

Sexto Concurso Local de Programación ACM ICPC

Universidad Autónoma Metropolitana

Examen eliminatorio, 1 a 6 de octubre de 2009

Instrucciones: Abajo encontrarás los enunciados de cinco problemas. Cada problema tiene un valor entre 20 y 60 puntos, los cuales se obtendrán de las salidas que entregue tu programa 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.

Para cada problema que resuelvas deberás enviar por correo electrónico el código fuente de un programa a la dirección `uam09acm@gmail.com` indicando claramente tu nombre completo. De preferencia envía todos tus códigos en un correo. El nombre de los archivos que envíes debe ser de la forma `xxx.zzz`, donde `xxx` es el nombre del problema y `zzz` es la extensión, según el lenguaje que hayas usado. 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. Quienes programen en C deberán usar `gcc`, quienes programen en C++ deberán usar `g++` y quienes programen en Java deberán usar `gcj` como sigue:

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

Deberás enviar tus códigos fuente a más tardar el **6 de octubre de 2009 a las 22:00**.

Problema 1: Secuencia creciente máxima (20 puntos)

Código fuente: `scm.c`, `scm.cpp`, `scm.java`

Sea n un entero positivo y sea a_1, a_2, \dots, a_n una secuencia de n enteros positivos. Sean h y k dos enteros con $1 \leq h < k \leq n$. Decimos que a_h, a_{h+1}, \dots, a_k es una secuencia creciente si $a_h < a_{h+1} < \dots < a_k$ y decimos que es máxima si no es parte de otra secuencia creciente más larga. Por ejemplo, si $n = 7$ y $a = (3, 1, 4, 1, 5, 9, 2)$ entonces $(1, 5)$ es una secuencia creciente, pero no es máxima porque es parte de $(1, 5, 9)$. Escribe un programa que encuentre la longitud s de la secuencia creciente máxima más larga de todas y la posición p donde inicia.

Entrada: Un número entero n con $1 \leq n \leq 1000$ seguido de n números enteros a_1, a_2, \dots, a_n separados por espacios y en el intervalo de 1 a 1,000,000.

Salida: Un número entero s y un número entero p .

Evaluación: 1 punto por el valor correcto de s y 1 punto si p es correcto.

<i>Ejemplo de entrada</i>	<i>Ejemplo de salida</i>
7 3 1 4 1 5 9 2	3 4

Problema 2: Área de la intersección de dos rectángulos (30 puntos)

Código fuente: `air.c`, `air.cpp`, `air.java`

Considere que tiene dos rectángulos con lados paralelos a los ejes de coordenadas: uno con vértices opuestos (u, v) y (x, z) y el otro con vértices opuestos (p, q) y (r, s) . Escribe un programa que calcule el área a de la

intersección de esos dos rectángulos. Por ejemplo, si un rectángulo tiene vértices opuestos (3, 1) y (1, 4) y el otro tiene vértices opuestos (2, 7) y (1, 2) entonces su intersección tiene área $a = 2$.

Entrada: Ocho números enteros u, v, x, z, p, q, r y s separados por espacios y todos ellos en el intervalo de 0 a 1,000,000.

Salida: Un número entero a .

Evaluación: 3 puntos por el valor correcto de a .

<i>Ejemplo de entrada</i>	<i>Ejemplo de salida</i>
3 1 1 4 2 7 1 2	2

Problema 3: Boletos por besos (40 puntos)

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

Para ir hacia su escuela, Juan aborda un camión. Cuando Juan sube al camión y deposita las monedas en la alcancía, el conductor le da un boleto con un número de serie consecutivo, el cual generalmente llega a la basura. Sin embargo, un día él se entera de algo interesante. Si la suma de los dígitos de dicho boleto es 21, él lo puede cambiar con quien quiera por un beso. Juan quiere conseguir muchos besos de la persona que está enamorado y ella ha aceptado dárselos a cambio de dichos boletos, pero no es tan fácil conseguirlos. Juan ha decidido que si el boleto que le da el conductor no suma 21 esperará hasta que el de una persona que suba después de él sume 21, y le pedirá a esta su boleto, para lo cual debe ir realizando las cuentas a mano, cosa que es muy tediosa. Ayuda a Juan escribiendo un programa que dado un número n de 7 dígitos, le proporcione el número p de personas que deben abordar después de él para que obtenga el boleto deseado y m , donde m representa los 7 dígitos de dicho boleto. Cuando un tiraje de boletos termina, es decir llega a 9999999, el conductor toma otro tiraje comenzando nuevamente en el número 0000000.

Entrada: Un número entero n de 7 dígitos (que posiblemente comienza con ceros).

Salida: Un número entero p y un número m de 7 dígitos.

Evaluación: 2 puntos por el valor correcto de p y 2 puntos por el valor correcto de m .

<i>Ejemplo de entrada</i>	<i>Ejemplo de salida</i>
0031416	33 0031449

Problema 4: Menor base para palíndromo (50 puntos)

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

Algunos enteros se pueden leer de la misma forma sin importar si se escriben de izquierda a derecha o de derecha a izquierda (obviamente sin cambiar de valor) como por ejemplo 33 en base 10 es palíndromo, pero también en base 2, ya que 33 en decimal es igual a 100001 en binario. Dado un número n en base 10, escribe un programa que proporcione la menor base b tal que $2 \leq b \leq 10$, en la que este se puede representar como palíndromo, así como su representación r , en caso de no existir dicha base, la salida debe de ser $b = -1$ y $r = 0$.

Entrada: Un número entero n en decimal tal que $0 < n < 2,147,483,647$.

Salida: Un número entero **b** y un numero **r** representado en base **b**.

Evaluación: 2 puntos por el valor correcto de **b** y 3 puntos por el valor correcto de **r**.

<i>Ejemplo de entrada</i>	<i>Ejemplo de salida</i>
33	2 100001

Problema 5: Acomodando una pila de hotcakes (60 puntos)

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

José es muy buen cocinero, pero es un poco desordenado. Por ejemplo, cuando hace hotcakes todos le quedan de tamaños distintos. Como al final debe darle una buena presentación a todo lo que sirve, con frecuencia tiene que volver a acomodar el platillo que ha preparado. En el caso de los hotcakes él desea que queden apilados de modo que los más pequeños queden encima de los más grandes. Como no va a usar las manos para acomodarlos (uno imaginaria que las tiene limpias... seguro es porque los hotcakes están calientes) decide usar una pala de la siguiente manera: mete la pala abajo de cierta cantidad de hotcakes y de un solo movimiento los voltea de modo que quedan exactamente en el orden inverso de como estaban. Por ejemplo, si preparó $n = 7$ hotcakes y sus tamaños respectivos son 3, 1, 4, 2, 5, 7 y 6 (de arriba hacia abajo en la pila) entonces puede meter la pala bajo los primeros 4 hotcakes y voltearlos, de modo que ahora quedan en el orden 2, 4, 1, 3, 5, 7 y 6. Escribe un programa que le ayude a José a acomodar sus hotcakes en tan pocos movimientos **m** como sea posible.

Entrada: Un número entero **n** con $1 \leq n \leq 10$ seguido de **n** números enteros distintos **a**₁, **a**₂, ..., **a**_n separados por espacios y en el intervalo de 1 a 100.

Salida: Un número entero $m \leq 2n$ seguido de **m** enteros **b**₁, **b**₂, ..., **b**_m separados por espacios, indicando bajo cuántos hotcakes se metió la pala en cada movimiento.

Evaluación: 1 punto si $m \leq 2n$ y los **m** movimientos indicados dejan la pila de hotcakes en orden de menor a mayor. En ese caso, $5M/m$ puntos adicionales, donde **M** es la cantidad mínima de movimientos necesaria para acomodar la pila de hotcakes.

<i>Ejemplo de entrada</i>	<i>Ejemplo de salida</i>
7 3 1 4 2 5 7 6	8 6 7 6 5 3 4 2 3

/* el comité organizador del concurso te desea suerte en tu examen */