



V Olimpiada de Informática del estado de Guanajuato Tercer Examen Selectivo



El comité organizador te da la bienvenida al Tercer Examen Selectivo Práctico de la V Olimpiada de Informática del Estado de Guanajuato.

- 1) El examen tiene una duración de 4:30 horas.
- 2) El examen consiste en 3 problemas de programación en el ambiente “Turbo C++”.
- 3) Tu carpeta de trabajo esta en “C:\OIEG\X”. X es tu nombre. Deberás nombrar cada programa con el nombre que se te indique respectivamente. Cada programa debe estar en una carpeta que lleve el mismo nombre del problema.
- 4) Debes hacer un programa para cada problema, el cual se evaluará en 10 casos de prueba. El puntaje que recibirás en cada problema, dependerá del número de casos de prueba que tu programa haya resuelto satisfactoriamente.
- 5). Todos los problemas valen el mismo puntaje. Cada caso de prueba tiene un valor de 1 punto.
- 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.*

¡El comité de la OIEG te desea MUCHA SUERTE!

Domingo 9 de Mayo del 2004



Celdas de Neutrinos

Archivo: **Celdas.cpp**

Historia

Durante muchos años se han utilizado celdas solares para dar energía a nuestros satélites y sondas. Recientemente, a la Organización Internacional de Estudios de Geociencias (OIEG) se le ha ocurrido una brillante idea para poder enviar a Marte (al planeta, sin alusiones personales) una sonda espacial llamada “Karel”. La idea consiste en elaborar una vela de celdas de neutrinos en vez de solares, dado que la cantidad de neutrinos que se reciben en Marte (el planeta) es la misma que la que se recibe en la Tierra. Además, los neutrinos atraviesan la materia como si no existiera, por lo que las celdas se pueden empalmar. De esta forma, la energía que recibirá la sonda espacial “Karel” sólo dependerá de la cantidad de celdas rectangulares con las que cuente la vela. Sin embargo, hubo un problema al tratar de desplegar la vela al llegar a Marte (sí, ya lo saben), y por ello sólo la mitad de la vela (que era cuadrada) a podido ser desplegada. Ahora, los científicos se enfrentan al problema de si la sonda recibirá suficiente energía con sólo la mitad de la vela desplegada. Aquí es donde entras tú.

Problema

Tú misión: realiza el programa que, dado el tamaño de la vela, calcule la cantidad de rectángulos que caben en la mitad de la vela.

Entrada: *Input.txt*

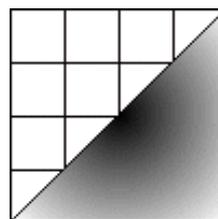
La entrada consiste únicamente de un entero $1 \leq N \leq 100$ que representa el tamaño de la vela cuadrada.

Salida: *Output.txt*

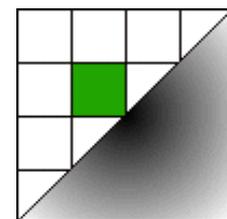
El archivo de salida debe contener un solo entero que indica el número de rectángulos que se pueden superponer. (ver figuras).

Ejemplo:

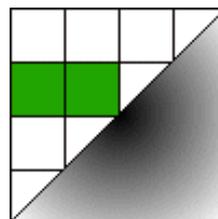
Entrada	Salida
4	15



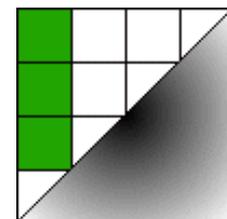
Vela



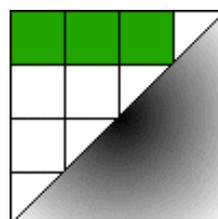
Cuadro 1x1



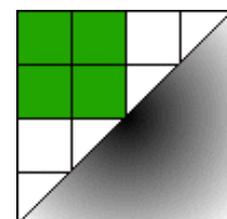
Cuadro 2x1



Cuadro 1x3



Cuadro 3x1



Cuadro 2x2



V Olimpiada de Informática del estado de Guanajuato Tercer Examen Selectivo



Karel Mapandum

Archivo: **Mapa.cpp**

“dibuje” en un archivo de texto el mapa detallado que describen los datos.

Historia

Una vez que fue posible saber a ciencia cierta que la sonda espacial si tiene suficiente energía, se ha procedido al reconocimiento del lugar de aterrizaje. La transmisión de datos desde Marte resulta muy tardada debido a la enorme distancia que nos separa del astro rojo, por lo cual la OIEG ha decidido contratar tus servicios como destacado programador para desarrollar un programa que codifique las señales enviadas desde la sonda “Karel” y que presente la información de forma fácil de analizar.

Problema

El código desarrollado por la OIEG consta de una serie de coordenadas seguidas por una clave que muestra el tipo de terreno que se encuentra en esas coordenadas. Las coordenadas son números enteros $0 \leq X, Y \leq 10,000$ y las claves son las siguientes:

0	1	2	3	4
agua	arena	roca	desconocido	inexplorado

Tu misión es escribir un programa que dados los datos recibidos de la sonda,

Entrada: *Input.txt*

En la primera línea encontrarás el número N de coordenadas-clave enviadas por la sonda. En las siguientes N líneas se encuentran ternas de enteros $Y_i X_i C_i$ separados por un espacio, que indican las coordenadas Y, X y la clave del tipo de terreno encontrado.

Salida: *Output.txt*

En el archivo de salida debes dibujar el terreno (ver ejemplo) de tal forma que sólo presente el mínimo rectángulo que representa toda la información enviada por la sonda. En la esquina superior izquierda debe ir la coordenada de menor significancia mientras que en la esquina inferior derecha el de mayor significancia. El terreno se representa con la siguiente clave :

A	N	R	D	*
agua	arena	roca	desconocido	inexplorado

Si existe una parte del mapa que debe dibujarse pero de la cuál no se ha recibido información, considera a ese terreno como “inexplorado”.

Ejemplo:

Entrada	Salida	Entrada	Salida
6 0 0 0 0 2 1 1 1 0 1 2 1 2 0 1 2 1 2	A*N *AN NR*	1 1 2 0	A



V Olimpiada de Informática del estado de Guanajuato Tercer Examen Selectivo



Ductos de agua

Archivo: **Ductos.cpp**

Historia

Gracias a los resultados presentados por el programa de decodificación, fue posible localizar yacimientos de agua en una región muy extensa de Marte. Estos se encuentran bajo la superficie del planeta. La OIEG considera que sería de gran utilidad que “Karel” construya un ducto subterráneo que comunique los abastecimientos de agua para poderla aprovechar.

La sonda comienza a cavar pero experimenta algunas fallas mecánicas que la dejan imposibilitada para avanzar hacia el norte y hacia el oeste y tampoco puede girar.

La OIEG insiste en que la construcción del ducto es de vital importancia para el desarrollo de misiones futuras, así que decide llamarte de nuevo (basado en tus excelentes resultados obtenidos hasta ahora) para que escribas un programa que encuentre la forma más eficiente de cavar el ducto.

Problema

Escribe un programa que dada la distribución y la cantidad de agua en

Ejemplo:

Entrada	Salida
6 5 1 7 0 0 0 0 5 1 0 0 0 2 0 0 0 0 0 0 0 0 0 0 20 0 0 50 0 0 0 1	52

cada yacimiento de un terreno, encuentre la mayor cantidad de agua que se puede conectar con un ducto cavando únicamente hacia el sur y hacia el este.

La sonda afortunadamente se encuentra en la posición más al noroeste del terreno.

Entrada: *Input.txt*

En la primera línea del archivo, se encuentran dos enteros $0 < M, N \leq 100$ que indican las dimensiones del terreno. M es el número de filas y N el número de columnas.

En cada una de las siguientes líneas se encuentra la distribución y capacidad del yacimiento de agua en cada posición del terreno. Un 0 en la fila i, columna j, indica que no hay yacimiento de agua en (i,j).

Salida: *Output.txt*

La salida es simplemente un entero que nos indique la máxima cantidad de agua que se puede conectar mediante un ducto cavado con las condiciones del problema.