

--	--	--

APELLIDOS (en MAYÚSCULAS)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NOMBRE (en MAYÚSCULAS)

--	--	--	--	--	--	--	--	--	--	--	--	--

1.- (2.5 pts) Escribir un programa que lea caracteres de teclado hasta encontrar una letra mayúscula. A partir de ahí, deberá volver a leer caracteres hasta encontrar la siguiente mayúscula, momento en el que acabará el programa imprimiendo como resultado el número de caracteres que han aparecido entre las dos mayúsculas (sin contar éstas). Por ejemplo, si la entrada de datos es:

a\$%t5wrrmasGt67/()|_!hhnmsr*+mqX

el programa mostrará un 19, por ser el número de caracteres entre la G y la X.

```
main()
{ int conta = 0;
  char c;

  printf ("Introduzca caracteres por teclado:\n");
  do
  { scanf("%c", &c);
    } while (c<'A' || c>'Z');
  scanf("%c", &c);
  while (c<'A' || c>'Z')
  { conta++;
    scanf("%c", &c);
  };

  printf ("\nEl número de caracteres entre las dos mayúsculas es %d.\n", conta);
}
```

2.- (2.5 pts) Implementar una función que, pasándole como parámetros dos arrays de N caracteres (considerando la constante N ya definida), cambie por un espacio en blanco aquellos caracteres del primer array que aparezcan en el segundo.

```
void quitar_car (char *cad1, char *cad2)
{ int i, j;

  for (i=0; i<N; i++)
  { for (j=0; j<N && cad1[i]!=cad2[j]; j++);
    if (j<N)
      cad1[i] = ' ';
  }
}
```

3.- (2.5 pts) Escribir una función que, recibiendo como parámetros una matriz de $N \times N$ enteros (considerando la constante N ya definida) y dos enteros f y c (con valores entre 0 y $N-1$ que representan una fila y una columna de la matriz), ponga a 0 todos los elementos de las diagonales que pasan por la posición indicada por f y c , incluida la misma posición. No se permite el uso de matrices auxiliares. Por ejemplo, para una matriz de 5×5 , $f=2$ y $c=1$, las diagonales referidas son las que se muestran en la siguiente matriz:


```
void diagonales (int mat[N][N])
{ int i, j;

  for (i=f, j=c; i<N && j<N; i++, j++) //hacia abajo dcha
    mat[i][j] = 0;
  for (i=f-1, j=c-1; i>=0 && j>=0; i--, j--) //hacia arriba izda
    mat[i][j] = 0;
  for (i=f-1, j=c+1; i>=0 && j<N; i--, j++) //hacia arriba dcha
    mat[i][j] = 0;
  for (i=f+1, j=c-1; i<N && j>=0; i++, j--) //hacia abajo izda
    mat[i][j] = 0;
}
```

4.- (2.5 pts) Implementar un programa principal que haga uso de las dos funciones anteriores, para lo cual tendrá que declarar e inicializar las estructuras de datos adecuadas. Bastará con que llame *adecuadamente* a las funciones, no es necesario comprobar el estado de las mismas antes ni después de las llamadas.

```
#define N 20

main()
{ int matriz[N][N], i, j, fila, col;
  char vector1[N], vector2[N];

  printf("Introduzca los valores enteros de la matriz: \n");
  for (i=0; i<N; i++)
    for (j=0; j<N; j++)
      scanf("%d", &matriz[i][j]);

  printf("Introduzca la fila de la posición de la matriz: \n");
  do
  { scanf("%d", &fila);
  } while (fila<0 || fila >=N);

  printf("Introduzca la columna de la posición de la matriz: \n");
  do
  { scanf("%d", &col);
  } while (col<0 || col >=N);

  diagonales (matriz, fila, col);
```

IDENTIFICADOR
ESTUDIANTE

--	--	--



FUNDAMENTOS DE PROGRAMACIÓN

PRIMER PARCIAL

7 de noviembre de 2022



APELLIDOS (en MAYÚSCULAS)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NOMBRE (en MAYÚSCULAS)

--	--	--	--	--	--	--	--	--	--	--	--	--

```
printf("\n\nIntroduzca los caracteres del primer vector: \n");
for (i=0; i<N; i++)
    scanf("%c", &vector1[i]);

printf("\n\nIntroduzca los caracteres del segundo vector: \n");
for (i=0; i<N; i++)
    scanf("%c", &vector2[i]);

quitar_car (vector1, vector2);
}
```