

### A- Ordenando mi directorio

Elly quiere escribir un programa que liste todos los archivos de un directorio dado. Ella ya tiene la lista de todos los archivos.

Ella te proporcionará esta lista en un vector de archivos (Strings). Adicionalmente tendrás 2 elementos en el vector: el Actual directorio con el String ("."), y el directorio padre con el String ("..").

Estos dos elementos pueden estar en cualquier lugar de los archivos. Sin embargo Elly quiere que (".", y "..") Sean los últimos dos elementos del vector. Entonces Elly tiene algunas reglas para poder mover los dos directorios a las últimas posiciones:

1. Si "." y ".." son los 2 últimos elementos del vector (en cualquier orden) el problema está resuelto.  
Caso Contrario:
2. Encontrar el primer elemento, ya sea "." o ".." e intercambiarlo con el último elemento de los archivos.
3. Si "." y ".." son ahora los últimos elementos de archivos (cualquier orden) el problema está resuelto  
Si no:
4. Buscar el primer elemento, ya sea "." o ".." e intercambiarlo con el penúltimo elemento de los archivos.

Con estos pasos se garantiza que el problema será resuelto

#### Entrada.-

Se te dará un entero T ( $1 \leq T \leq 100$ ) el número de casos de prueba, en la siguiente línea tendrás un entero N ( $1 \leq N \leq 50$ ) el número de elementos en el vector de Strings.

Después vendrán N líneas, el nombre de cada archivo. (Se garantiza que los Strings "." y ".." estarán en el vector dado y ningún string tendrá espacio entre ellos).

#### Salida.-

De acuerdo a cada caso de Prueba imprimir la palabra "Caso: #" donde # debe ser el respectivo número de entrada (1.. T)

Realizando los pasos que Elly te pidió imprimir el vector de Strings, cada string uno en una línea. Entre cada caso de prueba debe existir un espacio en blanco

**Ejemplo de Entrada.-**

```
2
6
ContestApplet.jnlp
.
Image.jpg
Book.pdf
Movie.avi
..
8
No
..
Zaphod
.
Just
Very
very...
Improbable
```

**Ejemplo de Salida**

```
Caso 1:
ContestApplet.jnlp
Movie.avi
Image.jpg
Book.pdf
..
.

Caso 2:
No
Improbable
Zaphod
very...
Just
very
.
..
```

**B- ¡Come todo el Helado que puedas!**

¿Quién no comió helado alguna vez?, si no lo hiciste no sabes de lo que te pierdes.

José es un muchacho que ama los helados y siempre soñó con comer helado todo el día, por suerte él fue ahorrando dinero para poder ir al gran Palacio de los helados y comer todo el helado que pueda.

El palacio de los helados cuenta con infinidad de sabores como Chocolate, Vainilla, Fresa, Naranja, etc. etc. y cada sabor viene en diferentes porciones y un precio. Por ejemplo podemos encontrar el sabor de chocolate en 3 porciones diferentes a 10Bs, 15Bs y 25Bs respectivamente.

José al llegar al palacio de los helados y ver los precios notó que no cuenta con suficiente dinero para poder comprar todo el helado que hay. Entonces el estará conforme con probar un helado de cada sabor.

Él quiere comer todos los sabores que hay y no sabe si el dinero le va a alcanzar. Así que pide tu ayuda. Él quiere gastar el máximo monto de dinero posible con la condición de probar todos los sabores que hay.

**Entrada.-**

La primera línea contiene un entero  $N$ , indicando el número de casos de prueba que hay. Para cada caso de prueba la primera línea tendrá 2 enteros  $M$  y  $C$ . separados por un espacio en blanco ( $1 \leq M \leq 200$ ), and ( $1 \leq C \leq 20$ ), donde  $M$  es la cantidad de dinero que pudo ahorrar José y  $C$  es la cantidad de sabores que hay.

Seguidamente existirá  $C$  líneas, de cada una de estas líneas te darán un entero  $K$  ( $1 \leq K \leq 20$ ), indicando el número de porciones del sabor que hay, seguido por  $K$  enteros indicando el precio de la porción del sabor  $C_i$ .

**Salida.-**

Para cada caso de prueba debe imprimir el máximo monto de dinero que José gaste en probar todos los sabores de helados (claro sin exceder este monto). Si no es posible probar todos los sabores que hay, debes imprimir "No es posible".

**Entrada.-**

```
3
100 4
3 8 6 4
2 5 10
4 1 3 3 7
4 50 14 23 8
20 3
3 4 6 8
2 5 10
4 1 3 5 5
5 3
3 6 4 8
```

2 10 6  
4 7 3 1 7

**Salida.-**

75  
19  
No es posible

**Para el primer caso de prueba:**

José compra las porciones más caras de cada sabor:  $8 + 10 + 7 + 50 = 75$

De esa manera cumple con su objetivo. Probar todos los sabores y a la vez gasta la gasta la máxima cantidad de dinero que tiene.

**Para el segundo caso de Prueba:**

José con los 20Bs. que tiene compra la porción más cara del primer y segundo sabor:  $8 + 10$ .

Le sobran 2Bs. y tiene 1, 3, 5 y 5 Bs para escoger. Pero no puede escoger 2Bs ni 5Bs, pues se pasaría del monto que tiene. Así que escoge 1Bs.

El monto máximo que gasta es 19Bs. también existe otra solución tomando los precios  $4 + 10 + 5 = 19Bs$ .

**Para el tercer caso de prueba:**

Solo tiene 5Bs para gastar y no es posible que José pueda probar todos los sabores.

### C- Mi música Favorita

Tendrás que hacer un viaje muy largo a otra ciudad, así que una muy buena compañía será tu música favorita. Tu auto cuenta con una lector de casetes, pero existe un pequeño inconveniente ya que tu música favorita se encuentra en CD's. Así que es necesario grabar tu música del CD al casete.

Tú tienes  $N$  minutos disponibles en el casete (Ya que existe el límite de la cinta de casete). Así que el objetivo del problema es llenar el máximo número de minutos posibles en la cinta del casete.

#### Límites del problema:

$N$  es un entero ( $1 \leq N \leq 1500$ )

$T$  es un entero ( $1 \leq T \leq 20$ )

Ningún  $T_i$  será más grande que  $N$

#### Entrada.-

La entrada consiste en un entero  $N$  (minutos que soporta la cinta del casete), seguidamente por un entero  $T$  (Numero de pistas que tiene el CD) seguidamente por  $T$  enteros que representa el tiempo de duración de cada pista.

#### Salida.-

Imprimir las pistas (la duración de cada pista) que maximizan el uso de la cinta del casete.

#### Ejemplo de Entrada.-

```
5 3 1 3 4
10 4 9 8 4 2
20 4 10 5 7 4
90 8 10 23 1 2 3 4 5 7
45 8 4 10 44 43 12 9 8 2
```

#### Ejemplo de Salida.-

```
1 4
8 2
10 5 4
10 23 1 2 3 4 5 7
4 10 12 9 8 2
```

En el caso 1: La manera óptima de llenar la cinta del casete es tomando 1 y 4 así llenamos completamente la cinta con 5 minutos

En el caso 3: Tenemos 20 minutos disponibles en la cinta del casete, pero no es posible llenar toda la cinta, así que escogemos las pistas que pueden llenar al máximo la cinta. En este caso 10, 5 y 4 que suman 19.

### D- La Ultima Esperanza

Los Terrans están siendo atacados por los Protoss una raza de humanoides que aparecen en la serie Star Craft.

Los Terrans ya casi sin tropas y con poca estructuras solo tienen una esperanza para vencer en esta guerra contra los Protoss. Y esa esperanza es poder usar el arma más poderosa que tienen, "**la Bomba Nuclear**".

Para tal proposito necesitan un soldado que pueda marcar el blanco para el ataque nuclear y ese soldado es **Ghost**, un soldado capaz de volverse invisible y poder infiltrarse en la base de los Protoss

Ghost se dirige hacia la Base de los Protoss y logra infiltrarse en ella para así marcar el blanco del ataque Nuclear.

Ghost está tiene dudas al marcar el blanco para el ataque nuclear así que el pide tu ayuda para este problema.

Ghost te enviara coordenadas de cada estructura de la base Protoss. Así que él quiere saber el número de estructuras que serán destruidas dadas las coordenadas del blanco de la bomba nuclear y el radio de alcance de esta.

#### Entrada.-

La primera línea es un entero  $T$  ( $1 \leq T \leq 100$ ) el número de casos de prueba. Seguidamente se te dará un entero  $N$  ( $1 \leq N \leq 100$ ) el número de coordenadas  $x$ ,  $y$  que te enviara Ghost.

A continuación vendrán  $N$  líneas, cada una de 2 valores  $x$ ,  $y$  ( $0 \leq x, y \leq 1000$ ) donde cada una representa la coordenada de una estructura Protoss.

La última Línea tendrá 3 valores  $x$ ,  $y$ ,  $r$ . Donde  $x$ ,  $y$  son las coordenadas del blanco del ataque nuclear y  $r$  ( $1 \leq r \leq 500$ ) el radio de alcance de la bomba Nuclear.

#### Salida.-

Para cada caso de Prueba imprimir el número de estructuras Protoss destruidas por la bomba Nuclear (Una estructura será destruida si la coordenada de la estructura Protoss está en el radio de alcance de la Bomba Nuclear)

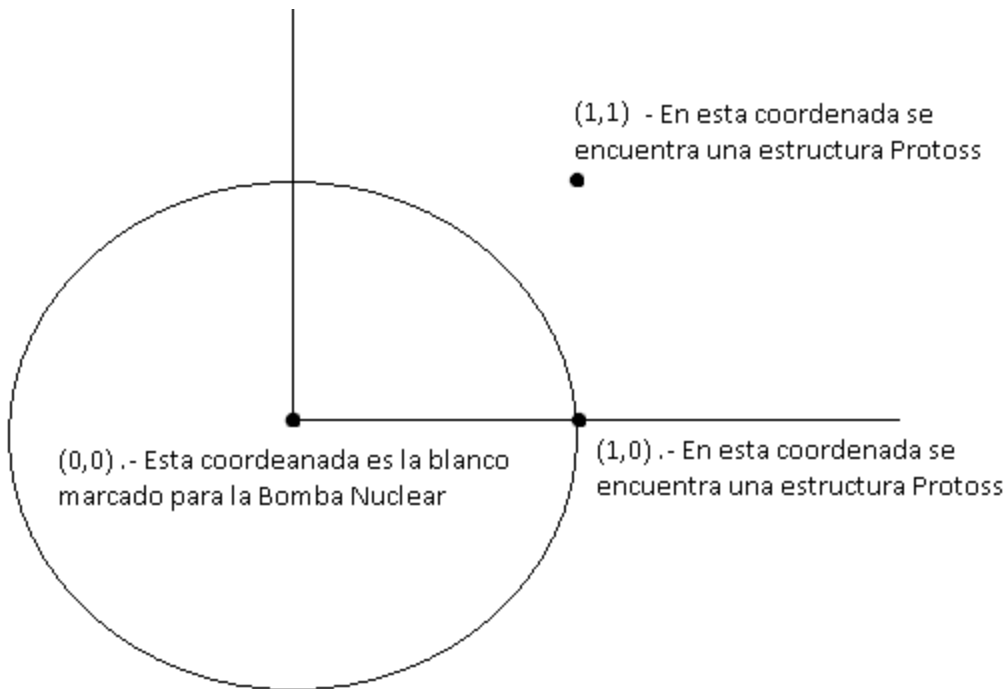
#### Entrada.-

```
3
2
1 1
1 0
0 0 1
2
1 1
1 0
0 0 5
```

2  
0 0  
50 50  
5 5 10

**Salida**

1  
2  
1



El gráfico es corresponde al primer caso de prueba, donde claramente podemos notar que el alcance del radio de la bomba nuclear llega a tocar el punto (1, 0) **(Pues el distancia del punto de la estructura Protoss a la bomba es menor e igual al radio de alcance de la bomba)**, por lo tanto la Bomba destruye una estructura, por otro lado la estructura que se encuentra en el punto (1, 1) no es alcanzada por la bomba Nuclear.

### E- UMSA WORD

La organización de la ICPC BOLIVIA hizo hacer letras grandes, para usarlas como publicidad en el evento.

Mientras transportaban el paquete que almacenaba las letras este se rompió y dejó caer las letras, al caer las letras varias de estas se perdieron. Así que ellos no están seguros si con las letras restantes podrán armar las palabras que ellos quieren.

Por ejemplo si tenemos las letras U, S, M, A, C, D, I, C, P, C y queremos armar la palabra ICPC UMSA, notaremos claramente que es posible armarla. En cambio si tuviéramos las letras U, M, S, D, G, I, I, C, P no será posible armar la palabra ICPC UMSA pues faltaría la letra "A".

Los organizadores piden tu ayuda. Ellos te darán las letras disponibles y quieren saber si es posible armar la palabra UMSA ICPC.

Entrada.-

La primera línea contendrá un entero N ( $1 \leq N \leq 10$ ) el número de casos de prueba. A continuación se te dará N líneas, cada una con una cadena (solo mayúsculas), donde cada carácter representa la letra disponible para armar las palabras ICPC UMSA.

Salida.-

Para cada Cadena que se te dio anteriormente, si es posible armar las palabras ICPC UMSA imprimir ES POSIBLE, caso contrario NO ES POSIBLE.

**Ejemplo de Entrada.-**

```
3
UMSAICPC
ICPCUMSA
ICPCUMSHJK
```

**Ejemplo de Salida-**

```
ES POSIBLE
ES POSIBLE
NO ES POSIBLE
```

## F- Futbol

Como es de conocimiento, el futbol es el deporte más jugado en el mundo.

El juego consiste de 2 equipos, cada uno de 11 jugadores y un balón.

Pepe amante del futbol armo un equipo para poder competir en el campeonato del barrio. Ahora el planea diferentes tácticas para poder ganar cada partido que les toque disputar.

Entre todas las tacitas que él tiene, una es:

Realizar pases precisos desde un jugador A a un jugador B pasando por diferentes jugadores del equipo, pero la condición es que al pasar el balón desde el jugador A al jugador B el recorrido del balón debe ser la distancia más corta posible.

Para que puedas localizar cada jugador en el campo, se te darán coordenadas x, y para cada jugador

### Entrada.-

La primera línea de entrada contiene un entero N, el número de jugadores en el equipo de pepe ( $N \leq 11$ ), después de la primera línea se te dará N líneas cada una de 2 valores x,y (donde x, y son las coordenadas del jugador  $N_i$   $1 \leq i \leq N$  )

A continuación se te dará un entero Q ( $Q \leq 100$ ), el número de consultas. Cada  $Q_i$  línea estará compuesta 2 enteros A,B ( $A \neq B$ ) (Donde A y B son los jugadores insertados anteriormente).

### Salida.-

Para cada consulta debe imprimir la distancia que atravesó el balón desde el jugador A al jugador B (redondeado a 2 dígitos después de la coma decimal).

### Ejemplo de Entrada

```
4
0 0
0 1
0 3
1 1
3
1 2
1 3
1 4
```

### Explicación de la entrada.-

La coordenada 0 0 es el jugador 1  
La coordenada 0 1 es el jugador 2  
La coordenada 0 3 es el jugador 3  
La coordenada 1 1 es el jugador 4

Ahora para la consulta 1: nos pide la distancia más corta desde el jugador 1 (coordenada 0, 0) al jugador 2 (Coordenada 0, 1), la cual es 1.00

### Ejemplo de Salida

```
1.00
3.00
1.41
```

### **G- Palindrome Extendido**

Un string palindrome, es un string que se lee igual cuando es invertida. Por ejemplo ABA, ABBA son palindromes, pero ABB no lo es.

En este problema tú debes agregar caracteres a la derecha del string dada y convertirla en palindrome (Obviamente si ya es palindrome no es necesario hacer nada mas).

#### **Entrada.-**

Entrada terminara con el string "END" (Este no debe ser procesado), cada línea tendrá un string no vacío de letras minúsculas. La longitud de la cadena será menor a 100000

#### **Salida.-**

Para cada caso de prueba imprimir el palíndrome del string dado.

#### **Ejemplo de Entrada.-**

```
aaaa
abba
amanaplanacanal
xyz
END
```

#### **Ejemplo de Salida.-**

```
aaaa
abba
amanaplanacanalpanama
xyzyx
```