

Olimpiada Boliviana de Informática  
Fases: Distrital, Departamental y Final  
2018



Olimpiada Boliviana de Informática

Recopilación de problemas por:  
Samuel Réne Loza Ramírez  
Los problemas corresponden a sus respectivos autores.

# Índice

|                                     |   |
|-------------------------------------|---|
| Filtro                              | 1 |
| Habla                               | 2 |
| Mano Caliente                       | 5 |
| Numeros Cuadrados                   | 1 |
| Suma de Comprobación                | 2 |
| Otra vez hallar los factores primos | 3 |
| Sumas Iguales                       | 4 |
| Teorema de Dirichlet                | 5 |
| Contaminación                       | 1 |
| Cueva del Diablo                    | 2 |
| Farey                               | 3 |
| Turismo                             | 4 |

**OBI - FASE DISTRITAL**

# Filtro

Se tiene una secuencia de números provenientes de sensor. Este sensor esta defectuoso y deja pasar datos errados.

Lo que se quiere es imprimir todos los datos errados del sensor.

Los datos vienen en una lista de números. Los números errados son aquellos que multiplicados por 2 son múltiplos de tres.

Si tenemos como entrada 3, 1, 4, 1, 5, 9, 2, 6, 5 los datos errados son: 3, 9, 6 se desea imprimir la suma de los mismos luego de multiplicarlos por 2 y hallar el modulo 3. Para el ejemplo  $6 + 18 + 12 = 36$

## Entrada

Los datos de entrada consisten de varios casos de prueba. La primera línea contiene un número que indica el número de casos de prueba. Cada caso de prueba consiste de dos líneas. En la primera un número  $1 \leq n \leq 10000$  que indica el número de elementos a procesar. La segunda con los  $0 \leq n \leq 1000$  números separados por un espacio.

## Salida

Por cada caso de prueba escriba en una línea la suma como se solicito.

### Ejemplo de entrada

```
1
9
3 1 4 1 5 9 2 6 5
```

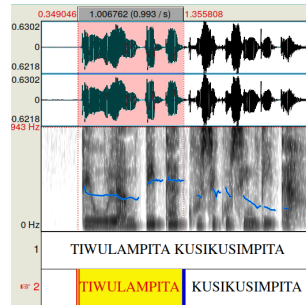
### Ejemplo de salida

La suma es 36

---

# Habla

Alicia tiene un trabajo de investigación muy bonito, en la generación de voz sintética conocido como TTS (Text to speech) por sus siglas en ingles, para hacer este trabajo Alicia necesita etiquetar los audios en un programa de análisis científico del habla. A continuación un ejemplo de como se etiquetan los audios.



El programa genera un archivo con el siguiente formato:

```
xmin = 0
xmax = 0.34
text =
```

```
xmin = 0.34
xmax = 1.35
text = "TIWULAMPITA"
```

```
xmin = 1.35
xmax = 2.57
text = "KUSIKUSIMPITA"
```

Alicia no tiene tiempo, así que necesita tu ayuda para construir un algoritmo que lea los datos y los muestre en el siguiente formato **hh:mm:ss.mm**.

## Entrada

La primera línea tendrá un número  $n \leq 100$ , que es la cantidad de casos de prueba. Seguido de  $n$  líneas. En cada línea se tiene **xmin,xmax y text**. Para mas detalle ver el ejemplo de entrada.

## Salida

Imprimir 3 líneas **xmin,xmax y text** seguido por un salto de línea menos en el ultimo caso.

Si sigue los siguientes pasos:

- 1) Todas las letras tiene que estar en minúsculas.
- 2) Solo se aceptan caracteres **a-z**, otros caracteres no tomarlos en cuenta..
- 3) Eliminar todas las entradas vacías, una entrada se considera vacía si **text** no tiene ningún carácter.

## Ejemplo de entrada

```

8
0
0.34
""

0.34
1.35
"TIWULAMPITA."

1.35
2.57
"KUSIKUSIMPITA"

2.57
2.97
""

2.97
3.73
"kuntiri:"

3.73
4.26
"alicia"

4.26
4.79
"ramos"

4.79
4.80
"      "
```

## Ejemplo de salida

```

00:00:00.34
00:00:01.35
TIWULAMPITA

00:00:01.35
00:00:02.57
kusikusimpita

00:00:02.97
00:00:03.73
kuntiri

00:00:03.73
00:00:04.26
alicia

00:00:04.26
00:00:04.79
ramos
```

**Explicación**

Para mostrar la salida del caso de prueba numero 2, se hizo lo siguiente:

**0.34** 0 segundos con 34 milisegundos, se convirtió en **00:00:00.34**

**1.35** 1 segundo con 35 milisegundos, se convirtió en **00:00:01.35**

**"tiwulampita"** siguiendo los pasos descritos en la salida se convirtió en **tiwulampita** .

---



# Mano Caliente

Mano Caliente es un termino que utilizan muchos fanáticos del baloncesto, y se refiere a que cuando un jugador mete muchos cestos seguidos. Para saber cuantas veces un jugador tuvo la mano caliente se hizo un registro de los lanzamientos al aro y se puso uno cuando logro encestar y cero cuando fallo. Un ejemplo de este registro es

```
1010011111001010110010111110
```

Cada vez que un jugador encesta mas de cuatro cestos seguidos diremos que tiene la mano caliente. En el ejemplo el jugador tuvo en dos ocasiones la mano caliente. Una temporada es un conjunto de partidos y un ejemplo es:

```
11000001011000111110100100101100110011010110001010
101001111100000100011100000111111010000111010010010101
111101000000100001100100011100001100100000110
1000000110111011101101001000001111101111011011100111111
```

En esta temporada el jugador tuvo la mano caliente en cada partido como sigue: 1 vez, 2 veces, 0 veces, y 1 vez. Por eso decimos que el 75,00 por ciento de la veces tuvo mano caliente.

## Entrada

La entrada consiste de múltiples casos de prueba. Cada caso de prueba corresponde a una temporada y cada partido viene en una línea con el detalle de los intentos de encestar que el jugador tuvo. Esta línea puede tener hasta 300 caracteres. La temporada termina cuando no hay más datos.

## Salida

Escriba en la salida, en una línea, la cantidad de veces que tuvo mano caliente en el partido. Una vez que terminan los datos de prueba escriba con dos decimales el promedio por partido que tuvo mano caliente. Como se ve en el ejemplo.

### Ejemplo de entrada

```
11000001011000111110100100101100110011010110001010
101001111100000100011100000111111010000111010010010101
111101000000100001100100011100001100100000110
1000000110111011101101001000001111101111011011100111111
```

### Ejemplo de salida

```
1
2
0
2
75,00
```

---

**OBI - FASE DEPARTAMENTAL**

# Numeros Cuadrados

Se define los números cuadrados a aquellos números enteros que pueden representar el área de un cuadrado.

Recordemos que el área de un cuadrado es la multiplicación de sus lados.

Por ejemplo 4, representa el área de un cuadrado con lado de longitud 2, 9 uno con longitud de lado 3. Los cuadrados tienen sus cuatro lados iguales.

## Entrada

La entrada consiste de múltiples casos de prueba. Cada caso de prueba consiste de dos números ( $2 \leq a, b \leq 10^8$ ). La entrada termina cuando no existen más datos.

## Salida

Por cada caso de prueba escriba en la salida cuantos números cuadrados existen entre  $a$  y  $b$  inclusive.

### Ejemplo de entrada

```
2 100
50 100
2361 6723
```

### Ejemplo de salida

```
9
3
33
```

---

# Suma de Comprobación

La suma de comprobación es una operación aritmética que se realiza sobre un texto para garantizar que el mismo no sufra modificaciones. Por ejemplo la suma de comprobación de la palabra *CURSO* sería:  $2*1+20*2+17*3+18*4+14*5 = 235$

Formalmente se escribe como:

$$n = \sum_{i=0}^{s.length()-1} (i + 1) * valor(s_i)$$

## Entrada

La entrada consiste de varios casos de prueba. Cada caso de prueba consiste de una frase que viene en una línea, que solo contiene letras mayúsculas y espacios. La entrada termina cuando no hay más casos.

## Salida

En la salida escriba la suma de comprobación.

### Ejemplo de entrada

```
CASA  
CASA DE JUAN  
CURSO  
NOTA FINAL
```

### Ejemplo de salida

```
56  
539  
235  
398
```

---

# Otra vez hallar los factores primos

Conocemos que cada número puede escribirse por el producto de sus factores primos. Por ejemplo el número 20 se puede escribir como  $2 \times 2 \times 5 = 20$ .

En este problema te piden descomponer también los exponentes de los factores por ejemplo el número el número 192 puede escribirse como  $2^6 * 3$ . Luego de descomponer el exponente 6 podemos escribir como  $(2^{(2)(3)})(3)$ . Todos los números que se debe multiplicar se escriben entre paréntesis.

Las potencias se escribe con un símbolo  $^2$  seguido un  $\{$ . Una ves terminadas las potencias se cierra con  $\}$  como se muestra en el ejemplo.

## Entrada

La entrada consiste de múltiples casos de pruebas. Cada caso de prueba consiste en una línea con un numero n Comienza con un numero n que indica el numero de casos de prueba. Luego siguen n números, cada uno, en una línea ( $n < 1000000000$ , mil millones).

## Salida

Imprima en una línea la descomposición en factores como se explico.

### Ejemplo de entrada

```
192
48
20
1073741
30
```

### Ejemplo de salida

```
(2^{(2)(3)})(3)
(2^{(2^{(2)})})(3)
(2^{(2)})(5)
(59)(18199)
(2)(3)(5)
```

---

# Sumas Iguales

Encuentre todos los números de cuatro dígitos que en base 10, 12 y 16 que tienen la propiedad que la suma de sus dígitos iguala.

Por ejemplo los dígitos del numero 2992 en base 10 suman 22 ( $2+9+9+2$ ). Este numero en base 12 es 1894 y la suma de sus dígitos ( $1+9+8+4$ ) es también 22. En base 16 es bb0 tiene la suma de ( $11+11+0$ ) y es 22.

## Entrada

No hay entrada de datos

## Salida

Escriba en la salida los números de cuatro dígitos que tienen la propiedad deseada.

### Ejemplo de entrada

No hay entrada de datos

### Ejemplo de salida

2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999

---

# Teorema de Dirichlet

El teorema de Dirichlet indica que dados dos números  $(a,b)$  que son dos números relativamente primos, la secuencia que comienza con  $(a)$  y se incrementa en  $(b)$  contiene un número infinito de números primos. Esto es:

$$a, a + b, a + 3b, a + 4b, \dots, a + nb$$

Por ejemplo si  $a = 2, b = 3$  la secuencia que se obtiene es:

$$2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, \dots$$

que contiene un número infinito de números primos.

Su tarea es que dado un número los números  $a, b, n$  imprimir el  $n$ ésimo número primo de la secuencia.

## Entrada

La entrada contiene múltiples casos de prueba. Termina cuando no hay más datos. La entrada consiste en tres números  $(2 \leq a, b, n \leq 10^7)$  separados por un espacio que vienen en una línea.

## Salida

Por cada caso de entrada imprima en la salida el  $n$ ésimo número primo de la secuencia.

## Ejemplo de entrada

```
2 3 4
3 10 3
103 230 1
27 104 185
```

## Ejemplo de salida

```
17
23
103
93523
```

---

**OBI - FASE NACIONAL**



# Contaminación

La zona alta de la ciudad de Potosí tiene un problema que es la contaminación, por ese sector existen numerosos ingenios mineros cuya labor produce contaminación en la zona, es por eso que se ha visto conveniente re ubicarlos en una zona mas alejada de la ciudad, en ese lugar se pretende ubicar a cada ingenio uno a lado de otro en linea recta y que tengan un distancia de 1 km entre cada uno de ellos, sin embargo hay un problema, la energía eléctrica. Debido a los estándares de calidad requeridos, los ingenios mineros requieren bastante electricidad y no es posible pensar en una sola linea de energía eléctrica para todos los ingenios, en muchos casos algunos necesitan su propia planta eléctrica.

En una reunión de planificación con los representantes de los ingenios mineros, la empresa que suministrar energía eléctrica (SEPSA) necesita conocer la menor cantidad de plantas eléctricas necesarias para proporcionar electricidad a todos los ingenios.

Una central eléctrica puede distribuir a varios ingenios aunque no se debe sobrecargar su capacidad y lo que se busca es construir la menor cantidad de centrales y que estas abastezcan a todos los ingenios de la zona, no todos los ingenios reúnen las condiciones para albergar una planta eléctrica

Se te proporcionara una lista de datos de los ingenios, algunos pueden albergar plantas eléctricas y otros no, entonces dado un rango de distribución de las plantas eléctricas, construye un programa para encontrar el menor numero de plantas necesarias de forma que todos los ingenios tengan energía eléctrica

## Entrada

La entrada inicia con un numero  $T$  que es el numero de casos de prueba, a continuación los casos de prueba están compuestos por dos lineas, en la siguiente linea están dos números  $N$  el numero de ingenios y  $K$  ( $1 \leq N, K \leq 100$ ) el rango de distribución de las plantas eléctricas a construir. En la siguiente linea existen  $N$  números  $C$  ( $0 \leq C_i \leq 1$ ) que indican si en ese ingenio es posible construir una planta eléctrica.

## Salida

Por cada caso de prueba, mostrar un único entero denotando el numero mínimo de plantas eléctricas que deben ser construidas de forma que todos los ingenios tengan energía eléctrica. En caso de que no sea posible, mostrar -1

### Ejemplo de entrada

```
3
3 1
0 1 0
7 2
1 1 1 0 1 1 0
8 3
1 1 1 0 1 1 1 0
```

### Ejemplo de salida

```
-1
3
2
```

---

# Cueva del Diablo

La Cueva del Diablo de Potosí se encuentra en la Quebrada de San Bartolomé, sobre la carretera que une esta capital con la ciudad de Oruro y los balnearios de Miraflores y Tarapaya. Los dos peñascos están separados por pocos metros, es un lugar muy bonito, donde, aunque no es una practica muy común, algunas personas practican escalada en roca en este hermoso lugar. Pánfilo es un estudiante que practica esta actividad que hace poco aprendió a programar, e intenta ver a la escalada como un proceso algorítmico, es decir, Pánfilo analiza la montaña desde su base como una matriz bidimensional donde cada posición de la matriz representa el costo de energía que le toma a pánfilo moverse hasta ahí. El puede realizar cada movimiento como 1 unidad en cualquiera de las cuatro direcciones: Arriba, Abajo, Izquierda y Derecha, y cada movimiento de un cuadrado diferente a otro disminuye la energía de Pánfilo en la cantidad en ese cuadrado. También existen lugares que le ayudan a ganar Energía y son fáciles de escalar a los cuales les asigna valores negativos. Esto es importante ya que si Pánfilo se queda sin energía, caerá al vacío y pasaran cosas malas. Como Pánfilo no quiere caer al vacío, necesita que le ayudes a calcular la cantidad mínima de energía que necesita para completar el ascenso, asumiendo que toma una ruta óptima.

## Entrada

La primera línea de entrada contiene un número  $T$  que es el número de casos de prueba, luego en las siguientes líneas, cada caso de prueba empieza con dos números enteros  $F, C (1 \leq F, C \leq 5)$ . La segunda línea de entrada constará de una fila de  $C$  caracteres, todos sera el carácter 'Z', separados por espacios, que representan la parte superior de la cueva del diablo . Estos consumen 0 unidades de energía al moverse ahí. Pánfilo puede elegir cualquiera de ellos. A continuación, habrá  $F$  filas de  $C$  columnas de números  $N (-9 \leq N \leq 9)$ , donde  $N$  representa la energía requerida para ingresar a esa sección de la quebrada. La última línea de entrada constará de una fila de  $C$  caracteres todos en 'A' que representan los posibles puntos de inicio de la subida. Estos requieren 0 unidades de energía para entrar.

## Salida

Por cada caso muestre en una línea, un solo número entero que es la cantidad inicial mínima de energía necesaria que necesita Pánfilo para empezar a ascender a la Cueva del Diablo sin caerse.

## Ejemplo de entrada

```
2
3 3
Z Z Z
6 8 6
-8 -8 -7
5 7 5
A A A
3 3
Z Z Z
6 5 5
-6 -7 -8
8 7 7
A A A
```

## Ejemplo de salida

```
5
7
```

---

# Farey

La secuencia de Farey es una de las series menos conocidas pero no por eso menos importante, esta se define así: "dado un entero positivo,  $N$ , la secuencia de todas las fracciones  $a/b$  con  $0 \leq a \leq b$ ,  $1 \leq b \leq N$  y siendo  $a$  y  $b$  primos relativos, enumeradas en orden creciente"

Por ejemplo, la secuencia de Farey de orden 6 es:

$$\frac{0}{1}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, \frac{1}{1}$$

Si consideras solo los denominadores de la Secuencia Farey de orden  $N$  como:  $b_1, b_2, \dots, b_K$  Entonces la suma Farey de orden  $N$  es igual a la suma de  $b_i / b_i + 1$  siendo  $i = 1 \dots K - 1$  Por ejemplo, la secuencia de Farey mostrada más arriba (de orden 6) es:  $\frac{1}{6} + \frac{6}{5} + \frac{5}{4} + \frac{4}{3} + \frac{3}{2} + \frac{2}{5} + \frac{5}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6} = \frac{35}{2}$

Escriba un programa para calcular la suma de Farell de orden  $N$  para un valor determinado de  $N$

## Entrada

La primera línea de entrada contiene un solo entero  $T$  ( $1 \leq T \leq 9999$ ), que es el número de casos de prueba. Luego le siguen  $T$  líneas donde cada línea contiene a  $N$  que es la orden  $N$  ( $2 \leq N \leq 10000$ ) de la suma de Farey que se debe calcular.

## Salida

Por cada caso de prueba, mostrar la suma de Farey como una fracción decimal. Si el denominador es 1, imprima solo el numerador.

### Ejemplo de entrada

```
5
6
16
26
9999
57
```

### Ejemplo de salida

```
35/2
239/2
635/2
91180457/2
2999/2
```

---

# Turismo

Estas en Potosí y no puedes perderte todos los hermosos lugares turísticos que hay aquí, Casa de la Moneda, Cerro Rico, Salar de Uyuni, Laguna Colorada, Árbol de Piedra, etc. Como queremos visitar la mayor cantidad de lugares turísticos, preguntamos a una agencia de turismo la mejor de la ciudad, la agencia de turismo OBI, esta agencia nos ofrece un mapa con los lugares turísticos de todo Potosí, este mapa muestra las rutas entre todos los lugares turísticos mas sus respectivas distancias. Como no tenemos mucho tiempo debemos organizar nuestras visitas y no tenemos tiempo para un circuito turístico y somos algo flojos queremos visitar algunos lugares de forma rápida desde un determinado lugar. Como aprendimos algo de programación, decidimos construir un programa que nos ayude en esta labor, entonces, sea la descripción del mapa con un numero de lugares turísticos, las rutas o caminos entre esos lugares y el punto donde nos encontramos; debes construir un programa que ayude a determinar si es posible o no llegar desde un punto en el mapa hasta un lugar turístico de la forma más rápida posible.

## Entrada

La entrada consiste de una línea con cuatro enteros no negativos,  $1 \leq N \leq 1000, 0 \leq R \leq 4000, 1 \leq C \leq 100$  y  $0 \leq P < n$ , separados por espacios. El primer valor N representa el número de lugares turísticos en el mapa, R es la cantidad de rutas o caminos entre esos lugares, luego C es la información de los lugares que queremos visitar y P es el índice del lugar en el mapa donde nos encontramos, nuestro punto de partida. Los lugares están numerados de 0 a n - 1. Luego siguen R líneas, cada línea que consta de tres números enteros (separados por espacios) U, V y W indican que hay un camino o ruta desde el lugar U al lugar V y la distancia es  $W, 0 \leq w \leq 1000$ . Luego siga C líneas de lugares a los que deseamos visitar, cada una de las líneas es un único entero no negativo, para el cual debemos calcular la distancia mínima desde el lugar donde nos encontramos, el nodo P

## Salida

Para cada consulta, mostrar una sola línea que contenga la distancia mínima desde el lugar donde nos encontramos hasta el lugar que queremos visitar en la consulta, si no hay forma de llegar mostrar la cadena "No es Posible" sin las comillas.

### Ejemplo de entrada

```
2
2 1 1 0
0 1 100
1
4 3 4 0
0 1 2
1 2 2
3 0 2
0
1
2
3
```

### Ejemplo de salida

```
100
0
2
4
No Es Posible
```