

# Maratón de programación LOOK UP - EAFIT

Agosto 30, 2014

## Comentarios generales

- Este conjunto contiene 4 problemas
- La maratón tiene una duración de 150 minutos en total.
- La tabla de resultados será congelada después de 120 minutos.
- Los envíos no serán respondidos si fueron enviados después del pasado 130 minutos de competencia. (Serán respondidos cuando la competencia acabe).
- Si están programando en Java, recuerden que el nombre del código fuente como el nombre de la clase que contiene el método *main* debe ser nombrado igual a como se describió en cada problema.
- Buena suerte para todos y diviértanse.

Página dejada intencionalmente en blanco.

## Problema A. Roxana y las pilas

Entrada: `entrada estándar`  
Salida: `salida estándar`  
Tiempo límite: 3 segundos  
Nombre archivo fuente (sólo para java): `pilas.java`

Roxana es una pequeña niña que está aprendiendo estructuras de datos. Ella te pidió que la ayudaras con un programa que simula cómo funciona una pila.

Roxana te dirá  $n$  instrucciones, cada una puede ser de sólo un tipo. Veamos qué tipo de instrucciones conoce ella:

- Cuando Roxana dice **PUSH**, este será seguido por un espacio y un número  $k$  ( $1 \leq k \leq 1000$ ); el cual debe ser insertado en la pila. En este tipo de instrucción no tienes que imprimir nada.
- Cuando Roxana dice **POP**, debes remover el último elemento de la pila, Pero no imprimirlo. Si no hay elementos en la pila debes ignorar esta instrucción.
- Cuando Roxana dice **TOP**, debes imprimir, en una línea, el último elemento de la pila sin removerlo. Si la pila está vacía, debes imprimir "EMPTY" sin comillas.

### Entrada

La primera línea de la entrada contiene un entero  $n$  ( $1 \leq n \leq 100$ ), representando el número de instrucciones que Roxana te dirá.

Después, siguen  $n$  líneas, cada una contiene cualquiera de las instrucciones explicadas anteriormente.

### Salida

La salida debe contener el resultado obtenido ejecutando las instrucciones.

### Ejemplo de entrada y salida

entrada estándar	salida estándar
6 PUSH 1 TOP POP TOP PUSH 5 TOP	1 EMPTY 5

## Problema B. Teoría de los seis grados

Entrada:	entrada estándar
Salida:	salida estándar
Tiempo límite:	3 segundos
Nombre archivo fuente (sólo para java):	grados.java

Existe una hipótesis llamada **"Seis grados de separación"** que intenta probar que cualquier persona en la tierra puede estar conectado con otra a través de una cadena de conocidos que no tiene más de cinco intermedios, es decir, conectando a ambas personas con sólo seis (6) enlaces.

Conociendo todas las relaciones que tienen las personas, hoy queremos determinar si, para dos personas, se cumple la teoría.

### Entrada

La primera línea de la entrada contiene un entero  $n$  ( $2 \leq n \leq 100$ ), representando la cantidad de personas que existen en el caso de prueba.

Después, siguen  $n$  líneas, cada una con el nombre de una persona. Se garantiza que la longitud del nombre no excede 20 caracteres, no se repiten nombres y estos están compuestos únicamente de letras mayúsculas o minúsculas del alfabeto inglés.

A continuación, sigue un entero  $m$  ( $1 \leq m \leq 500$ ) que son la cantidad de relaciones que se van a describir.  $m$  líneas siguen, cada una describiendo una relación, compuesta por dos nombres  $a$  y  $b$  separados por un espacio, esta relación significa que la persona  $a$  conoce a la  $b$  así como  $b$  conoce a  $a$ .

Finalmente, sigue un entero  $q$  ( $1 \leq q \leq 50$ ), que indica la cantidad de consultas que se quieren hacer. Después de esto, seguirán  $q$  líneas conteniendo una consulta cada una. Una consulta está compuesta por dos nombres separados por un espacio, se garantiza que los nombres son distintos y pertenecen a los dados por el caso de prueba.

### Salida

La salida debe contener  $q$  líneas, cada una diciendo **"Aceptada"** si las personas involucradas en la consulta pueden conocerse en seis (6) o menos pasos, o **"Rechazada"** de lo contrario.

## Ejemplo de entrada y salida

entrada estándar	salida estándar
10	Aceptada
Roxana	Aceptada
Esteban	Rechazada
Camila	Rechazada
Carlos	
Felipe	
Laura	
Alicia	
Santiago	
Alejandra	
Juliana	
8	
Roxana Esteban	
Esteban Camila	
Camila Carlos	
Juliana Alejandra	
Felipe Carlos	
Felipe Laura	
Alicia Santiago	
Alicia Laura	
4	
Santiago Laura	
Alicia Esteban	
Roxana Santiago	
Alejandra Felipe	

## Problema C. Google en EAFIT

Entrada: `entrada estándar`  
Salida: `salida estándar`  
Tiempo límite: `1 segundo`  
Nombre archivo fuente (sólo para java): `google.java`

El pasado miércoles 20 de agosto de 2014 en la Universidad EAFIT, se tuvo una visita por parte de Google, donde se planteó un problema de programación, el cual vamos a llevar a cabo de nuevo.

Dada cierta cantidad de líneas, que contiene cada una al menos una palabra, se deben imprimir las líneas en el mismo orden de entrada y sin repetirlas, es decir, si una línea aparece más de una vez, sólo debe ser mostrada en la primera ocurrencia.

### Entrada

La entrada está compuesta por varias líneas (máximo 100), una línea no puede tener un tamaño mayor a 100 caracteres. Los caracteres permitidos son únicamente mayúsculas y minúsculas del alfabeto inglés.

*Se debe leer hasta fin de archivo (EOF).*

### Salida

La salida contiene la cantidad de líneas que deben ser impresas resolviendo el problema, por consiguiente, no deben haber líneas repetidas en la salida.

### Ejemplo de entrada y salida

entrada estándar	salida estándar
Se ve muy feo escribir sin tildes	Se ve muy feo escribir sin tildes
Hola esto es una linea	Hola esto es una linea
Hola esto es otra linea	Hola esto es otra linea
Hola esto es una linea	Un problema facil
Un problema facil	
Se ve muy feo escribir sin tildes	

## Problema D. El paseo de Roxana

Entrada: `entrada estándar`  
Salida: `salida estándar`  
Tiempo límite: 3 segundos  
Nombre archivo fuente (sólo para java): `paseo.java`

Roxana está planeando un paseo a su ciudad favorita, ella conoce tanto la ciudad, tanto así que conoce las rutas que llegan a ella. Lastimosamente, ella posee un presupuesto muy apretado, por esto, te pidió el favor de calcular cuál sería el mínimo costo para realizar dicho viaje.

### Entrada

La primera línea de la entrada contiene dos enteros  $n$  ( $1 \leq n \leq 100$ ) y  $m$  ( $1 \leq m \leq 500$ ), representando el número de ciudades y la cantidad de rutas que conoce respectivamente. Las ciudades están numeradas de 1 hasta  $n$ .

Después, siguen  $m$  líneas, cada una contiene tres enteros  $a$ ,  $b$  y  $w$  ( $1 \leq a, b \leq n$ ), ( $w \geq 0$ ) que describen respectivamente, la ciudad origen, la ciudad destino y el costo de la ruta. Lastimosamente, las rutas no son en ambos sentidos.

### Salida

Deberás imprimir el mínimo costo de ir de la ciudad 1 hasta la ciudad  $n$ , si no es posible hacer este viaje el programa debe imprimir -1.

### Ejemplo de entrada y salida

entrada estándar	salida estándar
4 5 1 2 3 1 3 2 2 3 4 2 4 2 3 4 1	3

### Ejemplo de entrada y salida

entrada estándar	salida estándar
4 5 1 2 4 1 3 1 2 3 5 4 2 6 4 3 7	-1