quad (\text{Sol valors imparells})$$

$$-1000 \leq x_i \leq 1000 \quad (\text{Valors de l'electrocardiograma})$$

### Eixida

N números representant l'electrocardiograma suavitzat.

### Puntuació

**Test 1** (10 punts):  $Op = 0$ .  $K = 1$ . L'electrocardiograma resultant és el mateix que l'original.

**Test 2** (30 punts):  $Op = 0$ .  $K \geq 1$ .

**Test 3** (60 punts): Sense restriccions.

### Exemples

Entrada

```
0 27 1
0 2 1 4 16 3 -1 0 1 2 1 -2 -5 3 8 15 30 60 55 40 28 12 -1 -10 -6 -4 1
```

Eixida

```
0 2 1 4 16 3 -1 0 1 2 1 -2 -5 3 8 15 30 60 55 40 28 12 -1 -10 -6 -4 1
```

Entrada

```
0 27 3
0 2 1 4 16 3 -1 0 1 2 1 -2 -5 3 8 15 30 60 55 40 28 12 -1 -10 -6 -4 1
```

Eixida

```
0 1 2 7 8 6 1 0 1 1 0 -2 -1 2 9 18 35 48 52 41 27 13 0 -6 -7 -3 1
```

Entrada

```
1 27 3
0 2 1 4 16 3 -1 0 1 2 1 -2 -5 3 8 15 30 60 55 40 28 12 -1 -10 -6 -4 1
```



### III Olimpiada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior

Universitat de València - Escola Tècnica Superior d'Enginyeria

Eixida

1 1 2 7 8 6 1 0 1 1 0 -2 -1 2 9 18 35 48 52 41 27 13 0 -6 -7 -3 -1
--

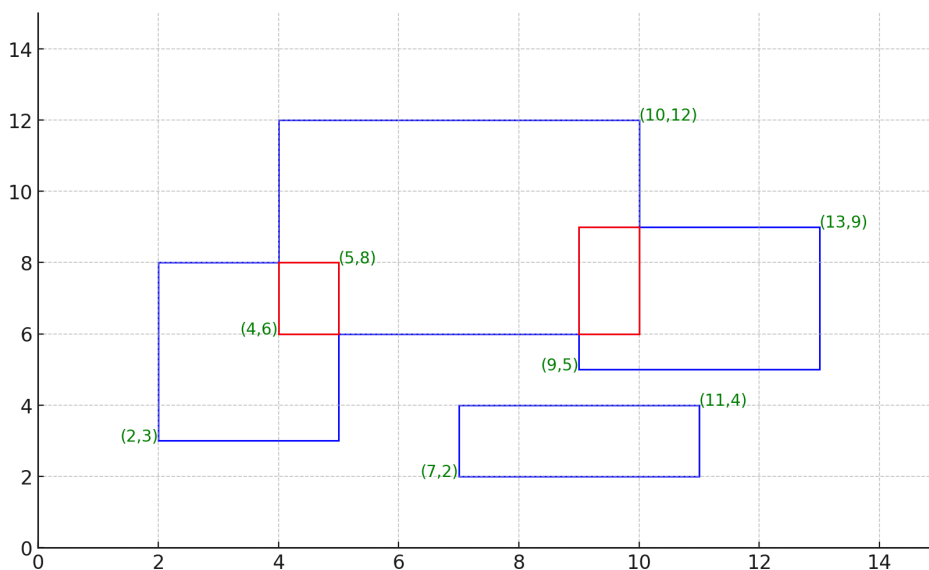
## Exposició de pintures

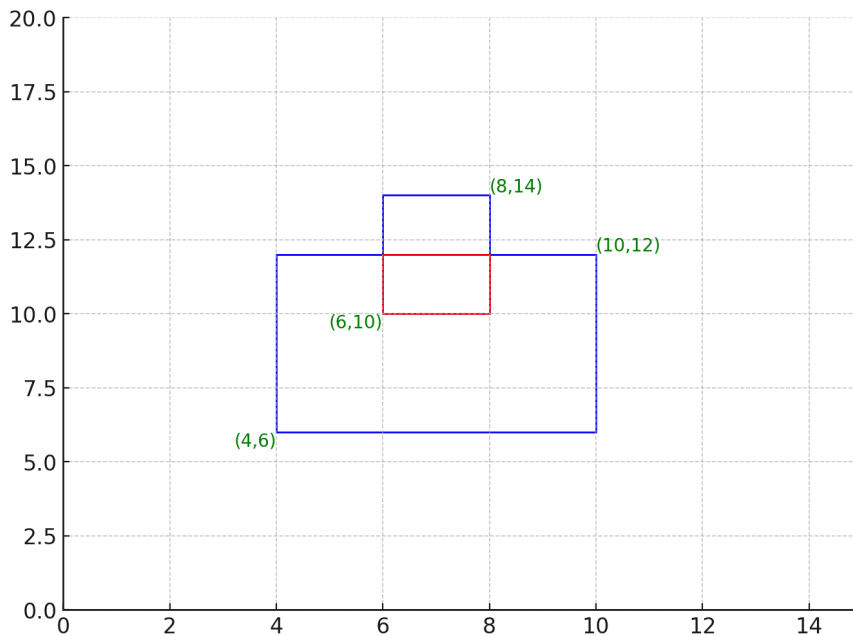
Suposem que eres el/la coordinador/a d'una exposició d'art en un prestigiós museu. L'exposició inclou una sèrie de valuoses pintures, cadascuna emmarcada en un quadre rectangular. Estos quadres s'exhibiran en una sola paret, i a causa de limitacions d'espai, alguns quadres poden superposar-se parcial o totalment amb uns altres.

Per a planificar adequadament l'espai i la disposició dels quadres, necessites **calcular el perímetre total de la superfície** que ocuparan tots els quadres en la paret, tenint en compte les **superposicions**. Això és necessari per a determinar la quantitat de material de marc necessari per al disseny de l'exposició.

**Objectiu:** Desenvolupa un programa que, donades les dimensions i posicions de cada quadre en la paret, calcule el **perímetre total ocupat** per tots els quadres en l'exposició, ajustant per a no comptar dues vegades les parts del perímetre que se **superposen** entre quadres.

A continuació es mostren de manera gràfica alguns exemples de possibles distribucions de quadres en la paret. El perímetre total de marc necessari per a emmarcar-los serà el que està representat en blau, sense tindre en compte el perímetre de les zones superposades.





## ENTRADA

La primera línia conté un enter  $N$ ,  $0 \leq N \leq 10$ , indicant el nombre de quadres en la paret.

Les següents  $N$  línies contenen quatre enters cadascun representant les coordenades  $x,y$  de cada quadre (cantonada inferior esquerra i cantonada superior dreta, respectivament) en relació amb la paret on s'exhibiran. El rang de les coordenades  $x,y$  serà:  $0 \leq x \leq 99$ ,  $0 \leq y \leq 99$

## EIXIDA

Un únic enter que representa el perímetre total de la superfície que ocuparan tots els quadres, tenint en compte les superposicions.

## Consideracions:

- Els rectangles sempre estaran orientats de tal manera que els seus costats siguin paral·lels als eixos del sistema de coordenades. No hi ha rotacions del rectangle en cap angle.
- Considerem que en una intersecció només hi ha dos rectangles involucrats.



## Exemples

Exemple 1 (el de la imatge):

Dades d'entrada:

4  
2 3 5 8  
4 6 10 12  
9 5 13 9  
7 2 11 4

Eixida:

54

Exemple 2 (sense superposició):

Dades d'entrada:

2  
1 1 4 4  
5 5 8 8

Eixida:

24

Exemple 3 (superposició):

Dades d'entrada:

2  
1 1 5 5  
3 3 8 8

Eixida:

28

Exemple 4 (rectangles niats):

Dades d'entrada:

2  
2 2 7 7  
3 3 6 6

Eixida:











20

## Puntuació

- **Test 1 (20 punts):** Es resolen bé els casos en els quals els rectangles no intersectan entre si.
- **Test 2 (80 punts):** Es resolen bé els casos en els quals hi ha interseccions entre diversos rectangles i/o nidaments.

## LABERINT ARITMÈTIC

El nostre professor de matemàtiques, en un intent per fer-nos aprendre aritmètica d'una forma amena, ha inventat els laberints aritmètics, com el que veus ací:

			3		
				+	
=			4		
	@	x			
	2				

Començant per la fletxa blanca, i seguint la direcció que marca, segueix en línia recta. Quan trobes una fletxa negra, canvia la direcció en el sentit indicat per la fletxa, i continua així fins a arribar a la casella final, indicada pel símbol '@'. Pel camí aniràs trobant números i operacions, que hauràs d'anar resolent a mesura que avances, fins a arribar a la casella final. Indica el resultat final de les operacions.

En aquest exemple, el resultat és  $(3 + 4) * 2 = 14$

Només hem passat una vegada per cada operand o operador en aquest exemple, però és possible que en alguns taulers es passe més d'una vegada pel mateix operand o operador (per exemple, la primera vegada passem verticalment i la segona horitzontalment). La seqüència d'operacions correcta més curta possible és un dígit seguit del signe igual (i per descomptat, @ al final). El signe igual només apareix al final de la seqüència d'operacions.

És possible que alguns taulers estiguen mal dissenyats i no tinguen solució, perquè seguint les fletxes mai s'arriba a la casella final, o bé la seqüència d'operacions no alterna operands i operadors, començant i acabant per un operand, seguit del signe '='.

## ENTRADA

La primera línia indica el nombre de files (N) i el nombre de columnes del tauler (M). Tots dos números són sencers entre 3 i 10. La segona línia indica les coordenades de la casella d'eixida (la cantonada superior esquerra del tauler té la coordenada (0,0)). A continuació, el tauler es representa mitjançant N files de longitud M. Cada fila conté M caràcters (sense espais en blanc entre ells) que representa el contingut d'una fila del tauler. Les fletxes (tant la inicial com la resta es representen mitjançant els següents caràcters:

fletxa a dalt: '^'  
fletxa a baix: 'v'  
fletxa cap a l'esquerra: '<'  
fletxa cap a la dreta: '>'

La resta de caràcters indiquen números de l'1 a 9, operacions aritmètiques ('+', '-', '\*', '/', '='), la casella final ('@') o un casella buida ('0'). Assumeix que en la casella d'eixida sempre hi haurà una fletxa.

La següent entrada equival a l'exemple anterior:

```
5 6
0 1
0>03v0
>v00+0
=0v40<
0@*0>^
^2<000
```

## EIXIDA

En l'eixida, has d'indicar en una única línia (amb salt de línia al final) el resultat de les operacions del tauler. Per a l'exemple anterior, l'eixida serà

14

Totes les operacions donaran un resultat sencer, positiu o negatiu. Si el tauler no té solució, has d'imprimir

## SIN SOLUCION

## PUNTUACION

- **Test 1 (10 punts):** Tots els taulers tenen solució.
- **Test 2 (20 punts):** Hi ha taulers sense solució pel fet que seguint les fletxes acabem 'eixint' del tauler sense trobar cap fletxa més ni la casella final.
- **Test 3 (30 punts):** A més de l'anterior, hi ha taulers sense solució on, encara arribant a la casella final, la seqüència d'operacions no és correcta.
- **Test 4: (40 punts):** A més de l'anterior, hi ha taulers on, sense arribar a eixir-nos del tauler, mai arribem a la casella final.

## Nombres Primers

Recordeu que un nombre enter  $n$  és primer si únicament té dos divisors: l'1 i ell mateix.

El vostre programa haurà de determinar si un número és primer i, a més, calcular una sèrie de nombres primers consecutius.

### Entrada

La primera línia contindrà un número  $t$ , que representa el nombre de casos a processar. Cada una de les següents línies contindrà dos nombres  $n$  i  $k$ , amb les següents condicions:

$$1 \leq t \leq 1000$$
$$1 \leq n \leq 10000$$
$$0 \leq k \leq 100$$

Si  $k$  és 0, s'ha de determinar si  $n$  és un nombre primer. Si  $k$  és major que 0, cal calcular els  $k$  primers nombres a partir de  $n$ , incloent-ho si aquest és primer.

### Sortida

Per a cada cas, escriviu una resposta en una línia diferent.

Si  $k$  és 0, escriviu 'SI' o 'NO' per a indicar si  $n$  és un nombre primer.

Si  $k$  és major que 0, mostreu els  $k$  primers nombres majors que  $n$ , separats per un espai.

### Puntuació

- **Test 1 (30 punts):**  $k = 0$  (determinar si un nombre és primer)
- **Test 2 (70 punts):**  $k > 0$  (listar els  $k$  primers nombres majors que  $n$ )

### Exemples

(Següent pàgina)

Entrada:

```
4
17 0
12 0
12 4
17 3
```

Sortida:

```
SI
NO
13 17 19 23
19 23 29
```