

POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
Examen Final Procedatos

Guillermo Buriticá Tobón

25 de junio de 2008

Sumas Largas

En algunos lenguajes de programación se utilizan 16 bits de memoria para almacenar las variables de tipo entero en donde el primer bit se utiliza para determinar el signo de la variable, lo que nos permite almacenar valores entre $-2^{15} = -32.768$ y $2^{15} - 1 = 32.767$. Una forma de representar números enteros más grandes en estos lenguajes de programación es utilizar arreglos donde en cada posición se almacene una cifra del número. Con esta idea, construya un programa que permita sumar dos enteros positivos de hasta 50 cifras.

Mensajeros Poco Fiables

Al rey de un pequeño reino ubicado en una pequeña isla del Océano Pacífico frecuentemente se le ocurren ideas infantiles. Un día dijo: “Se debe utilizar una secuencia de mensajeros cuando deseen informarme algo”. En respuesta a la orden del rey, seis sirvientes (cuyos nombres eran Sr. J, Srta. C, Sr. E, Sr. A, Dr. P y Sr. M) fueron seleccionados como mensajeros. Ellos debían ir pasando el mensaje al siguiente mensajero hasta que llegara al rey.

Los mensajes enviados al rey consisten de dígitos ('0'-'9') y letras ('a'-'z', 'A'-'Z') en donde es importante distinguir las mayúsculas de las minúsculas. Por ejemplo, “ke3E9Aa” es un mensaje.

Contrario a las expectativas del rey, siempre recibía mensajes incorrectos, ya que cada mensajero cambiaba el mensaje un poco, antes de pasarlo al siguiente. Dado que esto irritaba al rey, le pidió a usted, que se desempeña como Ministro de Ciencia y Tecnología del reino: “No deseamos recibir mensajes incorrectos. Usted debe desarrollar un software para corregirlos!”. En respuesta a la solicitud del rey, usted analizó los errores cometidos por los mensajeros utilizando toda la tecnología disponible en el reino y logró caracterizar los errores de cada mensajero. Algo sorprendente es que cada mensajero siempre cometía el mismo error cuando pasaba un mensaje.

- El Sr. J rota todos los caracteres del mensaje una posición hacia la izquierda. Por ejemplo, transforma “aB23d” en “B23da”.
- La Srta. C rota todos los caracteres del mensaje una posición hacia la derecha. Por ejemplo, transforma “aB23d” en “daB23”.
- El Sr. E intercambia la mitad izquierda del mensaje con la mitad derecha. Si el mensaje tiene un número impar de caracteres, el de la mitad no se mueve. Por ejemplo, transforma “e3ac” en “ace3”, y “aB23d” en “3d2aB”.

- El Sr. A invierte el mensaje. Por ejemplo, transforma “aB23d” en “d32Ba”.
- El Dr. P incrementa en una unidad todos los dígitos del mensaje. Si un dígito es ‘9’, se convierte en ‘0’. Las letras no se modifican. Por ejemplo transforma “aB23d” en “aB34d”, y “e9ac” en “e0ac”.
- El Sr. M decreta en una unidad todos los dígitos del mensaje. Si un dígito es ‘0’, se convierte en ‘9’. Las letras no se modifican. Por ejemplo transforma “aB23d” en “aB12d”, y “e0ac” en “e9ac”.

El software que usted debe desarrollar debe inferir el mensaje original a partir del mensaje final, dado el orden de los mensajeros. Por ejemplo, si el orden de los mensajeros es A J M P y el mensaje entregado al rey es “aB23d”, ¿cuál es el mensaje original? De acuerdo a la caracterización de los errores que cometen los mensajeros, la secuencia que lleva al mensaje final es “32Bad” \xrightarrow{A} “daB23” \xrightarrow{J} “aB23d” \xrightarrow{M} “aB12d” \xrightarrow{P} “aB23d”. Por lo tanto, el mensaje original debe ser “32Bad”.