

Segundo Concurso Local de Programación ACM ICPC

Universidad Autónoma Metropolitana Unidad Azcapotzalco

Examen eliminatorio, 6 de Octubre de 2005

Instrucciones: Abajo encontrarás los enunciados de cuatro problemas.

Para cada problema que resuelvas deberás dejar en el directorio de trabajo (acm\ si trabajas en Linux ó c:\acm si trabajas en Windows XP) los códigos fuente de un programa. El nombre de estos archivos debe ser de la forma xxxNN.zzz, donde xxx es el nombre del problema, NN es un número de identificación que se te dará al comenzar el examen y zzz es la extensión, según el tipo de archivo. Si así lo deseas, puedes resolver cada problema en un lenguaje distinto. Tu programa deberá leer e imprimir exactamente los datos que se indican (ni más ni menos) usando la entrada y la salida estándar. En particular, tu programa no deberá borrar la pantalla, escribir ningún tipo de letrero adicional, usar la biblioteca conio, etc.

Cada problema tiene un valor de 50 puntos, los cuales se obtendrán de las salidas que su programa entregue para cada una de 10 entradas distintas. **Todos los programas se evaluarán en Linux.** Que tu programa funcione con el ejemplo no quiere decir que funcionará siempre.

Está permitido el uso de libros o notas de curso siempre y cuando no contengan códigos fuente de problemas resueltos.

Cuestiones técnicas para Linux: Se te otorgará una cuenta y su clave en una máquina con Linux, a la que podrás conectarte a través del Secure Shell usando la dirección 148.206.67.155. En todas las cuentas debe de existir un directorio de trabajo llamado acm\ (en caso contrario, éste se puede crear con las tres instrucciones (a) cd ~ (b) mkdir acm (c) cd acm). Quienes programen en C deberán usar el gcc, quienes programen en C++ deberán usar el g++ y quienes programen en Java deberán usar el javac de la siguiente manera:

```
gcc -ansi -o nombre_del_ejecutable nombre_del_fuente.c -lm
g++ -ansi -o nombre_del_ejecutable nombre_del_fuente.cpp -lm
javac nombre_del_fuente.java
```

Cuestiones técnicas para Windows XP: En todas las computadoras debe de existir un directorio de trabajo llamado c:\acm (en caso contrario, éste se puede crear desde el símbolo del sistema con las tres instrucciones (a) cd c:\ (b) mkdir acm (c) cd acm). Quienes programen en C o C++ podrán usar el **DJGPP**, el cual se puede ejecutar con la instrucción rhide desde el directorio de trabajo. Quienes programen en Java podrán usar el **Sun Java SDK**. Puede ser que el día del examen existan otros ambientes de desarrollo disponibles para Windows XP. Si algo no funciona, repórtalo inmediatamente.

/* :-) Los organizadores te deseamos éxito en tu examen (-: */

Problema 1: Extremos de una función racional

Código fuente: efrNN.c, efrNN.cpp, efrNN.java

Escribe un programa que determine el valor máximo **p** y el valor mínimo **q** que toma la función $f(x) = (ax + b)/(cx + d)$ en los enteros que se encuentran en el intervalo dado por **m** y **n** (incluyéndolos). Observa que es posible que la función no esté definida en todos los puntos de ese intervalo. Además, deberá determinar los enteros **r** y **s** en los cuales se alcanzan estos valores (es decir, tales que $f(r) = p$ y $f(s) = q$). En caso de que haya varios enteros **r** y **s** con esa propiedad, se deben dar aquellos cuyos valores sean los menores posibles.

Por ejemplo, si $f(x) = (x + 2)/(x - 2)$ y el intervalo está dado por **m** = -2 y **n** = 3 entonces vemos que $f(-2) = 0$, $f(-1) = -1/3$, $f(0) = -1$, $f(1) = -3$, $f(2)$ no está definida y $f(3) = 5$, por lo que **r** = 3 y **s** = 1.

Entrada: Seis números enteros **a**, **b**, **c**, **d**, **m**, **n** (con $m \leq n$) separados por espacios y todos ellos en el intervalo de -1000 a 1000 (incluyéndolos). Puedes suponer que el intervalo $m \leq x \leq n$ contiene al menos un entero **x** en el cual la función $f(x)$ está definida (y por lo tanto el máximo y el mínimo existen).

Salida: Dos números enteros **r**, **s**, separados por espacios.

Evaluación: 2 puntos por cada uno de **r**, **s**, 1 punto mas si ambos están bien.

Ejemplo de entrada	Ejemplo de salida
1 2 1 -2 -2 3	3 1

Problema 2: Sumas de dos cuadrados

Código fuente: sdcNN.c, sdcNN.cpp, sdcNN.java

Algunos números enteros se pueden representar como suma de dos enteros al cuadrado, por ejemplo $50 = 1^2 + 7^2 = 5^2 + 5^2$. Escribe un programa que determine la cantidad de formas distintas **t** en las que se puede escribir un número entero **n** como suma de dos cuadrados $a^2 + b^2$ donde $0 \leq a \leq b \leq n$.

Por ejemplo, si **n** = 50 entonces hay dos formas, por lo que **t** = 2.

Entrada: Un número entero **n** tal que $1 \leq n \leq 2,000,000,000$.

Salida: Un número entero **t**.

Evaluación: 5 puntos por el valor de **t**.

Ejemplo de entrada	Ejemplo de salida
50	2

Problema 3: Dos bases numéricas

Código fuente: dbnNN.c, dbnNN.cpp, dbnNN.java

Si un entero n se escribe en base b como $a_m a_{m-1} \dots a_1 a_0$ esto quiere decir que $n = a_m b^m + a_{m-1} b^{m-1} + \dots + a_1 b^1 + a_0 b^0$ y que $0 \leq a_i < b$ para cada $0 \leq i \leq m$. Por supuesto, al escribir el mismo número n en bases distintas se obtienen expresiones distintas, por ejemplo, el número decimal $n = 31$ se escribe 11111 en base 2 y 1011 en base 3. Escribe un programa al cual se le dan dos expresiones A y B en bases desconocidas p y q que representan al mismo número n y que encuentre ese valor. Si hay varios números que se puedan escribir de esas dos formas, debe encontrar al más pequeño de ellos.

Por ejemplo, si se dan las expresiones 11111 y 1011 entonces $n = 31$, ya que 11111 en base 2 es igual a 1011 en base 3 que es igual a 31 en decimal.

Entrada: Dos expresiones A y B , cada una con un máximo de 64 dígitos, separadas por un espacio. La primera expresión estará en base p y la segunda en base q . Puedes suponer que $2 \leq p, q \leq 10$ y que $1 \leq n \leq 4,000,000,000$.

Salida: Tres números enteros n, p, q (en decimal) separados por espacios.

Evaluación: 1 punto por el valor de n , 2 puntos por cada uno de p y q .

Ejemplo de entrada	Ejemplo de salida
11111 1011	31 2 3

Problema 4: Matriz con entradas en diagonal

Código fuente: medNN.c, medNN.cpp, medNN.java

Dados dos enteros m y n , escriba un programa que construya una matriz con m renglones y n columnas cuyas entradas sean los números 1, 2, ..., $m*n$ acomodados en diagonal, comenzando con el 1 en la entrada que está en la esquina superior izquierda, siguiendo con el 2 a la derecha del 1 y el 3 abajo del 1, y así sucesivamente.

Entrada: Dos números enteros $2 \leq m, n \leq 100$, separados por un espacio.

Salida: La matriz requerida (ver el ejemplo de salida para el formato).

Evaluación: 5 puntos por la salida correcta.

Ejemplo de entrada	Ejemplo de salida
3 5	1 2 4 7 10 3 5 8 11 13 6 9 12 14 15