

## Minimizando el tiempo de alarma

*Contribución de Alejandro Deymonnaz*

### Descripción del problema

Una ciudad medieval rodeada de un muro tiene **T** torres de vigilancia en distintos puntos de su extensión. Cada torre tiene asignado un guardia que vigila el predio.

Con el fin de dar la alarma rápidamente en caso de ataque del enemigo, se dispone de **C** campanas a ser instaladas en sendas torres de vigilancia.

Cuando un guardia avista presencia enemiga deberá correr por el muro hasta la torre con campana más cercana para dar aviso.

Para que este procedimiento resulte eficiente se deben elegir adecuadamente las **C** torres que tendrán una campana, de manera de minimizar el tiempo máximo que se pueda tardar desde que un guardia descubre al enemigo hasta que suena una campana.

Se conocen los tiempos que tardan los guardias en correr de una torre a otra vecina, a lo largo del muro que las une.

Se te pide que mediante un programa, **al arma.cpp**, **al arma.pas** o **al arma.c**, ayudes a la ciudad hallando una solución que minimice el tiempo máximo en dar la alarma.

### Datos de entrada

Se recibe un archivo **al arma.in** del directorio actual, que contiene:

- Primer renglón: la cantidad **T** de torres de vigilancia y cantidad **C** de campanas de alarma a instalar. ( $3 \leq T \leq 500.000$ ), ( $1 \leq C \leq T$ )
- En los siguientes **T** renglones, vienen los tiempos necesarios para recorrer los muros entre torres, primero el tiempo para el muro que une la torre **1** con la **2**, luego el tiempo para el muro que une la torre **2** con la **3**, etc. En el último renglón el tiempo para el muro que une la torre **T** con la torre **1**. El tiempo necesario para recorrer el perímetro completo no supera a **1.000.000.000**. Los tiempos son números enteros.

### Datos de salida

El programa debe generar el archivo **al arma.out**, en el directorio actual con:

- Un renglón con un entero indicando cuanto es el máximo tiempo que puede necesitar un guardia en transmitir una alarma tras optimizar la elección de la ubicación de las campanas.
- Una línea con **C** números que describa el conjunto de torres con campana que configuren una solución optimal.
- Una línea con **T** caracteres, uno por torre con la siguiente convención:
  - 0 : no se mueve (tiene una campana)
  - + : debe correr en sentido horario
  - : debe correr en sentido antihorario.
 Las torres fueron numeradas en sentido horario. Si fuera indistinto se acepta ambos valores.

### Puntuación parcial

La solución correcta de la primera línea de la respuesta vale **50** puntos, de la segunda **30**, de la tercera **20**.

### Ejemplo

Si el archivo **al arma.in** contiene:

```
8 3
10
17
3
6
13
8
19
11
```

El archivo **al arma.out** podría contener:

```
11
5 7 1
0-++0+0+
```