



## V Olimpiada de Informática del estado de Guanajuato Séptimo Examen Selectivo



El comité organizador te da la bienvenida al séptimo 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 “DJGPPTurbo 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) Podrás consultar dudas tanto de redacción como de lógica sobre los problemas. Del mismo modo, contarás con 10 casos de prueba (para cada problema) similares a los casos de prueba con los que se evaluarán tus programas.
- 7) Deberás crear un archivo de texto en tu carpeta de trabajo con el nombre de “Datos.txt”. Donde guardarás: *nombre completo, escuela, teléfono, correo electrónico*.

¡El comité de la OIEG te desea MUCHA SUERTE!

Domingo 24 de Abril del 2004



# V Olimpiada de Informática del estado de Guanajuato Séptimo Examen Selectivo



## Pasos

Archivo: **pasos.cpp**

Archivo de entrada: input.txt

Archivo de salida: output.txt

## Historia:

Te encuentras de camino a Durango para participar el concurso nacional de la OMI y Mars se da cuenta que la camioneta que los lleva se ha descompuesto. Después de detenernos e intentar componer la camioneta logramos hacerla que se mueva, lamentablemente tanto el motor como los frenos han visto disminuidas sus capacidades. Ahora el motor solo puede aumentar la velocidad de la camioneta en un kilómetro cada hora que pasa y los frenos solo pueden disminuirla en un kilómetro en el mismo tiempo. Al Dueñez se le ocurre revisar el mapa y se da cuenta que estamos en el kilómetro X de la carretera y que el hotel en que nos quedaremos en Durango se encuentra en el kilómetro Y. ¿Cuántas horas tardaremos en llegar al Hotel?

## Problema

Dados el número de kilómetro de la carretera en el que se encuentra la camioneta y el kilómetro en el que se encuentra el hotel en Durango, debes encontrar cuantas horas nos tardaremos en llegar a este último.

Debes considerar que la camioneta empieza detenida y que nos debemos parar exactamente en el hotel. A la camioneta no le funciona la reversa.

## Entrada

Tu programa deberá leer del archivo de texto **input.txt** dos enteros  $0 \leq x \leq y \leq 2^{31}$  que corresponden a las posiciones de la camioneta y a la de la gasolinera.

## Salida

Tu programa deberá escribir en el archivo de texto **output.txt** un solo entero que diga la cantidad mínima de horas que deben pasar para llegar de x a y.

Entrada	Salida
45 50	4



## V Olimpiada de Informática del estado de Guanajuato Séptimo Examen Selectivo



### El Juego de la Vida

*Archivo:* juego.cpp

**Archivo de entrada:** input.txt

**Archivo de salida:** output.txt

### Historia

El juego de la vida es un autómatas celular diseñado por el matemático británico John Horton Conway en 1970. Es el mejor ejemplo de un autómatas celular.

Hizo su primera aparición pública en el número de octubre de 1970 de la revista Scientific American, en la columna de juegos matemáticos de Martin Gardner. Desde un punto de vista teórico, es interesante porque es equivalente a una máquina universal de Turing, es decir, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

Desde su publicación, ha atraído mucho interés debido a la gran variabilidad de la evolución de los patrones. La vida es un ejemplo de emergencia y autoorganización. Es interesante para los científicos, matemáticos, economistas y otros observar cómo patrones complejos pueden provenir de la implementación de reglas muy sencillas.

La vida tiene una variedad de patrones reconocidos que provienen de determinadas posiciones iniciales. Poco después de la publicación, se descubrieron el pentominó R y el planeador (en inglés glider), lo que atrajo un mayor interés hacia el juego. Contribuyó a su popularidad el hecho de que se publicó justo cuando se estaba lanzando al mercado una nueva generación de miniordenadores baratos, lo que significaba que se podía jugar durante horas en máquinas que, por otro lado, no se utilizarían por la noche. Para muchos aficionados, el juego de la vida sólo era un desafío de programación y una manera divertida de usar ciclos de la CPU. Para otros, sin embargo, el juego adquirió más connotaciones filosóficas. Desarrolló un seguimiento casi fanático a lo largo de los años 1970 hasta mediados de los 80.

El juego de la vida es en realidad un juego de cero jugadores, lo que quiere decir que su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posterior. El "tablero de juego" es una malla formada por cuadrados ("células") con la topología de un toro\*. Cada célula tiene 8 células vecinas, que son las que están próximas a ella, incluso en las diagonales. Las células tienen dos estados: están "vivas" o "muertas" (o "encendidas" y "apagadas"). El estado de la malla evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por turnos). El estado de todas las células se tiene en cuenta para calcular el estado de las mismas al turno siguiente. Todas las células se actualizan simultáneamente.

Las transiciones dependen del número de células vecinas vivas:

Una célula muerta con exactamente 3 células vecinas vivas "nace" (al turno siguiente estará viva).



## V Olimpiada de Informática del estado de Guanajuato Séptimo Examen Selectivo



Una célula viva con 2 o 3 células vecinas vivas sigue viva, en otro caso muere o permanece muerta (por "soledad" o "superpoblación")

\* Con topología de un toro nos referimos a que consideramos que las casillas que están en la frontera superior, son vecinas de las que están en la frontera inferior de la malla y lo mismo con las fronteras de los lados.

### Problema

Debes escribir un programa que dada una configuración inicial de una malla de  $M \times N$ , obtenga la configuración que tendrá esta después de aplicarle las reglas del juego de la vida durante  $K$  turnos.

### Entrada

Tu programa deberá leer del archivo **input.txt** los siguientes datos, en la primera línea los números  $1 \leq M, N \leq 100000$  que son las dimensiones de la malla. El número  $0 \leq K \leq 10000$  que es la cantidad de turnos (iteraciones) que deberás procesar

A continuación, se encontrarán  $M$  líneas de  $N$  0's y 1's. Un 0 significa que esa célula esta muerta, un 1 significa que está viva.

### Salida

Tu programa deberá escribir en el archivo de texto **output.txt** la configuración de la malla después de aplicarle las reglas del Juego de la Vida durante las  $K$  generaciones. Un 1 denotará que la célula está viva y un 0 que está muerta.

Entrada	Salida
8 9	
101	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0
0 0 0 0 0 1 1 1 0	0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0	0 1 1 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	



# V Olimpiada de Informática del estado de Guanajuato Séptimo Examen Selectivo



## Ríos

Archivo: rios.cpp

Archivo de entrada: input.txt

Archivo de salida: output.txt

### Historia:

En cimatolandia estudian la distribución del agua pluvial. Para ello, requieren saber como escurre el agua a través del terreno. Un mapa de flujo es una matriz en la cual cada casilla tiene una dirección que indica a que dirección escurre el agua (norte, sur, este y oeste). En ocasiones, hay casillas que no escurren a otra casilla, esas casillas se llaman pozos. Los ríos son los caminos que sigue el agua cuando escurre de una casilla a otra hasta llegar a un pozo (siempre se llega a un pozo). Dos casillas pertenecen a la misma cuenca si los ríos que contienen a las casillas, terminan en el mismo pozo.

### Problema

Dado un mapa de flujo y una casilla  $x$ , determinar cuales casillas pertenecen a la misma cuenca a la que pertenece  $x$ .

### Entrada

En la primera línea hay 4 enteros  $1 \leq ren \leq 10000$ ,  $1 \leq col \leq 10000$ ,  $0 \leq posRen < ren$  y  $0 \leq posCol \leq col$  en ese orden. Donde  $ren$  y  $col$  representan el número de renglones y columnas del mapa de flujo y  $posRen$  y  $posCol$  indican el renglón y columna de la casilla  $x$ . En las siguientes  $ren$  líneas vienen  $col$  enteros, separados por un espacio, representando el mapa de flujo. El valor -1 representa que no hay flujo en la casilla. El valor cero indica que hay flujo al norte, el uno que hay flujo al este, el 2 al sur y el 3 al oeste.

**Nota:** La casilla de la esquina superior izquierda tiene coordenadas (0,0) y la esquina inferior derecha se identifica por el renglón  $(ren - 1)$  y columna  $(col - 1)$ .

### Salida

Una matriz de  $ren \times col$  ceros y unos, separados por espacio, donde un uno indica que la casilla pertenece a la misma cuenca y un cero a una diferente a la que pertenece  $x$ .

Entrada	Salida
5 5 1 2	1 1 1 0 0
-1 3 3 1 -1	1 1 1 1 0
1 0 3 3 0	0 1 1 0 0
2 1 0 2 2	0 0 0 0 0
1 1 1 1 -1	0 0 0 0 0
1 1 1 1 0	