# VI Olimpiada de Informática del estado de Guanajuato
# Tercer Examen Selectivo

El comité organizador te da la bienvenida al Primer Examen Selectivo de la VI Olimpiada de Informática del Estado de Guanajuato.

1) El examen tiene una duración de 5 horas.

2) El examen consiste en 5 problemas de programación en el ambiente “C++”.

3) Tu carpeta de trabajo esta en “C:\OIEG\AlumnoX”. Donde X es el número que se te asigno. Deberás nombrar cada programa con el nombre que se te indique respectivamente.

4) Los casos de prueba están en: “C:\OIEG\Ejemplos” y el programa “DJGPP” esta en: “C:\OIEG\DJGPP\BIN\rhide.exe”

5) 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.

6).Todos los problemas tienen el mismo puntaje, por lo que te recomendamos que intentes primero los problemas que consideres más fáciles.

7) No esta permitido el uso de libros, calculadoras, tablas o cualquier otro documento que el comité no te haya proporcionado.

8) 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!

# Swap Matriz

**Archivo fuente**: Swap.cpp

**Archivo de entrada**: input.txt **Archivo de salida**: output.txt

## Historia

A los entrenadores les han dado las puntuaciones de cada alumno de *n* problemas. Cada alumno se enumeró de alguna manera, contando desde cero hasta *m*-1, donde *m* es el número de alumnos. Los resultados están en *m* columnas con *n* reglones, una columna por alumno y un renglón por problema.

Alguien entró furtivamente (a la computadora de Mars) y cambio los resultados pero dejó una nota que dice así: "He cambiado los resultados por columnas, pero también he dejado una lista de números como pista para recuperar los resultados. El primer número de la lista, es el número de la columna que debería estar en la primera posición, el segundo número de la lista es el número de la columna que debería estar en la segunda posición, etc…".

## Problema

Tu trabajo consiste en rescribir la matriz de manera correcta dada la matriz alterada y la lista que reconstruye el orden.

## Entrada

El archivo de entrada contiene dos números $0 \leq n \leq 10000$ y $0 \leq m \leq 10000$. En las siguientes *n* líneas se tienen *m* números que representan las calificaciones en desorden. Seguidamente viene una última línea que contiene *m* números que representan la clave.

## Salida

El archivo de salida consiste en la matriz de *m*x*n* (renglón, columna) ordenada de calificaciones.

| input.txt | output.txt |
|---|---|
| 2 3<br>10 0 5<br>7 10 5<br>1 2 0 | 0 5 10<br>10 5 7 |

# Región

**Archivo fuente**: Region.cpp

**Archivo de entrada**: input.txt **Archivo de salida**: output.txt

## Historia

Limo se propuse hacer un juego de "buscaminas". El buscaminas se juega en una cuadrícula de *n*x*m* de números. Los números van de cero a 8. Se dice que dos cuadros son vecinos si comparten un lado en común. Del mismo modo, se dice que dos cuadros están en la misma región **si tienen el mismo número y existe un camino de vecinos y todos ellos tienen el mismo número.** Limo se ha fijado en unos de los cuadros de coordenada (*w*,*z*), donde *w* es el renglón y *z* la columna, y desea saber cuales puntos se encuentran en la misma región.

**Nota: El primer renglón se considera como la coordenada cero en Y y la primera columna como la coordenada cero en X.**

## Problema

Tu trabajo consiste en marcar los cuadros que se encuentran en la misma región dado un cuadro inicial.

## Entrada

El archivo de entrada consiste en dos líneas. La primera línea contiene dos $0 \leq n \leq 10000$ y $0 \leq m \leq 10000$. Donde *n* representa el número de renglones y *m* el número de columnas. En las siguientes *n* líneas se tienen *m* números que representan la cuadrícula. Seguidamente viene una última línea que contiene dos números *w* y *z* que representan el número de columna y renglón del cuadro inicial.

## Salida

Consiste en una matriz de *n*x*m* de ceros y unos. Un cero significa que cuadro no está en la región y un 1 que si se encuentra en la región.

| input.txt | output.txt |
|---|---|
| 3 4<br>1 1 3 3<br>2 2 3 1<br>0 1 3 3<br>1 4 | 0 0 1 1<br>0 0 1 0<br>0 0 1 1 |

# Pesca

**Archivo fuente**: Pesca.cpp

**Archivo de entrada**: input.txt **Archivo de salida**: output.txt

## Historia

Limo se ha ido de pesca y ha observado que los peces se encuentran distribuidos a los largo de un lago. El tiene una red de tamaño *nxm.* Un pez es atrapado si se encuentra en el área en donde Limo tire la red. Ayuda a Limo a decidir donde tirar su red. La red siempre cubre una superficie de *n* unidades en la vertical y *m* en lo horizontal (Limo no sabe tirarla de otra manera).

## Problema

Dado un arreglo de $0 \leq k \leq 1000$ puntos de coordenadas enteras $0 \leq x_i , y_i \leq 10000$. Hallar el mayor número de peces que puede atrapar Limo con su red de *nxm* con $1 \leq n \leq 10000$ y $1 \leq m \leq 10000$.

## Entrada

El archivo de entrada consiste en *k*+1 líneas. La primera línea contiene tres enteros *n*, *m y k*. En las siguientes *k* líneas vienen las coordenadas de los peces en formato (X,Y).

## Salida

El archivo de salida consiste en una única línea con un solo entero que indica el número máximo de peces que puede atrapar Limo con su red.

| input.txt | output.txt |
|---|---|
| 4 3 5<br>0 0<br>0 1<br>2 1<br>4 4<br>10 0 | 3 |

# Ácidos-Básicos

**Archivo fuente**: Acidos.cpp

**Archivo de entrada**: input.txt **Archivo de salida**: output.txt

## Historia

Trabajas para una compañía que desarrolla ácidos y los químicos necesarios para neutralizarlos. Acaban de desarrollar un ácido muy poderoso y como no quieren que se salga de control te piden que encuentres la forma de neutralizarlo. Luego de mucha investigación llegas a la conclusión de que sólo la mezcla de 3 compuestos lo neutraliza, pero hay unas ciertas restricciones, el **volumen de la solución que neutraliza** al ácido es constante (**n**), la cantidad mínima de cada uno de los **3 componentes que se deben mezclar** es también fija ($\mathbf{k_1}$, $\mathbf{k_2}$, $\mathbf{k_3}$). Desafortunadamente, sólo cuentas con una cantidad (**m**) de cada uno de los tres reactivos que neutralizan.

La empresa está interesada en saber qué ten fácil es neutralizar el nuevo ácido.

## Problema

Escribe un programa que dados los números **n, m,** $\mathbf{k_1}$, $\mathbf{k_2}$ y $\mathbf{k_3}$, calcule en número de soluciones distintas que neutralizan al ácido.

## Entrada

Tu programa recibirá del archivo *input.txt* los siguientes datos. En la primera línea, los enteros **n** y **m** separados por un espacio. $0 < \mathbf{n} < 100000$, $\mathbf{n/2} < \mathbf{m} < \mathbf{n}$. En la siguiente línea los enteros $\mathbf{k_1}$, $\mathbf{k_2}$ y $\mathbf{k_3}$ separados por un espacio, $0 < \mathbf{k_i} < \mathbf{m}$. Además, siempre se tiene que $n \leq 3*\mathbf{m}$

## Salida

Debes escribir una única línea en el archivo de salida *output.txt* que contenga un entero representando la cantidad de soluciones que neutralizan al ácido.

| Entrada | Salida |
|---|---|
| 5 3<br>1 2 1 | 3 |

# Matrushkas

**Archivo fuente**: Matrush.cpp

**Archivo de entrada**: input.txt **Archivo de salida**: output.txt

## Historia

Has decidido comprar una Matrushka enorme en la feria mundial de Matrushkas (las Matrushkas son esos simpáticos adorno con forma de persona que dentro tienen otra persona y dentro otra, y otra…). Al llegar a la feria de das cuenta que es como en la feria de León, es decir hay un único recorrido por todos los estantes y que no te puedes regresar. Por suerte en la entrada de la feria te dan el tamaño de las Matrushkas que se venden en cada estante.

Lo que te interesa es comprar la mayor cantidad de Matrushkas que sea posible con la restricción de que siempre que compres una Matrushka, todas las que llevas quepan adentro de la nueva (por aquello de sólo cargar una.

## Problema

Escribe un programa que dados los tamaños que venden en cada puesto de la feria, calcule la mayor cantidad de Matrushkas que puedes comprar. Para que una Matrushka quepa dentro de otra, su tamaño debe ser menor (no igual).

## Entrada

Tu programa recibirá del archivo *input.txt* los siguientes datos. En la primera línea, un entero **n**, que representa el número de puestos en la feria, $0 < \mathbf{n} < 30$. En la siguiente línea, **n** enteros $\mathbf{k_1}, \mathbf{k_2}, \ldots, \mathbf{k_n}$ separados por un espacio, $0 < \mathbf{k_i} < 10000$, que representan el tamaño de la Matrushka que se vende en el **i**-ésimo puesto. (entras por el primer puesto de la lista).

## Salida

Debes escribir una única línea en el archivo de salida *output.txt* que contenga un entero representando la máxima cantidad de Matrushkas que puedes comprar.

| input.txt | output.txt |
|---|---|
| 5<br>3 1 6 5 8 | 3 |