**Resuelva los siguientes problemas, cada uno vale 25%.**

1. Realice la función **int complementodig (int num),** la cual retorna el número compuesto con el complemento a 10 de cada dígito del número “num”. Por ejemplo, si **num** es el entero 29415; la llamada **complementodig (num)** retornaría 81695.

int complementodig (int num)

{

int dignuevo, cantdig, numnuevo = 0;

double denominador;

while ( num )

{

cantdig = log10(num);

denominador = pow((double)10,(double)cantdig);

dignuevo = 10 - (num / (int) denominador);

numnuevo = numnuevo \* 10 + dignuevo;

num %= (int)denominador;

}

return numnuevo;

}

1. La traza de una matriz cuadrada es la suma de los elementos de la diagonal superior. Se dice que una matriz es mayoritaria si la cantidad de elementos no pertenecientes a la diagonal superior, que son mayor que la traza, sobrepasa la cantidad de elementos no pertenecientes a la diagonal superior que son menores o iguales a la traza. Si se conoce la función **float traza(float m[ ][COL], int n)** que determina la traza de la matriz **m**, realice la función **int esmatmayoritaria(float m[ ][COL], int n)** que retorne **1** si **m** es mayoritaria y **0**  si **m** no lo es.

int esmatmayoritaria(float m[ ][COL], int n)

{

int fila, columna, cantvalmay = 0, cantvalmen = 0;

float trazamat = traza(m,n);

for ( fila = 0; fila < n; fila ++)

for (columna = 0; columna < n; columna ++ )

if ( fila != columna )

{

if ( m[fila][columna] > trazamat )

cantvalmay ++;

else

cantvalmen ++;

}

// return cantvalmay > cantvalmen;

if (cantvalmay > cantvalmen )

return 1;

return 0;

}

1. Dado la cadena de caracteres **s,** se desea saber cuál es el promedio de repetición de sus caracteres. Realice la función **float promchar(char s[ ])**, que retorne el promedio de repetición de los caracteres de **s.**

int veces(char cadena[], char caract)

{

int ind, contador = 0;

for ( ind = 0; cadena[ind] != '\0'; ind ++ )

if ( tolower(cadena[ind]) == tolower(caract) )

contador ++;

return contador;

}

float promchar(char s[ ])

{

int frecuencia[MAXABC] = {0};

char caract;

int ind, cantletras = 0;

float suma = 0;

for ( ind = 0; s[ind]; ind ++ )

{

caract = tolower(s[ind]);

if ( caract >= 'a' && caract <= 'z' )

frecuencia[caract - 'a'] = veces(s,caract);

}

for (ind = 0; ind < MAXABC; ind ++ )

if ( frecuencia[ind] != 0 )

{

printf("%c - %d\n",ind+'a',frecuencia[ind]);

suma += frecuencia[ind];

cantletras ++;

}

return suma / cantletras;

}

1. Dados tres arreglos de números de dimensión **n** se desea una función que permita obtener el cuadrado de la diferencia de los dos primeros en el tercer arreglo.

Por ejemplo: Si se tienen 3 arreglos A1, A2, A3 de dimensión 5 y con los siguientes elementos:

A1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12.34 | 23.56 | 18.45 | 20.32 | 10.23 |

A2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10.34 | 18.45 | 32.12 | 25.23 | 18.45 |

Luego de haber llamado la función en cuestión el arreglo A3 tendría:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4.0 | 26.1121 | 186.8689 | 24.1081 | 67.5684 |

void cuaddif(float arr1[],float arr2[], float arr3[], int n)

{

int ind;

for (ind = 0; ind < n; ind ++ )

arr3[ind] = (arr1[ind] - arr2[ind]) \* (arr1[ind] - arr2[ind]) ;

return;

}