

1. Imagen

Gracias a sus habilidades en el curso de Programación de Computadores fue seleccionado por los profesores de Computación Gráfica para que trabaje en el área de procesamiento de imágenes. La tarea será trabajar en un nuevo algoritmo de compresión y codificación de imágenes.

Después de analizar la tarea se decide por los siguientes métodos, para la **codificación** un sistema de sustitución simple y para la **compresión** un sistema de codificación de largo de línea. Los métodos que selecciona funcionan de la siguiente manera.

Codificación por sustitución simple:

En éste método se cuenta con el conjunto de símbolos clave, cada símbolo representa una letra del alfabeto o un dígito, por ejemplo, una clave podría ser (%=P, \$=U, a=J), en éste caso el mensaje "%\$a" significa en realidad "PUJ".

Compresión por largo de línea:

Éste tipo de compresión es muy útil para imágenes que tienen colores similares, en ésta compresión la imagen se va barriendo línea por línea y se forman grupos de puntos del mismo color. Supongamos que la siguiente figura es una imagen que queremos comprimir, para el ejemplo, sustituiremos cada punto de la imagen por una letra del alfabeto.

AAAAABBB
BBBCCCC
CBAACC
CCBBBBB

La compresión queda **A5B5C5B1A3C4B5**

Indicando que la imagen tiene 5 puntos del color A, después 5 del B, 5 del C, 1 del B, etc. En la versión del método de compresión que está usando **la imagen comienza siempre con un número que indica el ancho en puntos de la imagen**. Además, en los casos en los que solo hay un punto de cierto color (como en el punto B en el tercer renglón de la imagen) no le sigue ningún número. Por último, los grupos de puntos nunca

son de más de 9 puntos, es decir, no puede haber un A25. En el caso de que hubiera 25 puntos del color A consecutivos, se tendría que poner A9A9A7. La imagen comprimida completa queda **7A5B5C5B1A3C4B5**. El **ancho** en puntos de la imagen siempre es un número entre 2 y 500.

Debe escribir un programa haciendo uso de funciones que teniendo como entrada la clave de codificación y la imagen codificada, sea capaz primero de decodificar la imagen y posteriormente de crear la imagen.

ENTRADA

El programa deberá leer dos cadenas de caracteres. La primera cadena está formada por 36 símbolos. Esta cadena es la **clave de codificación**. Los primeros 26 símbolos representan las letras mayúsculas A..Z, sin contar la Ñ, CH, LL, ni la RR. Los últimos 10 símbolos representan los dígitos del 0..9. Los 36 símbolos están consecutivos y no tienen ningún espacio separándolos. La segunda cadena está formada también por una serie de símbolos consecutivos que puede tener hasta 500 símbolos. La segunda cadena representa la **imagen comprimida y codificada**.

SALIDA

En este problema, la salida consta de 2 partes: La primera es una cadena que representa la imagen comprimida La segunda parte es crear la imagen, utilizando letras.

EJEMPLO

ENTRADA						
Primera Cadena :						
A%ab&cd84)kegh792341L2xzv'm/'"#¿-.;=!						
Segunda Cadena : ;A-%-a-%A#a¿%-						
SALIDA						
Imagen Comprimida : 7A5B5C5BA3C4B5						
Imagen utilizando letras :						
A	A	A	A	A	B	B
B	B	B	C	C	C	C
C	B	A	A	A	C	C
C	C	B	B	B	B	B

Elabore:

- Una función que reciba las dos cadenas de acuerdo al ejemplo y devuelva la cadena con la **imagen comprimida** (la cadena que devuelve debe utilizar memoria dinámica).
- Una función que reciba la **imagen comprimida** y devuelva la imagen almacenada en una matriz utilizando letras.

(Esta matriz debe ser creada usando memoria dinámica y los conceptos de apuntadores, no se puede usar la notación tradicional de vectores []).

- El programa Principal. (imprima las dos cadenas de entrada, la cadena con la imagen comprimida que se devuelve de la primera función y la imagen utilizando las letras que se devuelven en la segunda función).

2. Faltante

Amanda compró una tarjeta para recarga de su celular, pero debido a que se encontraba en el sol al rasparle (aunque lo hizo con cuidado) se borró un número de la clave de la tarjeta con la cual al introducirlo en su celular no podría gozar de su saldo disponible. Amanda desesperada hablo al servicio al cliente y le dijeron lo siguiente:

“Para poder encontrar el número faltante primero obtendrá la suma de los valores **ASCII** de los caracteres a la izquierda del número faltante y después la suma de los valores ASCII de la derecha de dicho número faltante. Posteriormente los compara y se quedará con el número más grande (Si son iguales se quedará con cualquiera de los dos), ya con el resultado sumará entre si todas las cifras obteniendo un número, repetirá ese procedimiento hasta que solo quede un numero de una sola cifra ese es el numero faltante (Es decir si la suma dio 391, entonces $3 + 9 + 1 = 13$; $1 + 3 = 4$, en este caso el 4 es el numero buscado)”

Amanda se quedó pasmada con la explicación así que corrió a solucionar su “PROBLEMA” ya que posee conocimientos de Pensamiento Algorítmico y Programación de Computadores.

ENTRADA

Un número N indicando cuantas claves va a leer. Posteriormente va a leer N cadenas donde cada una representa la clave de la tarjeta formada por caracteres del abecedario tanto en mayúsculas y minúsculas además de números (No existirá ningún otro carácter a excepción del carácter '?'), el carácter faltante que busca Amanda estará

representado dentro de la cadena por el carácter '?'.

SALIDA

Para cada una de las claves obtendrá el número que representa al carácter faltante en cada clave y en cada clave reemplaza el carácter '?' por el número que se obtuvo, va a devolver todas las claves corregidas al programa principal.

EJEMPLO

ENTRADA	SALIDA
3	
aZ34?zz	aZ34 2 zz
wdcx?zxw01	wdcx 8 zxw01
1234?4321	1234 4 4321

Elabore

- Una función que reciba el número de claves a considerar y las claves y devuelva las claves con el número correcto en cada una de ellas. (La función sólo se puede llamar una vez desde el programa principal).
- El programa Principal. (imprima las claves de las tarjetas donde se vea donde falta el carácter) y posteriormente las claves con el carácter faltante en cada una de ellas.

Recuerde hacer uso de los conceptos de apuntadores, memoria dinámica y puede usar las funciones de la librería string.h si lo requiere.