

IDENTIFICADOR  
ESTUDIANTE

--	--	--



**FUNDAMENTOS DE PROGRAMACIÓN**  
**EXAMEN FINAL SEGUNDO PARCIAL**  
**17 ENERO 2022**



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 **15 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 **1:30 h**. No se podrá salir de los bucles usando **break o return**.

1.- (2 pts) Escribir un programa que lea una palabra por teclado y, a continuación, lea líneas de un fichero de texto de nombre *docu1.txt* y grabe en un segundo fichero, de nombre *docu2.txt*, aquellas líneas del primero que contengan la palabra leída del teclado.

```
#include <stdio.h>
#include <string.h>

int main()
{ FILE *f_in, *f_out;
  char palabra[15], linea[255];

  if (! (f_in = fopen("docu1.txt", "r")))
  { printf ("Error en la apertura del fichero.\n");
    return (1);
  }
  if (! (f_out = fopen("docu2.txt", "w")))
  { printf ("Error en la apertura del fichero.\n");
    return (2);
  }

  printf("Introduzca la palabra a buscar en el fichero: ");
  scanf("%s", palabra);

  while (fgets(linea, 255, f_in))
  if (strstr(linea, palabra))
    fputs(linea, f_out);

  fclose (f_in);
  fclose (f_out);
  return 0;
}
```

2.- a) (0.5 pts) Definir los tipos necesarios para implementar una lista dinámica en la que en cada nodo se almacenarán un número entero y otro real.

```
typedef struct nodo
{ int suma;
  float media;
  struct nodo * sig;
} NODO;

typedef NODO * LISTA;
```

**b) (1.5 pts)** Escribir una función que, recibiendo como parámetros un fichero binario abierto que contiene matrices de NxN enteros (siendo N una cte ya definida) y una lista vacía como la definida en el primer apartado, inserte en ella, por cada matriz contenida en el fichero, un nodo con la suma y la media de los elementos de la matriz.

```
void sum_media (FILE *f, LISTA *l)
{ int matriz[N][N], suma, i, j;
  LISTA aux;

  while (fread(matriz, sizeof(matriz), 1, f))
  { for (i=0, suma=0; i<N; i++)
    for (j=0; j<N; j++)
      suma+=matriz[i][j];
    aux = (LISTA) malloc (sizeof(NODO));
    aux->suma = suma;
    aux->media = (float) suma / (N*N);
    aux->sig = *l;
    *l = aux;
  }
}
```

**3.- (2 pts)** Escribir una función que, pasándole como parámetro una matriz de N cadenas de caracteres (siendo N una cte ya definida), encripte el contenido de la misma cambiando en cada carácter los bits de mayor y menor peso. Por ejemplo, si la representación de un carácter es 10001110, al encriptarlo quedará 00001111.

```
void encriptar (char *matriz[N])
{ int i, j;
  char masc = 128 | 1;

  for (i=0; i<N; i++)
    for (j=0; matriz[i][j]; j++)
      matriz[i][j] ^= masc;
}
```

**4.- (2 pts)** Escribir un programa que lea de teclado un número entero  $n$  entre 2 y 6 que va a determinar el número de secuencias crecientes a leer. A continuación leerá números enteros hasta que se hayan leído  $n$  secuencias crecientes. Por ejemplo, si el número  $n$  es 3, una posible entrada de datos sería:

1 2 8    4 5 7 19    -6 0 8 100    27

```
main()
{ int n, ant, act;

  do
  { printf("Introduzca el número de secuencias crecientes a leer (2..6): ");
    scanf("%d", &n);
  } while (n<2 || n>6);

  printf("\n\nIntroduzca la secuencia de números enteros:\n");
  scanf("%d", &ant);

  do
  { scanf("%d", &act);
    if (ant>act)
      n--;
    ant=act;
  } while (n);
}
```

IDENTIFICADOR  
ESTUDIANTE

--	--	--



**FUNDAMENTOS DE PROGRAMACIÓN**  
**EXAMEN FINAL SEGUNDO PARCIAL**  
**17 ENERO 2022**



APELLIDOS (en MAYÚSCULAS)

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

NOMBRE (en MAYÚSCULAS)

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

5.- (2 pts) Escribir una función que, recibiendo como parámetros una matriz de  $N \times N$  enteros (siendo  $N$  una cte ya definida) y dos enteros  $f$  y  $c$  que representan una fila y columna dentro de la matriz, ponga a 0 todas las posiciones externas de la submatriz que comienza en la posición pasada como parámetro. No se podrá usar otra matriz auxiliar. Por ejemplo, si  $f=1$  y  $c=2$ ,  $N=5$ , la matriz inicial y la resultante son:

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

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

```
void recuadrar (int mat[N][N], unsigned f, unsigned c)
{ int i;

  for (i=f; i<N; i++)
    mat[i][c]=0;

  for (i=f; i<N; i++)
    mat[i][N-1]=0;

  for (i=c+1; i<N-1; i++)
    mat[f][i]=0;

  for (i=c+1; i<N-1; i++)
    mat[N-1][i]=0;
}
```