

--	--	--

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.- (1.5 ptos) Escribir un programa en lenguaje C que lea caracteres por teclado (al menos 3) hasta que un carácter aparezca tres veces seguidas. El programa deberá imprimir un mensaje por pantalla indicando cuál ha sido este carácter.

```
main ()
{ char c1, c2, c3;

printf ("Introduzca caracteres por teclado, al menos 3:\n");
scanf ("%c%c%c", &c1, &c2, &c3);

while (c1!=c2 || c2!=c3)
{ c1 = c2;
  c2 = c3;
  scanf ("%c", &c3);
}
printf ("\nEl carácter repetido tres veces es: %c\n", c3);
}
```

2.- (2.0 ptos) Dada la definición siguiente de la estructura de datos que representa a un alumno:

```
typedef struct
{ char nombre[20], apell[50], matric[7];
} ALUMNO;
```

Se pide escribir una función en lenguaje C que, recibiendo como parámetro una estructura de este tipo, la encripte modificando todos los caracteres que formen parte del *nombre*, *apell* y *matric*, para lo que cambiará cada bit por el contrario.

```
void encriptar (ALUMNO *alum)
{ int i;

for (i=0; alum->nombre[i]; i++)
  alum->nombre[i] ^= ~0;
for (i=0; alum->apell[i]; i++)
  alum->apell[i] ^= ~0;
for (i=0; alum->matric[i]; i++)
  alum->matric[i] ^= ~0;
}
```

3.- (2.5 ptos) Dado el fichero “*nombres.txt*”, formado por un conjunto indefinido de líneas, cada una de las cuales contiene el nombre de un fichero de texto, escribir un programa en lenguaje C que, para cada fichero cuyo nombre está incluido en el fichero “*nombres.txt*”, imprima por pantalla el nombre y número de palabras que contiene. En el caso de que un fichero no se consiga abrir se deberá indicar, también por pantalla, esta situación.

```
int main ()
{ FILE *f_in, *f_aux;
  char palabra[20], nombre[50];
  unsigned conta;

  if (! (f_in = fopen ("nombres.txt", "r")))
  { printf ("Error en la apertura del fichero nombres.txt\n");
    return (1);
  }
  while ( fgets (nombre, 50, f_in))
  if (! (f_aux = fopen (nombre, "r")))
    printf ("\nError en la apertura del fichero %s.\n", nombre);
  else
  { conta = 0;
    while (fscanf(f_aux, "%s", palabra) != EOF)
      conta++;
    fclose (f_aux);
    printf ("El fichero %s contiene %d palabras.\n", nombre, conta);
  }
  fclose (f_in);
  return (0);
}
```

4.- a) (2.0 ptos) Escribir una función en lenguaje C que, recibiendo como parámetro una matriz de 9x9 enteros en el rango de valores 1..9, compruebe que en todas las líneas de la matriz aparecen todos los dígitos del 1 al 9 sin repetirse ninguno (está correcta). Deberá devolver como resultado la fila en la que se encuentre el error y, en caso de que todas las filas estén correctamente rellenas, devolverá un 10.

```
int comprobar_filas (unsigned matr[9][9])
{ unsigned aux[9], i, j, repes=0; //aux es un array de contadores para los dígitos que aparecen en una fila

  for (i=0; i<9 && !repes; ) //se van pasando las filas hasta que aparezca una con repetidos
  { for (j=0; j<9; j++) //se ponen a 0 los contadores de cada dígito
    aux[j] = 0;
    for (j=0; j<9; j++)
      aux[matr[i][j]-1]++; //cada dígito de la fila aumenta su contador
    for (j=0; j<9 && aux[j]; j++); //se recorre el array de contadores hasta el final o hasta que aparezca un 0
    if (!(repes = (j<9))) //si la j no ha recorrido todo el bucle es porque hay algún contador a 0 y, por tanto, repetidos
      i++; //si no hay repetidos se pasa a evaluar la línea siguiente
  }
  return ( i<9 ? i : 10); //si no se han recorrido todas las filas se devuelve la actual, que contiene el error
}
```

--	--	--

APELLIDOS (en MAYÚSCULAS)

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

NOMBRE (en MAYÚSCULAS)

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

4.- b) (2.0 ptos) Escribir un programa en lenguaje C que, recibiendo como parámetro el nombre de un fichero que contiene matrices como las explicadas en el apartado anterior, lea cada una de las matrices del fichero y, para aquellas que contengan alguna fila errónea, muestre por pantalla los valores de ésta. Se debe hacer uso de la función anterior.

```
int main (int argc, char *argv[])
{ FILE *f_in,
  unsigned matriz[9][9], fila, j;

  if (argc<2)
  { printf ("Error en el número del parámetros, falta el nombre del fichero.\n");
    return (1);
  }
  if (! (f_in = fopen (argv[1], "rb")))
  { printf ("Error en la apertura del fichero %s.\n", argv[1]);
    return (2);
  }

  while (fread(matriz, sizeof(matriz), 1, f_in))
  { fila = comprobar_filas (matriz);
    if (fila<10)
    { printf ("\nLa matriz es errónea, tiene elementos repetidos en la fila %d.\n", fila);
      for (j=0; j<9; j++)
        printf ("%3d", matriz[fila][j]);
      printf ("\n");
    }
  }
  fclose (f_in);
  return (0);
}
```