IDENTIFICADOR ESTUDIANTE

FUNDAMENTOS DE PROGRAMACIÓN
EXAMEN DEL PRIMER PARCIAL
26 OCTUBRE 2021

APELLIDOS (en MAYÚSCULAS)

NOMBRE (en MAYÚSCULAS)

Los smartwatches y smartphones deben estar apagados. Cada hoja se entrega por separado y es obligatorio entregar todas las hojas. Se debe poner el ***nombre completo en todas las hojas y el identificador del estudiante****. No se podrá entregar hasta pasados* ***20 minutos****. No se pueden usar hojas de sucio, para ello se dispone del espacio sobrante en cada hoja. El examen tendrá una duración máxima de 2:00 h.* ***No se podrá salir de los bucles usando break o return****.*

1.- (2 ptos) Implementar una función que, pasándole como parámetros una matriz M*N caracteres (siendo M y N ctes definidas previamente) y dos letras minúsculas que representan un intervalo (se garantiza que la primera es inferior o igual a la segunda), ponga a ' ' todas las posiciones de la matriz que contengan una letra minúscula comprendida en el rango pasado como parámetro.

```
void a_blancos (char mat[M][N], char a, char b)
{  int i, j;

  for (i=0; i<M; i++)
     for (j=0; j<N; j++)
        if (mat[i][j]>=a && mat[i][j]<=b)
           mat[i][j]=' ';
}
```

2.- (2 ptos) Implementar una función que, pasándole como parámetro una matriz de MxN enteros (siendo M y N ctes definidas previamente), devuelva la fila y la columna en la que se encuentra el mayor valor.

```
void valor_mayor (int mat[N][N], unsigned *f, unsigned *c)
{  int i, j, max=mat[0][0];

  *f = *c = 0;
  for (i=0; i<M; i++)
     for (j=0; j<N; j++)
        if (mat[i][j] > max)
           {  max = mat[i][j];
              *f = i;
              *c = j;
           }
}
```

3.- (2 ptos) Escribir un programa que vaya leyendo de teclado números enteros y los almacene en un una array de N posiciones (siendo N una cte definida previamente) de forma que los negativos se vayan almacenando en la parte izquierda del array y los positivos (excluyendo el 0) en la parte derecha según vayan llegando. El programa dejará de leer datos cuando el array esté lleno e imprimirá el contenido del mismo. Por ejemplo, si N es 10 y se introduce la secuencia de números: 2, 11, -6, -5, 0, -47, -31, 35, 0, 9, 0, 0, -8, 0, -26, el array que se imprimirá será:

-6	-5	-47	-31	-8	-26	9	35	11	2

```
main ()
{  int array[N], i=0, j=N-1, n;

  printf ("introduzca números enteros.\n");
  do
  {  scanf ("%d", &n);
    if (n<0)
       array[i++]=n;
      else
        if (n>0)
          array[j--]=n;
  } while (i<=j);

  printf("\nLos valores almacenados en el array son:\n");
  for (i=0; i<N; i++)
     printf ("%3d", array[i]);
}
```

4.- (2 ptos) Escribir un programa que lea de teclado números enteros (al menos uno) que irá sumando en una variable (inicializada a 0). Por cada número leído imprimirá ese valor y la suma acumulada hasta ese momento. El programa acabará cuando la suma de todos los números leídos sea 0.

```
main ()
{  int n, suma=0;

  printf (“introduzca números enteros.\n);
  do
  {  scanf ("%d", &n);
    suma+=n;
    printf ("\n%6d Suma: %6d.\n", n, suma);
  } while (suma);
}
```

5.- (2 ptos) Implementar una función que, pasándole como parámetro una matriz de NxN dígitos (0..9) (siendo N una cte definida previamente), imprima por pantalla el número de veces que aparece cada dígito en la matriz.

```
void veces_digito (unsigned mat[N][N])
{  int i, j, dígitos[10]={0};

  for (i=0; i<N; i++)
     for (j=0; j<N; j++)
        dígitos[mat[i][j]]++;

  printf ("El número de veces que aparece cada dígito es:\n");
  for (i=0; i<10; i++)
     printf ("%1d%4d\n", i, digitos[i]);
}
```