

Primer Parcial de Programación Imperativa (72.31)

Segundo Cuatrimestre 2011 – 22/09/2011

Ej. 1	Ej. 2	Ej. 3	Ej. 4	Nota
/ 2.5	/ 2.5	/ 2.5	/ 2.5	

- ❖ Condición mínima de aprobación: Sumar al menos 5 puntos.

❖ Se tendrá en cuenta en la calificación el ESTILO y la EFICIENCIA de los algoritmos.

❖ Los ejercicios que no se ajusten estrictamente al enunciado, no serán aceptados.

- ❖ Puede entregarse en lápiz

❖ Pueden usarse las funciones de la librería estándar y de getnum

❖ No es necesario escribir los #include

Ejercicio 1:

Escribir un programa que lea un texto carácter a carácter de entrada estándar hasta EOF y envíe a la salida estándar el mismo texto pero transformando los números decimales a su expresión hexadecimal. Para este ejercicio no se pueden utilizar arreglos.

Se asegura que todos los números que figuran en el texto entran en un entero

Ejemplos:

Si la entrada es

hubo 56 patentes entre HBC000 y HZA123 con más de 340 multas, abonadas antes del 24/05 del 2011

La salida es

hubo 38 patentes entre HBC000 y HZA123 con más de 154 multas, abonadas antes del 24/05 del 7DB

Si la entrada es

no confundir los números con dígitos que forman parte de una palabra, no es lo mismo 64 que a64b, o -64 que a-64

La salida es

no confundir los números con dígitos que forman parte de una palabra, no es lo mismo 40 que a64b, o -40 que a-64

Ejercicio 2:

Escribir una función que reciba dos vectores de enteros y la dimensión de cada uno y retorne:

- 1 si todos los elementos del primer vector están en el segundo
- 2 si todos los elementos del segundo vector están en el primero
- 0 en caso contrario

Ejemplos:

Vector 1	Vector 2	Respuesta
{1, 6, 5, 3, 2}	{1, 2}	2
{1, 3, 2}	{3, 1, 2}	1 ó 2
{}	{5, 8, 54}	1
{1}	{2}	0
{1, 1, 5, 5, 9, 1}	{5, 9, 3, 1}	1
{6, 8}	{6, 6, 6}	2
{}	{}	1 ó 2

Ejercicio 3:

Escribir una función que reciba una matriz de FILS filas y COLS columnas, un número de fila y un número de columna, y reemplace todos los valores de dicha fila y de dicha columna por el máximo entre los cuatro promedios de los sectores que la fila y columna determinan. La función retorna 1 si pudo realizar la operación con éxito y 0 en caso contrario.

Ejemplo: suponiendo FILS igual a 5, COLS igual 6

Si la matriz es:

3	4	4	6	5	4
3	2	12	7	8	9
0	8	9	0	8	1
2	7	8	9	0	9
9	7	6	5	4	3

Y se recibe fila 2 y columna 3:

3	4	4	6	5	4
3	2	12	7	8	9
0	8	9	0	8	1
2	7	8	9	0	9
9	7	6	5	4	3

El promedio de cada uno de los cuatro sectores es:

Sector 1:  $(3+4+4+3+2+12)/6 = 28/6 = 4,66$

Sector 2:  $(5+4+8+9)/4 = 6,50$

Sector 3:  $(2+7+8+9+7+6) / 6 = 6,50$

Sector 4:  $(0+9+4+3)/4 = 4$

El mayor promedio es 6,50. Tomado como entero, es 6. La matriz resulta:

3	4	4	6	5	4
3	2	12	6	8	9
6	6	6	6	6	6
2	7	8	6	0	9
9	7	6	6	4	3

Con la misma matriz, si se recibe fila 5, retorna cero. Si recibe fila 0 retorna cero.

Ejercicio 4:

Escribir una función que reciba los siguientes parámetros:

- un string **s**
- un char **c**
- un vector de enteros **vec**
- un entero **maxDim**

La función debe eliminar del string **s** las repeticiones consecutivas del caracter **c**. Deberá asimismo almacenar en **vec** cuántas repeticiones fueron eliminadas en cada oportunidad, para que el string se pueda reconstruir.

El vector **vec** puede almacenar hasta **maxDim** elementos.

La función debe retornar un valor entero indicando cuántas apariciones se pudieron procesar

Ejemplo:

```
int
main(void)
{
    char s1[20] = "Aaa, bb aaaa cc aaa";
    char s2[20] = "Aaa bb aaaa cc aaa";
    int vec[10];
    int dim;

    dim = comprime(s1, 'a', vec, 20);      /* dim es 3, vec es {1, 3, 2} */
    printf("%s\n", s1);    /* imprime Aa, bb a cc a */

    dim = comprime(s2, 'a', vec, 2); /* dim es 2, vec es {1, 3} */
    printf("%s\n", s2);    /* imprime Aa bb a cc aaa */
}
```