



**VII Olimpiada de Informática
del estado de Guanajuato
Segundo examen selectivo**



El comité organizador te da la bienvenida al segundo examen selectivo de la VII Olimpiada de Informática del Estado de Guanajuato.

- 1) El examen tiene una duración de 4:30 horas.
- 2) El examen consiste en 4 problemas de programación en el ambiente “C”.
- 3) Tu carpeta de trabajo esta en “C:\OIEG\AlumnoX”. Donde X es el número de la maquina que se asigno. Deberás nombrar cada programa con el nombre que se te indique respectivamente.
- 4) Debes hacer un programa para cada problema, cada problema será evaluado con alguna cantidad de casos de prueba. El puntaje que recibirás en cada problema, dependerá del número de casos que tu programa haya resuelto satisfactoriamente.
- 5). Todos los problemas tienen el mismo puntaje, por lo que te recomendamos que intentes primero los problemas que consideres más fáciles.
- 6) No esta permitido el uso de libros, calculadoras, tablas o cualquier otro documento que el comité no te haya proporcionado.
- 7) Deberás crear un archivo de texto en tu carpeta de trabajo con el nombre de “Datos.txt”. Donde guardaras: nombre completo, escuela, teléfono, correo electrónico y grado escolar.

¡El comité de la OIEG te desea MUCHA SUERTE!

Removiendo Montículos

Archivo fuente: Monticulo.cpp

Archivo de entrada: input.txt

[Graham Poulter/USACO, 2005]

Archivo de Salida: output.txt

Historia

Siempre pensando en la experiencia de pasteo de las vacas, el Granjero Juan (GJ) ha encontrado que el debe remover N montículos invisibles del pastizal pues piensa que hay terroristas debajo de ellos. Los montículos están convenientemente arreglados en una línea recta y numerados $1...N$ con cada montículo teniendo alguna altura H_i ($1 \leq H_i \leq 10,000$).

GJ usaría los explosivos tradicionales para destruir los montículos pero a él le gusta la emoción y prefiere usar explosivos que están formulados para destruir montículos adyacentes en tanto estos montículos adyacentes sean estrictamente más cortos que el montículo más cercano que está siendo destruido. La explosión puede continuar pasando al montículo más cercano adyacente al montículo adyacente siguiente si es aún más corto que el montículo más cercano recién destruido. Tan pronto como un montículo encontrado por la explosión no es más corto, dejan de destruirse los montículos hacia ese lado, el otro lado sigue las mismas reglas con cualesquiera montículos que aparezcan ahí.

Por ejemplo considera una línea de nueve montículos con estas alturas:

1 2 5 4 3 3 6 6 2

Si GJ vuela el tercer montículo (con altura 5), entonces el segundo montículo también será destruido (altura 2) y el primer montículo (altura 1) también será destruido. De manera similar, el cuarto montículo (altura 4) y el quinto montículo (altura 3) serán destruidos dado que ellos son sucesivamente más cortos, dejando la línea como esto:

* * * * 3 6 6 2

Dos explosiones más (en los montículos 7 y 8) destruirán el resto.

Problema:

Dados N , y N alturas ayude a GJ a determinar que montículos debe explotar para que use el número mínimo de cargas de explosivos para destruir los montículos.

Consideraciones:

- N siempre es $1 \leq N \leq 50,000$.

Sábado 28 de enero de 2006

- El IQ de GJ es de 10 (supo poner bien su nombre).
- Las alturas H_i ($1 \leq H_i \leq 10,000$).
- GJ ha presentado severos casos de esquizofrenia.

Entrada:

- * Línea 1: Un solo entero, N
- * Líneas 2...N+1: La línea i+1 contiene H_i

Salida:

- * Líneas 1..?: Cada línea contiene un entero el cual es el índice de un montículo a ser volado. Los índices deben ser mostrados en orden ascendente.

Entrada	Salida
9	3
1	7
2	8
5	
4	
3	
3	
6	
6	
2	

Ecuación

Archivo fuente: Ecuacion.cpp

Archivo de entrada: input.txt

Archivo de Salida: output.txt

Historia:

A tu amigo Pepito le dejaron de tarea que encontrara una solución para la siguiente ecuación $2x + 3y = 1$, como eres un muy buen amigo, has decidido ayudarlo a hacer su tarea. Más aún, has decidido hacer un programa que encuentre una solución para todas las ecuaciones de esa forma usando lo que acabas de aprender con Pepe.

Problema:

Dados los enteros a , b , c , encontrar los enteros x , y , tales que $a*x + b*y = c$.

Consideraciones:

- No importa si no conoces a ningún Pepito.
- $0 \leq a, b, c \leq 1,000,000$.

Entrada:

El archivo de entrada contiene 1 sola línea con tres enteros a , b , c separados por un espacio.

Salida:

El archivo de salida deberá contener una línea con dos enteros, (x, y) , tales que $a*x + b*y = c$. En caso de que no hubiera tales x , y , deberás imprimir

“No existen x , y tales que $a*x + b*y = c$.”

sustituyendo las letras a , b , c , por los valores de la entrada.

Entrada	Salida
3 7 1	-2 1
2 4 1	No existen x , y tales que $2*x + 4*y = 1$

Unos

Archivo fuente: Unos.cpp

Archivo de entrada: input.txt

Archivo de Salida: output.txt

Historia:

Como paga por tu ayuda, tu amigo Pepito te ha pasado un tip acerca de una página de Internet en la cual hacen una especie de rifa.

Pepito dice que se ha dado cuenta de que el número ganador, cuando lo conviertes a base 2, siempre tiene la misma cantidad de unos que el día en que se hizo el concurso.

Para tener más probabilidades lo que tienes que hacer es encontrar todos los números que tengan k unos en su expansión binaria, donde k es el día del concurso y apostarle a alguno de ellos.

Problema:

Dados a , b y c , encontrar todos los números k , $a \leq k \leq b$, tales que k en base 2 tiene exactamente c unos.

Entrada:

El archivo de entrada se compone únicamente de tres enteros, $0 \leq a, b, c \leq 1,000,000$, separados por un espacio.

Salida:

El archivo de salida deberá contener dos líneas, la primera con la cantidad n de números k encontrados tales que $a \leq k \leq b$ y $(k)_2$ tiene c unos, la segunda deberá contener a todos los enteros k separados por un espacio.

Entrada	Salida
1 10 2	5
	3 5 6 9 10

EL PASATIEMPO DEL REY

Archivo fuente: pasatiempo.cpp

Archivo de entrada: input.txt

Archivo de Salida: output.txt

HISTORIA

El rey “fulano de tal” tiene un extraño pasatiempo (claro es estos problemas todo es extraño) que consiste en que en su reino existen murales poco usuales, estos murales están en forma de línea recta. Los murales que el rey compra en un inicio son rectangulares formados por cuadros de colores, cada color es identificado con un numero.

Para llevar el mural rectangular a una línea recta quita cada cuadro del mural y lo colocando uno después de otro. Pero la forma es que los retira los cuadros del mural es en forma de espiral, en sentido de las manecillas del reloj, esto es quita la primera hilera de cuadro, luego la ultima columna, después la hilera inferior, para terminar con la primera columna y luego prosigue con el resto del mural de la misma forma hasta que termina.

Como ha hecho demasiadas veces esto, ahora quiere determinar como se vera el mural después de retirar los cuadros, para ver si lo compra o no.

El rey le ha pedido a la comunidad de programadores de su reino para que le muestren como se vera cierto mural es su reino. De no aceptar esta tarea, el rey se vera en la penosa necesidad de mandar contarte la cabeza, suerte.

Problema

Dado el mural de $n \times m$, determina para el rey como se vera el nuevo mural.

Entrada

En la primera línea habrá dos enteros positivos n , m , donde, $1 \leq n < m \leq 500$. En las siguientes n líneas habrá m enteros positivos, que indican el color de cada cuadro del mural, el color de un cuadro esta en el intervalo $[0, 255]$.

Salida

Una única línea que contiene la distribución de los cuadros en el nuevo mural



**VII Olimpiada de Informática
del estado de Guanajuato**
Segundo examen selectivo



Entrada	Salida
4 4	0 1 2 3 0 6 1 1 1 1 5 3 2 1 5 10
0 1 2 3	
3 2 1 0	
5 10 5 6	
1 1 1 1	