



**VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)**



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

- 1) El examen tiene una duración de 4 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 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 “Rhide” esta en:  
“C:\OIEG\DJGPP\BIN\”
- 5) Debes hacer un programa para cada problema, estos serán evaluados 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!



**VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)**



## Merge

**Problema:** merge.cpp

**Entrada:** input.txt

**Salida:** output.txt

### Historia

Pepe es un alumno muy dedicado que siempre termina rápido sus ejercicios en la clase de teoría de números, por lo que mientras el resto de sus compañeros termina, él se pone a jugar con los números que están escritos en el pizarrón. Para ello, toma primero **n** números al azar y los anota en su libreta en el orden que los va eligiendo, y luego toma otros **m** números al azar sin importar cuales y también los anota en su libreta en el orden que los va eligiendo. Lo que quiere hacer ahora es ponerlos en una sola lista de tal manera que los que eligió primero vayan primero y luego los que eligió después, pero se da cuenta de que a veces los últimos números de la primera lista son los mismos que los primero de la segunda lista, así que decide que no quiere repetirlos, y quiere que su lista final sea lo mas corta posible.

Por ejemplo si en la primera lista eligió: 89, 2, 3, 4 y en la segunda lista 3, 4, 9, 8, la lista final que tendrá Pepe será 89, 2, 3, 4, 9, 8.

Esto es por que:

89 2 3 4

3 4 9 8

-----  
89 2 3 4 9 8

Pepe ahora quiere hacer esto con muchos números pero es un tanto flojo y conociendo tus habilidades para programar te pide que le ayudes.

### Problema

Dados **n** y **m** y las dos listas de números, di cuanto mide la lista mas corta en la que puede meter los números y la lista de los número en el orden que deben ser, siguiendo las reglas que te dió Pepe.

### Consideraciones:

- Todos los números que están escritos en el pizarrón son menores que 1000
- La maestra de Pepe se llama Camila.
- Pepe a lo mas eligira 500 números en la primera lista y 500 en la segunda lista. Esto es  $1 \leq n, m \leq 500$ .
- El problema se llama “merge” por que se trata de unir.

### Entrada

En la primera línea están los dos números enteros **n** y **m** separados por un espacio. En las siguientes dos líneas, vienen **n** y **m** números enteros respectivamente.

Domingo 19 de febrero de 2006



**VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)**



**Salida**

En la primera línea el tamaño de la lista mas corta de la lista que pide Pepe , y en la segunda la lista de los números **separados por un espacio cada uno**.

| Entrada  | Salida     |
|----------|------------|
| 4 4      | 6          |
| 89 2 3 4 | 89 2 3 4 9 |
| 3 4 9 8  |            |



VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)



## El Ladrón y la Mochila

**Problema:** mochila.cpp

**Entrada:** input.txt

**Salida:** output.txt

### Historia

Una noche un ladrón entró clandestinamente a una casa a robar. Llevaba una mochila donde le caben alguna cantidad de objetos, dependiendo de su tamaño. Cada objeto tiene algún valor y un tamaño. Lo que el ladrón quiere, es robarse lo más que pueda, esto es, maximizar el valor de los objetos que le quepan en la mochila.

Tú como complice del ladrón, tienes que ayudarle a escoger que objetos conviene llevarse.

### Problema

Debes hacer un programa dados los datos, de los objetos y la mochila, calcule el monto máximo que pudo haber sido robado.

### Entrada

La primera línea contendrá 2 enteros separados por un espacio:  $m \leq 1\,000\,000$  y  $n \leq 100$ ., capacidad de la mochila y cantidad de objetos respectivamente.

La segunda línea contendrá  $n$  enteros separados por espacios menores o iguales a  $1\,000\,000$ , el donde el  $i$ -ésimo entero representará el valor de cada objeto.

La última línea contendrá  $n$  enteros separados por un espacio, donde cada entero representa el tamaño de cada objeto.

### Salida

La primera y única línea deberá contener 1 entero el cual deberá representa el monto máximo que pudo haber sido robado.

| Entrada                        | Salida |
|--------------------------------|--------|
| 12 5<br>2 3 4 8 7<br>4 6 2 1 5 | 21     |



VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)



## Tanques

**Problema:** tanques.cpp

**Entrada:** input.txt

**Salida:** output.txt

### Historia

Por situaciones ajenas a tu persona te has visto envuelto en una guerra urbana, tu deber en esta guerra es manejar un tanque y destruir a todos los tanques enemigos que sean localizados en la ciudad.

La ciudad en la que te encuentras es una cuadrícula, todas las calles son verticales u horizontales y todas las calles (ya sea verticales u horizontales) van desde un extremo de la ciudad hasta el opuesto. Además todas las calles están espaciadas exactamente la misma distancia.

Por facilidad se ha representado a la ciudad en una cuadrícula donde cada par  $(x,y)$  de coordenadas enteras representa el cruce de dos calles de la ciudad. En este mapa las calles van desde -1000 hasta 1000 tanto en la dirección horizontal como en la vertical.

Al inicio del programa te encuentras con tu tanque en la posición  $(0,0)$ .

En algunos cruces de calles hay radares que detectan a los tanques enemigos, por lo que cada que algún tanque pasa por este cruce se te avisa que se ha localizado un tanque. Cada que se localice un tanque, tu tarea consiste en destruirlo, para hacerlo debes desplazar tu tanque hasta que se encuentre en la misma coordenada  $x$  o en la misma coordenada  $y$  que el tanque enemigo, ya que tu tanque puede disparar a cualquier distancia dentro de la ciudad, siempre y cuando tenga en la mira al tanque enemigo, una vez que tengan alguna coordenada en común puedes disparar y destruirlo.

Los tanques enemigos irán llegando de uno por uno, y debes destruir cada tanque antes de que llegue el siguiente. Los tanques enemigos no se mueven mientras tú te desplazas.

Cuando mueves tu tanque la distancia que lo moviste se puede calcular sumando la distancia recorrida en  $x$  con la distancia recorrida en  $y$ .

Debes escribir un programa que dado el número  $T$  de tanques enemigos que van a llegar y las coordenadas  $(x,y)$  de cada uno de los tanques determine cuál es la distancia *minima* que debes recorrer para destruirlos todos.

### Entrada

Tu programa deberá leer del teclado los siguientes datos: En la primera línea el número  $T$  de tanques enemigos. En las siguientes  $T$  líneas habrá 2 números enteros separados por un espacio que indican las coordenadas  $X, Y$  donde es localizado el tanque enemigo respectivo.

El número de  $T$  de tanques enemigos puede estar entre 2 a 80, Las coordenadas  $x,y$  pueden estar entre -1000 y 1000.

### Salida

Tu programa deberá escribir en la pantalla un único número que indique la distancia *minima* que se tiene que recorrer para destruir a todos los tanques enemigos.



VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)



| Entrada                   | Salida |
|---------------------------|--------|
| 3<br>4 8<br>-2 8<br>-3 10 | 10     |



VII Olimpiada de Informática  
del estado de Guanajuato  
Tercer Examen Selectivo  
Examen Febrero (Parte 1.- C++)



## Swap Matriz

**Problema:** swap.cpp

**Entrada:** input.txt

**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  $n$  renglones, 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 reescribir 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  $a \leq m \leq 10000$ . 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  $m \times n$  (renglón, columna) ordenada de calificaciones.

| Entrada                          | Salida           |
|----------------------------------|------------------|
| 2 3<br>10 0 5<br>7 10 5<br>1 2 0 | 0 5 10<br>10 5 7 |