



16o. Concurso de Programación  
Facultad de Ingeniería  
Universidad ORT Uruguay

Setiembre 2016

**EJERCICIO 1: FUTBOL**

Casi todos tenemos nuestra propia lista de mejores jugadores de fútbol del mundo. Como es algo subjetivo, las listas pueden diferir, pero usualmente son similares. Se desean comparar dos de esas listas y determinar su similitud. Para ello, deben partirse en conjuntos, donde cada conjunto represente el mismo rango consecutivo de posiciones en ambas listas, y tienen que contener los mismos nombres dentro de ese rango, siendo este rango lo más pequeño posible. Deben informarse los tamaños respectivos de esos conjuntos. Si las dos listas son idénticas y contienen  $n$  nombres, hay  $n$  conjuntos, todos de tamaño 1.

Ejemplos:

a)

SUAREZ FORLAN PELE MARADONA MESSI	SUAREZ PELE MARADONA FORLAN MESSI
---	---

Serían 3 conjuntos:

- SUAREZ
- FORLAN MARADONA PELE
- MESSI

Los tamaños son: 1 3 1

b)

FORLAN SUAREZ PELE MARADONA MESSI NEYMAR RONALDO	SUAREZ MESSI MARADONA FORLAN PELE RONALDO NEYMAR
--	--

Serían 2 conjuntos:

- FORLAN SUAREZ PELE MARADONA MESSI
- NEYMAR RONALDO

Los tamaños son: 5 2

**Archivo de Entrada:**

La primera línea contiene el número  $N$  ( $1 \leq N \leq 100$ ) de casos. Luego viene la información de cada caso. La primera línea de cada caso contiene la cantidad  $J$  de jugadores en la lista ( $1 \leq J \leq 1.000.000$ ). Las siguientes  $J$  líneas contienen cada una un nombre de un jugador de la primera lista. Luego de estas  $J$  líneas, hay  $J$  líneas más, cada una contiene un nombre de un jugador de la segunda lista. Todos los nombres están en mayúsculas y tienen hasta 8 letras del alfabeto inglés. Las dos listas tienen los mismos jugadores y no hay jugadores con el mismo nombre. Naturalmente, un jugador no está más de una vez en una lista.

**Ejemplo de Entrada:**

4  
5  
SUAREZ  
FORLAN  
PELE  
MARADONA  
MESSI  
SUAREZ  
PELE  
MARADONA  
FORLAN  
MESSI  
3  
PELE  
MARADONA  
SUAREZ  
PELE  
MARADONA  
SUAREZ  
3  
RONALDO  
TEVEZ  
GODIN  
TEVEZ  
GODIN  
RONALDO  
3  
ICARDI  
LOPEZ  
MARADONA  
LOPEZ  
ICARDI  
MARADONA

**Archivo de Salida:**

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. Indicar el tamaño de cada conjunto en orden, separando los números con un único espacio en blanco.

**Ejemplo de Salida:**

Caso 1: 1 3 1  
Caso 2: 1 1 1  
Caso 3: 3  
Caso 4: 2 1





16o. Concurso de Programación  
Facultad de Ingeniería  
Universidad ORT Uruguay

Setiembre 2016

**EJERCICIO 2: GANGAS**

Gangas, una importante tienda internacional, está de ¡liquidación! Decidieron vender todas las prendas con un 25% de descuento, o sea, al 75% del precio normal. Todos sus precios normales son enteros y divisibles por cuatro, así que los precios con descuento también son enteros.

Para preparar la gran liquidación, mandaron imprimir una etiqueta para cada prenda con su valor nuevo. También encargaron etiquetas con los precios normales para cada prenda, para usar luego de la liquidación.

Por error, la imprenta mandó las etiquetas en una única pila, ordenada por precio creciente: la etiqueta del precio normal y la con descuento de cada prenda están en ¡alguna parte de esa pila!

Las etiquetas son todas del mismo formato, así que no se pueden distinguir cuáles corresponden a precio con descuento y cuáles a precio normal. Por ejemplo, si los precios normales son: 20, 80 y 100, los precios con descuento son 15, 60 y 75. La pila contiene: 15, 20, 60, 75, 80 y 100.

Ayudemos a Gangas a identificar las etiquetas de precios con descuento.

**Archivo de Entrada:**

El archivo contiene múltiples casos. La primera línea contiene el número  $T$  de casos ( $1 \leq T \leq 100$ ). Cada caso consiste en 2 líneas. La primera tiene la cantidad  $N$  ( $1 \leq N \leq 100$ ) de prendas en Gangas. La siguiente línea contiene  $2N$  enteros  $P_i$ , cada uno corresponde al precio impreso en cada etiqueta ( $1 \leq P_i \leq 10^9$ , se cumple para todo  $i$ :  $P_i \leq P_{i+1}$ , esto es, están en orden creciente).

Se garantiza que existe solución única.

**Ejemplo de Entrada:**

```
2
3
15 20 60 75 80 100
4
9 9 12 12 12 15 16 20
```

Nota: observar que pueden haber varias prendas que tengan el mismo precio, así como prendas que su precio normal sea igual al precio de descuento de otra prenda.


**Archivo de Salida:**

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. Luego mostrar los precios de descuento, en orden creciente.

**Ejemplo de Salida:**

```
Caso 1: 15 60 75
Caso 2: 9 9 12 15
```



	<p style="text-align: center;">16o. Concurso de Programación Facultad de Ingeniería Universidad ORT Uruguay</p> <p style="text-align: right;">Setiembre 2016</p> <p style="text-align: center;"><b>EJERCICIO 3: LASER</b></p>
---	---

En el Laboratorio de Robótica están probando un nuevo robot que tiene un laser para cortar chapas. El robot tiene 2 operaciones:

- moverse en línea recta (cortando la superficie debajo de ella a medida que el robot se mueve) y
- girar en el mismo lugar para apuntar a otra dirección.

Si el robot corta un hueco en la chapa, podría caer ahí. Nuestra tarea es, dado un conjunto de instrucciones para el robot, indicar si resultará en hacer un hueco (esto es, el camino recorrido se intersecta a sí mismo en algún lugar) o no. Se asume que la superficie a cortar es infinita y que el ancho del corte es despreciable.

El robot se ubica sobre un plano cartesiano, comienza siempre en la coordenada (0,0) y está orientado hacia arriba, en la dirección Y positiva. Las instrucciones se alternan: "GIRAR" y "MOVER". Todas las instrucciones son relativas, por ejemplo "GIRAR" es girar una cierta cantidad de grados en sentido antihorario relativo a la dirección actual.

Ejemplo:

GIRAR -90	El robot hará un giro de -90°. El robot queda en (0, 0), pero mirando hacia la derecha, esto es, dirección X positiva.
MOVER 10	Se mueve 10 unidades hacia donde está apuntando. Lleva el robot a la posición (10, 0).
GIRAR 90	Gira 90°. El robot está en (10, 0) y apuntará hacia arriba, en la dirección Y positiva.
MOVER 5	Se mueve 5 unidades, llega a (10, 5). (Ver figura 1)
GIRAR 135	Gira 135°. Ahora el robot apunta diagonalmente a la dirección XY negativa.
MOVER 10	Intenta moverse 10 unidades. Si se completara esta instrucción, llevaría el robot a aproximadamente (2.93, -2.07). Al moverse hacia allí, el robot intersectaría uno de los cortes previos, haciendo un hueco (Ver figura 2). Esta instrucción y las siguientes no pueden ser completadas.
GIRAR -90	
MOVER 5	

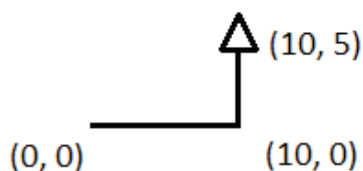


Figura 1: luego de 2 instrucciones GIRAR/MOVER

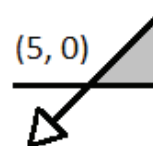


Figura 2: luego de 3 instrucciones GIRAR/MOVER

### Archivo de Entrada:

El archivo contiene múltiples casos. La primera línea contiene la cantidad **C** de casos ( $1 \leq C \leq 20$ ). Cada caso consiste en varias líneas. La primera tiene la cantidad **N** de instrucciones GIRAR/MOVER a ejecutar ( $1 \leq N \leq 100$ ). Las siguientes **N** líneas contienen las intrucciones. Cada instrucción contiene 2 enteros **T** y **M**, donde **T** es el número de grados a girar ( $-179 \leq T \leq 179$ ) y **M** es la distancia a moverse luego del giro ( $1 \leq M \leq 100$ ).

Observar que, aunque T y M son enteros, no se puede asumir que la posición del robot sea siempre en coordenadas enteras. Los datos de prueba han sido elegidos de forma que las líneas que intersectan lo hagan lejos de los extremos y que las líneas que no intersectan queden suficientemente separadas de las otras.

#### **Ejemplo de Entrada:**

```
3
4
-90 10
90 5
135 10
-90 5
4
-90 10
90 5
135 5
-90 10
6
60 10
60 10
60 10
60 10
60 10
60 10
80 20
```

#### **Archivo de Salida:**

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. En el caso, si se puede ejecutar el conjunto completo de instrucciones sin hacer un agujero, incluir la palabra en mayúsculas "SEGURO". Si el caso crea un hueco, indicar el número de la primera instrucción MOVER que lo crea (ej. 3 si el tercer MOVER crea el agujero). Observar que una vez detectado un hueco, las demás instrucciones del caso son irrelevantes.

#### **Ejemplo de Salida:**

```
Caso 1: 3
Caso 2: SEGURO
Caso 3: 6
```





16o. Concurso de Programación  
Facultad de Ingeniería  
Universidad ORT Uruguay

Setiembre 2016

**EJERCICIO 4: PRIMOS**

Juan, estudiante de Matemática, ha desafiado a Luis, estudiante de Computación, a resolver un problema relativo a números primos. Juan le recuerda a Luis que un número primo es un entero mayor que 1 donde sus únicos divisores son 1 y él mismo; el número 1 no se considera primo.

Se da una secuencia de  $n$  enteros positivos. La secuencia debe ser partida en  $k$  regiones contiguas consecutivas, con al menos un entero en cada región. Luego de encontrar una partición, se le calcula su puntaje de una forma particular. En cada región, se debe calcular el mayor número primo (que no necesariamente pertenece a la región) que divida a todos y cada uno de los números de esa región. Si no hay tal primo en la región, entonces esa región tiene puntaje 0. Si lo hay, el puntaje de la región es ese primo. El puntaje total de la partición es el mínimo de todas sus regiones.

El desafío para Luis es encontrar el máximo puntaje posible. ¿Podemos ayudarlo?

**Archivo de Entrada:**

El archivo contiene múltiples casos. La primera línea contiene la cantidad  $c$  de casos ( $1 \leq c \leq 25$ ). Cada caso consiste en 2 líneas. La primera tiene dos enteros:  $n$  ( $1 \leq n \leq 20.000$ ) y  $k$  ( $1 \leq k \leq \min(100, n)$ );  $n$  representa el número de enteros positivos en la secuencia a procesar y  $k$  es el número de regiones en la partición. La siguiente línea contiene la secuencia a ser partida. Contiene  $n$  enteros  $v$  ( $1 \leq v \leq 1.000.000$ ).

**Ejemplo de Entrada:**

```
2
5 3
10 5 4 8 3
5 3
10 11 12 13 14
```

**Archivo de Salida:**

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. En el caso, indicar un entero que represente el puntaje máximo de partir la secuencia de  $n$  enteros positivos en  $k$  regiones.

**Ejemplo de Salida:**

```
Caso 1: 2
Caso 2: 0
```





16o. Concurso de Programación  
Facultad de Ingeniería  
Universidad ORT Uruguay

Setiembre 2016

**EJERCICIO 5: SETS**

Para una importante empresa, una de las habilidades que más valoran de sus empleados es la capacidad de auto-organizarse en “sets” o grupos, según ciertas reglas. En este caso, la única regla a aplicar es que, en cada grupo, todos sus miembros deben cumplir que, o bien: a) tienen la misma inicial de su nombre o bien b) tienen la misma inicial de su apellido. Para aumentar el desafío, la cantidad de grupos debe ser la menor posible.

Por ejemplo, si se tienen estas personas: Sandra Castro, Vicente Coll, Edmundo Clark, Juan Perez, Sergio Garcia, y Silvia Miguez, se podrían auto-organizar en 3 sets:

- Sandra Castro, Vicente Coll, y Edmundo Clark (todos sus apellidos empiezan con C)
- Sergio Garcia y Silvia Miguez (todos sus nombres empiezan con S)
- Juan Perez (no es un grupo muy interesante pero cumple que todos sus integrantes –uno sólo en este caso- tiene su nombre que comienza con la misma inicial: J)

Se pide, a partir de los datos de los empleados, indicar cuál es la menor cantidad posible de sets.

**Archivo de Entrada:**

El archivo contiene múltiples casos. La primera línea del archivo contiene la cantidad **C** de casos ( $1 \leq C \leq 20$ ). Cada caso consiste en varias líneas. La primera de ellas tiene la cantidad **N** de personas ( $1 \leq N \leq 300$ ). Luego vienen **N** líneas, cada una corresponde a los datos de una persona. De cada persona se tiene su nombre y su apellido, separados por un espacio en blanco. El nombre y el apellido son strings, comienzan por mayúscula y están formados por letras del alfabeto inglés. En particular, los apellidos comienzan con una letra exclusivamente del rango “A” a “R” inclusive (esta característica no aplica a los nombres, los que pueden comenzar con cualquier letra del alfabeto inglés).

**Ejemplo de Entrada:**

2

6

Sandra Castro

Vicente Coll

Edmundo Clark

Juan Perez

Sergio Garcia

Silvia Miguez

9

Richard Hamming

Mickey Mouse

John McCarthy

Ed Dijkstra

Donald Knuth

Miguel Rios

John Backus

Roberto Fernandez

Tony Hoare



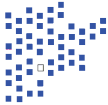
**Archivo de Salida:**

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. Incluir la cantidad mínima de sets del caso.

**Ejemplo de Salida:**

Caso 1: 3

Caso 2: 6

	<p style="text-align: center;">16o. Concurso de Programación Facultad de Ingeniería Universidad ORT Uruguay</p> <p style="text-align: right;">Setiembre 2016</p> <p style="text-align: center;"><b>EJERCICIO 6: TARJETAS</b></p>
---	--

Anita colecciona tarjetas numeradas de Los Simpson. Como es muy generosa, quiere regalarle una tarjeta a cada una de sus tres amigas. Sus tres amigas son especiales y requieren que, cuando se les regale una tarjeta, se cumpla que:

- a) Cada par de amigas no pueden recibir el mismo número de tarjeta
- b) Los números de las tarjetas recibidas entre cada par de amigas no pueden diferir en más de 2.

Por ejemplo, Anita tiene las tarjetas 4, 5, 3, 90, 5, 5, 91 y 93. No podría regalarles las tarjetas 5, 5 y 4, porque rompería la primera regla. Tampoco podría regalarles las tarjetas 90, 91 y 93 porque rompería la segunda regla.

Ayudemos a Anita a verificar si puede elegir, o no, 3 de sus tarjetas que cumplan las reglas.

#### Archivo de Entrada:

El archivo contiene múltiples casos. La primera línea contiene la cantidad **C** de casos ( $1 \leq C \leq 100$ ). Cada caso consiste en 2 líneas. La primera contiene un entero **N** ( $3 \leq N \leq 50$ ) que corresponde a la cantidad de tarjetas que tiene Anita. La siguiente línea contiene N enteros:  $t_1, t_2, \dots, t_N$  ( $1 \leq t_i \leq 1000$ ) que corresponde cada uno a un número de tarjeta.

#### Ejemplo de Entrada:

```
3
4
18 55 16 17
6
40 44 43 44 44 41
8
5 972 3 4 1 4 970 971
```

#### Archivo de Salida:

Por cada uno de los casos, imprimir una línea con "Caso x: ", siendo x el número de caso, empezando desde 1. Luego incluir en mayúsculas la palabra "SI" si Anita puede elegir 3 tarjetas que cumplan las reglas dadas o la palabra "NO" si no es posible.

#### Ejemplo de Salida:

```
Caso 1: SI
Caso 2: NO
Caso 3: SI
```

