

Concierto

La organización de un festival musical está programando un concierto dentro del festival. Para ello dispone de un listado ordenado de 12 grupos musicales (o solistas, pero les llamaremos *grupo* a todos) que se han ofrecido a participar, y debe elegir sólo algunos de ellos para configurar uno de los conciertos del festival.

El **listado ordenado de grupos** es el siguiente:

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

La organización tiene las siguientes reglas para organizar el concierto:

- El grupo que comience el concierto tocará también al final y, por tanto, cerrará el concierto. Por consiguiente, este grupo tocará 2 veces. Y será el único que repita.
- De ese listado ordenado de grupos, la organización tiene que elegir un **número N de grupos (entre 2 y 8)**. Por tanto, el concierto tendrá N+1 actuaciones, ya que, como hemos comentado antes, el grupo que comienza el concierto repite al final.
- De ese listado ordenado **no se pueden escoger** grupos que **estén distanciados más de S posiciones** en la lista, siendo **S un número entre 2 y 5**.
- Los grupos se deben escoger siguiendo el orden de la lista de grupos. Es decir, [Aitana, Love of Lesbian, Dorian, The Killers, Aitana] no es una configuración válida.
- Se considera que la lista es circular, por lo que, si mi selección de grupos comenzara por *Daddy Yankee* y acabara en *The Killers*, se consideraría que entre *The Killers* y *Daddy Yankee* (que también actuaría al final) habría un salto de 5 posiciones en la lista ordenada de grupos.
- Al ser circular la lista, mi selección puede comenzar, por ejemplo, en *Love of Lesbian* y terminar, por ejemplo, en *La Casa Azul*. La lista está ordenada, pero no tiene ni inicio ni final, ya que es circular.
- No puede darse más de una vuelta a la lista circular.

Ejemplos de configuraciones de concierto válidas:

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

sería válida con $N=4$ y $S=3$.

[La Casa Azul, Love of Lesbian, Ska-P, Viva Suecia, Daddy Yankee, La Casa Azul]

sería válida con $N=5$, $S=3$.

[Ska-P, Viva Suecia, Daddy Yankee, La Casa Azul, Love of Lesbian, Ska-P] sería válida con $N=5$ y $S=4$

Formato de la entrada

Grupo que comienza el concierto

N

S

Ejemplo:

The Killers

8

5

Esto significa que buscamos un concierto con $8+1=9$ actuaciones que comienza con el grupo *The Killers* y acaba con el mismo grupo, de manera que el salto máximo en la lista circular ordenada de grupos sea de 5 posiciones.

Formato de la salida

El programa ha de imprimir el número de posibles configuraciones diferentes del concierto dados los datos de entrada proporcionados. Es posible que no exista para algunos casos ninguna configuración posible. En tal caso, la salida sería 0.

Ejemplos

Si queremos configurar un concierto que comience con *Aitana* con $N=4$ grupos (y por tanto que tenga $4+1=5$ actuaciones) con un espaciado máximo $S=3$, pondríamos en la entrada:

Aitana

4

3

Y la salida sería 1, porque la única configuración válida con estas reglas sería esta:

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

Por tanto, en este caso la especificación de entrada/salida sería:

Entrada:

Aitana 4 3

Salida:

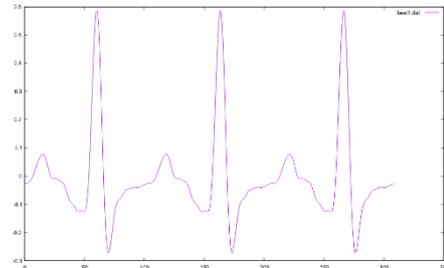
1

Puntuación

- **20 puntos:** Si se resuelve bien con $S=2$ y $N=7$.
- **30 puntos:** Si se resuelve bien con $S=2$, y cualquier valor válido de N .
- **50 puntos:** Si se resuelve bien con cualquier valor válido de N y de S

Con corazón

Hace unos días, hiciste una revisión médica y, entre otras cosas, te hicieron un electrocardiograma. Le has pedido al médico que te guarde la información del electrocardiograma en un fichero. Desafortunadamente, la señal tiene mucho ruido por lo que necesitas suavizarla y para ello vas a utilizar un método de media móvil.



La media móvil en un punto se calcula como la media aritmética de los k datos (con k impar) alrededor del punto actual, incluido el propio punto. Incluirá por tanto, $(k-1)/2$ datos antes del punto y otros $(k-1)/2$ datos después del punto.

Considerando i el índice o posición del punto actual, la fórmula para la media móvil en ese punto será:

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

Puesto que necesitamos varios datos antes y después de i , en principio la media solo la definimos para $i \geq (k-1)/2$ e $i \leq n - 1 - (k-1)/2$, considerando que los índices para los datos van de 0 a $n-1$, el resto de valores se quedan sin modificar. Por ejemplo, si los datos son: [0 2 1 4] y $k=3$, entonces la media móvil para la 2ª posición (índice 1) será $(0+2+1)/3 = 1$ y para la 3ª posición (índice 2) será $(2+1+4)/3 = 2.3333$ que redondearemos al entero más próximo, es decir, **2**, con lo que el resultado sería: [0 1 2 4]. El primer elemento (índice 0) se queda igual, puesto que no hay elemento a su izquierda para poder calcular la media y lo mismo le sucede al último elemento. Este problema se corresponderá con la opción 0 de suavizado. Los datos del electrocardiograma serán siempre datos **enteros** entre -1000 y +1000 y los datos de salida deberán ser también **enteros**, pero el resultado de la división al realizar la media deberá **redondearse al entero más próximo**.

En la opción 1, intentaremos generalizar la media para que incluya también los primeros y últimos puntos. Para ello, consideraremos que el primer y último punto se extienden indefinidamente. De esta forma, se podrá calcular la media desde $i \geq 0$ hasta $i \leq n-1$. En el ejemplo anterior, para la primera posición (índice 0), tendríamos $(0+0+2)/3 = 0.6666$ que redondeado da 1, y para la última: $(1+4+4)/3 = 3$, con lo que el resultado quedaría [1 1 2 3].

Escribe un programa que reciba como entrada la opción de media móvil a usar y la señal del electrocardiograma y devuelva dicha señal suavizada.

Entrada

La entrada constará de 2 líneas. En la primera línea estará la opción elegida para el suavizado (Op), el tamaño del vector que contiene el electrocardiograma (N) y el número de elementos a utilizar en total para realizar la media de cada punto (K) y en la segunda línea, los N números enteros representando el electrocardiograma.

$$0 \leq Op \leq 1$$

$$1 \leq N \leq 1000$$

$$1 \leq K \leq 21 \quad (\text{Solo valores impares})$$

$$-1000 \leq x_i \leq 1000 \quad (\text{Valores del electrocardiograma})$$

Salida

N números representando el electrocardiograma suavizado.

Puntuación

Test 1 (10 puntos): $Op = 0$. $K = 1$. El electrocardiograma resultante es el mismo que el original.

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

Test 3 (60 puntos): Sin restricciones.

Ejemplos

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
```

Salida

```
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
```

Salida

```
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
```



III Olimpiada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior
Universitat de València - Escola Tècnica Superior d'Enginyeria

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
```

Salida

```
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
```

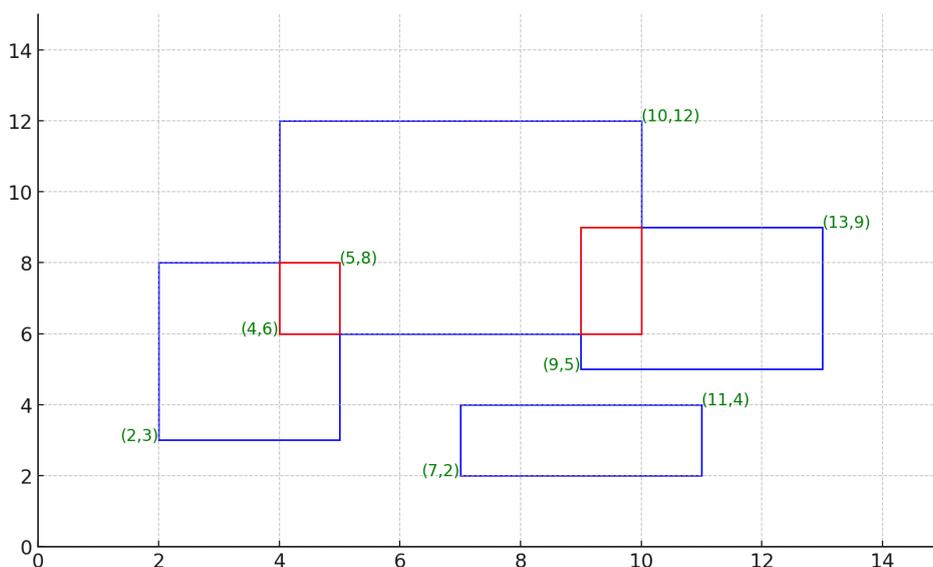
Exposición de pinturas

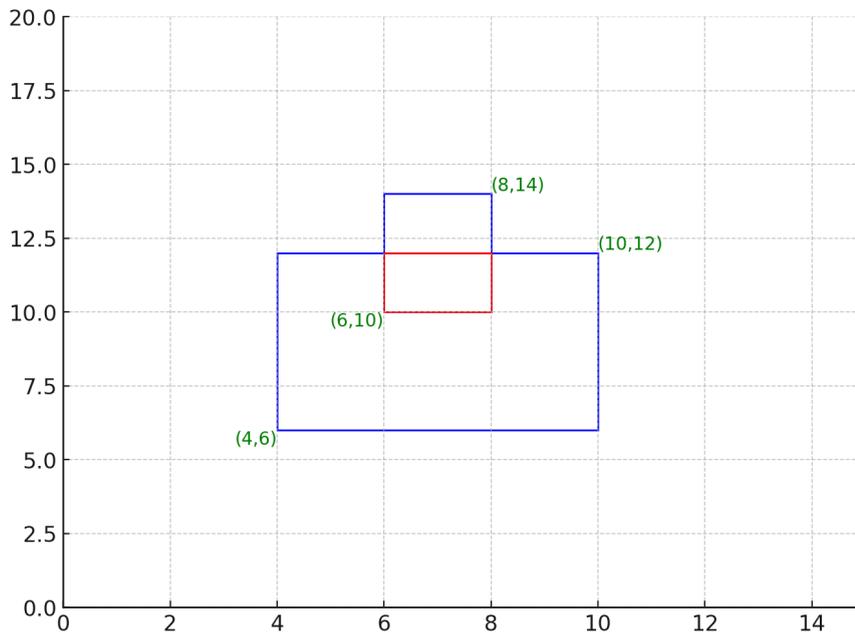
Supongamos que eres el/la coordinador/a de una exposición de arte en un prestigioso museo. La exposición incluye una serie de valiosas pinturas, cada una enmarcada en un cuadro rectangular. Estos cuadros se exhibirán en una sola pared, y debido a limitaciones de espacio, algunos cuadros pueden superponerse parcial o totalmente con otros.

Para planificar adecuadamente el espacio y la disposición de los cuadros, necesitas **calcular el perímetro total de la superficie** que ocuparán todos los cuadros en la pared, teniendo en cuenta las **superposiciones**. Esto es necesario para determinar la cantidad de material de marco necesario para el diseño de la exposición.

Objetivo: Desarrolla un programa que, dadas las dimensiones y posiciones de cada cuadro en la pared, calcule el **perímetro total ocupado** por todos los cuadros en la exposición, ajustando para no contar dos veces las partes del perímetro que se **superponen** entre cuadros.

A continuación se muestran de forma gráfica algunos ejemplos de posibles distribuciones de cuadros en la pared. El perímetro total de marco necesario para enmarcarlos será el que está representado en azul, sin tener en cuenta el perímetro de las zonas superpuestas.





ENTRADA

La primera línea contiene un entero N , en el rango $0 \leq N \leq 10$, indicando el número de cuadros en la pared.

Las siguientes N líneas contienen cuatro enteros cada uno representando las coordenadas x,y de cada cuadro (esquina inferior izquierda y esquina superior derecha, respectivamente) en relación con la pared donde se exhibirán. El rango de las coordenadas x,y será: $0 \leq x \leq 99$, $0 \leq y \leq 99$.

SALIDA

Un único entero que representa el perímetro total de la superficie que ocuparán todos los cuadros, teniendo en cuenta las superposiciones.

Consideraciones:

- Los rectángulos siempre estarán orientados de tal manera que sus lados sean paralelos a los ejes del sistema de coordenadas. No hay rotaciones del rectángulo en ningún ángulo.
- Consideramos que en una intersección solo hay dos rectángulos involucrados.

Ejemplos

Ejemplo 1 (el de la imagen):

Datos de entrada:

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

Salida:

54

Ejemplo 2 (sin superposición):

Datos de entrada:

2
1 1 4 4
5 5 8 8

Salida:

24

Ejemplo 3 (superposición parcial):

Datos de entrada:

2
1 1 5 5
3 3 8 8

Salida:

28

Ejemplo 4 (rectángulos anidados):

Datos de entrada:

2
2 2 7 7
3 3 6 6

Salida:

20

Puntuación

- **Test 1 (20 puntos):** Se resuelven bien los casos en los que los rectángulos no intersectan entre sí.
- **Test 2 (80 puntos):** Se resuelven bien los casos en los que hay intersecciones entre varios rectángulos y/o anidamientos.

LABERINTO ARITMÉTICO

Nuestro profesor de matemáticas, en un intento por hacernos aprender aritmética de una forma amena, ha inventado los laberintos aritméticos, como el que ves aquí:

	→		3	↓	
→	↓			+	
=		↓	4		←
	@	x		→	↑
↑	2	←			

Empezando por la flecha blanca, y siguiendo la dirección que marca, sigue en línea recta. Cuando encuentres una flecha negra, cambia la dirección en el sentido indicado por la flecha, y continúa así hasta llegar a la casilla final, indicada por el símbolo '@'. Por el camino irás encontrando números y operaciones, que deberás ir resolviendo a medida que avances, hasta llegar a la casilla final. Indica el resultado final de las operaciones.

En este ejemplo, el resultado es $(3 + 4) * 2 = 14$

Sólo hemos pasado una vez por cada operando u operador en este ejemplo, pero es posible que en algunos tableros se pase más de una vez por el mismo operando u operador (por ejemplo, la primera vez pasamos verticalmente y la segunda horizontalmente). La secuencia de operaciones correcta más corta posible es un dígito seguido del signo igual (y por supuesto, @ al final). El signo igual solo aparece al final de la secuencia de operaciones.

Es posible que algunos tableros estén mal diseñados y no tengan solución, porque siguiendo las flechas nunca se llega a la casilla destino, o bien la secuencia de operaciones no alterna operandos y operadores, comenzando y terminando por un operando, seguido del signo '='.

ENTRADA

La primera línea indica el número de filas (N) y el número de columnas del tablero (M). Ambos números son enteros entre 3 y 10. La segunda línea indica las coordenadas de la casilla de salida (la esquina superior izquierda del tablero tiene la coordenada (0,0)). A continuación, el tablero se representa mediante N filas de longitud M. Cada fila contiene M caracteres (sin espacios en blanco entre ellos) que representa el contenido de una fila del tablero. Las flechas (tanto la inicial como el resto se representan mediante los siguientes caracteres:

flecha arriba: '^'
flecha abajo: 'v'
flecha hacia la izquierda: '<'
flecha hacia la derecha: '>'

El resto de caracteres indican números del 1 a 9, operaciones aritméticas ('+', '-', '*', '/', '='), la casilla final ('@') o un casilla vacía ('Ø'). Asume que en la casilla de salida siempre habrá una flecha.

La siguiente entrada equivale al ejemplo anterior:

```
5 6
Ø 1
Ø>Ø3vØ
>vØØ+Ø
=Øv4Ø<
Ø@*Ø>^
^2<ØØØ
```

SALIDA

En la salida, debes indicar en una única línea (con salto de línea al final) el resultado de las operaciones del tablero. Para el ejemplo anterior, la salida será

14

Todas las operaciones darán un resultado entero, positivo o negativo. Si el tablero no tiene solución, debes imprimir

SIN SOLUCION

PUNTUACION

- **Test 1 (10 puntos):** Todos los tableros tienen solución.
- **Test 2 (20 puntos):** Hay tableros sin solución debido a que siguiendo las flechas acabamos 'saliendo' del tablero sin encontrar ninguna flecha más ni la casilla final.
- **Test 3 (30 puntos):** Además de lo anterior, hay tableros sin solución donde, aún llegando a la casilla final, la secuencia de operaciones no es correcta.
- **Test 4: (40 puntos):** Además de lo anterior, hay tableros donde, sin llegar a salirnos del tablero, nunca llegamos a la casilla final.

Primos

Recuerda que un número entero n es primo cuando sus únicos divisores son el 1 y el propio número.

Tendrás que hacer un programa que decida si un número es primo o no y que calcule un número de primos consecutivos.

Entrada

La entrada comienza con un número t que indica el número de casos a procesar. Cada una de las siguientes líneas contiene dos números n y k .

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

Si el número k es 0, el número n indica un número del que debemos calcular si es primo o no. Si el número k es mayor de 0, indica el número de primos que debemos imprimir a partir del número inicial n .

Salida

Se deberá escribir una línea con cada respuesta.

En el caso en que k es 0 se deberá escribir SI (sin acento) o NO indicando si el número n es primo o no.

En el caso en que k sea mayor que 0 se deberán escribir los k números primos mayores que n separados por un espacio.

Puntuación

- **Test 1 (30 puntos):** $k = 0$ (comprobar si un número es o no primo)
- **Test 2 (70 puntos):** $k > 0$ (escribir los k números primos mayores que n)

Ejemplos

(Siguiendo página)

Entrada:

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

Salida:

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