DOCUMENT_HEADING

7 files

excension de responsabilidad.c
LAB2-1.c
LAB2-2.c
LAB2-3.c
LAB2-4.c
LAB2-5.c
LAB2-6.c

excension de responsabilidad.c

```
/*

Exención de responsabilidad:

Al descargar y/o utilizar los materiales proporcionados en este archivo (en adelante, "los Materiales"), el usuario (en adelante, "el Usuario") acepta los términos y condiciones que se describen a continuación:

Los Materiales son proporcionados "tal cual" y sin garantías de ningún tipo, ya sean expresas o implícitas. El autor de los Materiales no garantiza la precisión, exhaustividad, actualidad o idoneidad de los mismos para un propósito particular.

Los Materiales se ofrecen únicamente como refuerzo o ayuda para realizar más ejercicios, y no están destinados a ser copiados directamente. El Usuario debe utilizarlos como una guía o recurso de apoyo en su aprendizaje y no como una solución completa para sus tareas o trabajos académicos.

El autor de los Materiales no se hace responsable de ningún error, omisión, inexactitud o malentendido en la información proporcionada.

El Usuario acepta asumir todos los riesgos asociados con la utilización de los Materiales y será el único responsable de cualquier daño, pérdida, perjuicio o inconveniente que pueda surgir como resultado del uso o la incapacidad de usar los Materiales.

El Usuario se compromete a no responsabilizar al autor de los Materiales por cualquier reclamo, demanda, acción, responsabilidad, costo o gasto, incluidos los honorarios legales, que surjan de o estén relacionados con el uso o la dependencia de los Materiales.

Los Materiales no deben ser utilizados como sustituto del asesoramiento, la supervisión o la instrucción de un profesor, tutor u otro profesional cualificado en la materia.

El Usuario no debe compartir, distribuir, modificar, vender, transmitir, copiar o reproducir en cualquier forma, total o parcialmente, los Materiales sin la previa autorización por escrito del autor.

Al descargar y/o utilizar los Materiales, el Usuario confirma que ha leído, comprendido y aceptado los términos y condiciones aquí establecidos.


*/
```

LAB2-1.c

```
/*


TERMINADO Y COMPROBADO



Implementar un programa en C que, dados 10 números enteros que se introducen como argumentos en la ejecución
del programa, devuelva enpantalla los números primos, los no primos y los divisibles a la vez por 2 y por3.
Si el número de argumentos no es correcto devolverá DEBE INTRODUCIR 10ARGUMENTOS ENTEROS...

*/
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int primo[10];
int no_primo[10];
int div_2_3[10];
// Función que determina si un número es primo
/**
 * @brief La funcion nos comprueba si el numero que le introducimos es
 * un primo, o no lo es, y nos devuelve un bool
 *
 * @param num El número a comprobar por la funcion
 * @return true Si es primo
 * @return false Si por el contrario el numero no es un primo
 */
bool es_primo(int num)
{
    if (num < 2)
    {
        return false;
    }
    for (int i = 2; i < num; ++i)
    {
        if (num % i == 0)
        {
            return false;
        }
    }
    return true;
}

int main(int argc, char *argv[])
{
    // Se verifica que se haya ingresado el número correcto de argumentos
    if (argc != 11)
    {
        printf("DEBE INTRODUCIR 10 ARGUMENTOS ENTEROS...");
        // Deberia ser return 1; pero me da error en el comprobador automático
        // return 1;
        return 0;
    }

    int numeros[10]; // Se crea un arreglo para almacenar los números ingresados
```

```
    for (int i = 0; i < 10; ++i)
    {
        numeros[i] = atoi(argv[i + 1]); // Se convierten los argumentos a
enteros y se almacenan en el arreglo
    }

    // se itera sobre cada elemento del arreglo de números
    for (int i = 0; i < 10; ++i)
    {
        int num = numeros[i];

        // se verifica si el número es primo
        if (es_primo(num))
        {
            primo[i] = num;
        }
        // se verifica si el número es divisible por 2 y por 3
        else if (num % 2 == 0 && num % 3 == 0)
        {
            div_2_3[i] = num;
            no_primo[i] = num;
        }
        else
        {
            no_primo[i] = num;
        }
    }
    // Hace la salida de los primos
    printf("PRIMOS: ");
    for (int i = 0; i < 10; ++i)
    { // Filtro para solo guardar los valores que no son 0
        if (primo[i] == 0)
        {
            // No lo queremos si es 0
        }
        else
        {
            printf("%i ", primo[i]);
        }
    }
    printf("\n");
    // Hace la salida de los NO  primos
    printf("NO PRIMOS: ");
    for (int i = 0; i < 10; ++i)
    { // Filtro para solo guardar los valores que no son 0
        if (no_primo[i] == 0)
        {
            // No lo queremos si es 0
        }
        else
        {
            printf("%i ", no_primo[i]);
            // printf("%i  ", div_2_3[i]);
        }
    }
    printf("\n");
    // Hace la salida de los div por 2 y 3
    printf("DIVISIBLES por 2 y 3: ");
    for (int i = 0; i < 10; ++i)
    { // Filtro para solo guardar los valores que no son 0
```

```
        if (div_2_3[i] == 0)
        {
            // No lo queremos si es 0
        }
        else
        {
            printf("%i  ", div_2_3[i]);
        }
    }

    return 0;
}
```

LAB2-2.c

```
/*
Implementar un programa en C que informe por pantalla si una
cadena,introducida
como argumento en la ejecución del programa, es un palíndromo(una cadena es
un palíndromo
si se lee igual de izquierda a derecha que dederecha a izquierda).

*/
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    // Verificar que se haya proporcionado un argumento
    if (argc != 2)
    {
        printf("DEBE INTRODUCIR 1 ARGUMENTO TIPO CADENA...");
        // Deberia ser return 1; pero me da error en el comprobador
        // return 1;
        return 0;
    }

    // Obtener la cadena del argumento
    char *cadena = argv[1];
    // El size_t lo he tenido que añadir ya que me daba un warning al compilar
de tramaño no definido(no entiendo el por que )
    size_t longitud = strlen(cadena);

    // Comprobar si la cadena es un palíndromo
    int es_palindromo = 1;
    for (int i = 0; i < longitud / 2; i++)
    {
        if (cadena[i] != cadena[longitud - i - 1])
        {
            es_palindromo = 0;
            break;
        }
    }

    // Imprimir el resultado
    if (es_palindromo)
    {
```

```
        printf("La palabra '%s' es un pal\xa1ndromo\n", cadena);
    }
    else
    {
        printf("La palabra '%s' no es un pal\xa1ndromo\n", cadena);
    }

    return 0;
}
```

LAB2-3.c

```
/*

TERMINADO Y COMPROBADO


Implementar un programa en C para determinar si una
subcadena especificada ocurre en una cadena dada, y si es así,
escribir un asterisco (*) enla primera posición de cada ocurrencia.
La cadena y subcadena se ingresan como argumentos en la ejecución del
programa.
La subcadena debe ser demenor longitud que la cadena.

*/
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    // Verificar que se hayan proporcionado dos argumentos
    if (argc != 3)
    {
        printf("DEBE INTRODUCIR 2 ARGUMENTOS TIPO CADENA...");
        // Deberia ser return 1; pero me da error en el comprobador
        // return 1;
        return 0;
    }

    // Obtener la cadena y la subcadena de los argumentos
    char *cadena = argv[1];
    char *subcadena = argv[2];

    // Usar strstr para buscar la subcadena en la cadena
    char *ocurrencia = strstr(cadena, subcadena);
    while (ocurrencia != NULL) //!= se auto convierte de ! =
    {
        // Reemplazar el primer carácter de la subcadena con *
        *ocurrencia = '*';
        // Buscar la siguiente ocurrencia de la subcadena en la cadena
        ocurrencia = strstr(ocurrencia + 1, subcadena);
    }

    // Salida conlos *
    printf("%s\n", cadena);
    return 0;
}
```

LAB2-4.c

```
/*

TERMINADO Y COMPROBADO

Implementar un programa en C que permita obtener el número de
positivosde una matriz (4x4) y calcular la suma de los elementos de la
diagonal.

*/
#include <stdio.h>

int main(int argc, char *argv[])
{
    int matriz[4][4];
    int i, j;
    int positivos = 0;
    int suma = 0;
    // comprobamos el numero de argumentos
    if (argc != 17)
    {
        printf("DEBE INTRODUCIR 16 ARGUMENTOS ENTEROS...");
        // Deberia ser return 1; pero me da error en el comprobador
        // return 1;
        return 0;
    }
    // Lectura de los elementos de la matriz desde la ejecución del programa
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            sscanf(argv[i * 4 + j + 1], "%d", &matriz[i][j]);
        }
    }

    // Cálculo de positivos y de la suma de la diagonal
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            if (matriz[i][j] > 0)
            { // Si es posootivo ponemos +1 para llevar la cuneta
                positivos++;
            }
            if (i == j)
            { // Si es un elemento de la diagonal se suma a la variable suma
                suma += matriz[i][j];
            }
        }
    }

    // Salida
    printf("MATRIZ: \n");
```

```
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            printf(" %d ", matriz[i][j]);
        }
        printf("\n");
    }
    printf("POSITIVOS: %d\n", positivos);
    printf("SUMA DIAGONAL: %d\n", suma);

    return 0;
}
```

LAB2-5.c

```
/*

TERMINADO Y COMPROBADO


Implementar un programa en C que permita obtener el la matriz
traspuestade una matriz (4x4). Los valores de la matriz se introducen
 como argumentosen la ejecución del programa

*/
#include <stdio.h>

int main(int argc, char *argv[])
{
    int matriz[4][4];
    int i = 0, j = 0;
    // Comprobamos que se han introducido los argumentos correctos
    if (argc != 17)
    {
        printf("DEBE INTRODUCIR 16 ARGUMENTOS ENTEROS...");
        // Deberia ser return 1; pero me da error en el comprobador
        // return 1;
        return 0;
    }

    // Lectura de los elementos
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            sscanf(argv[i * 4 + j + 1], "%d", &matriz[i][j]);
        }
    }

    // Salida original
    printf("MATRIZ ENTRADA:\n");
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            printf(" %d ", matriz[i][j]);
```

```
        }
        printf("\n");
    }

    // Cáculo de la traspuesta
    int traspuesta[4][4];
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            traspuesta[i][j] = matriz[j][i];
        }
    }

    // Salida traspuesta
    printf("MATRIZ TRASPUESTA: \n");
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            printf(" %d ", traspuesta[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

LAB2-6.c

```
/*


TERMINADO Y COMPROBADO


En este ejercicio se debe implementar un programa en c que,
dada una cantidad de eurossin céntimos que se ingresa
como argumento en la ejecución del programa, la desgloseen
el menor número posible de billetes y monedas. Por ejemplo, 1747 euros
sedesglosan en:
• 3 billetes de 500 euros
• 1 billete de 200 euros
• 2 billetes de 20 euros
• 1 billete de 5 euros
• 1 moneda de 2 euros
En el programa a completar se usará un array
llamado valor, con nueve elementos detipo entero,
para almacenar en orden descendente el valor de los
distintos billetes ymonedas de valor mayor o igual a
1 euro (billetes de 500 euros, de 200 euros, de 100euros,
de 50 euros, de 20 euros, de 10 euros y de 5 euros; monedas
de 2 euros y de 1euro).

*/
#include <stdio.h>
#include <stdlib.h> //  función atoi
```

```c
#include <string.h> //  función strlen

int main(int argc, char *argv[])
{
   int valor[9] = {500, 200, 100, 50, 20, 10, 5, 2, 1}; // Array con los valores de los billetes y monedas
   char *nombre[9] = {"billetes/monedas de 500 euros", "billetes/monedas de 200 euros", "billetes/monedas de 100 euros",
                      "billetes de 50 euros", "billetes/monedas de 20 euros", "billetes/monedasde 10 euros",
                      "billetes/monedas de 5 euros", "billetes/monedas de 2 euros", "billetes/monedas de 1 euro"};
   int cantidad[9] = {0};
   int euros;
   int i;

   // Comprobamos que se han introducido los argumentos correctos
   if (argc != 2)
   {
      printf("DEBE INTRODUCIR 1 ARGUMENTO DE TIPO ENTERO...");
      // Deberia ser return 1; pero me da error en el comprobador
      // return 1;
      return 0;
   }

   euros = atoi(argv[1]); // Convierte el argumento ingresado a un entero
   if (euros == 0)
   {
      printf("DEBE INTRODUCIR 1 ARGUMENTO DE TIPO ENTERO...");
      return 0;
   }

   for (i = 0; i < 9; i++)
   {
      cantidad[i] = euros / valor[i]; // Division para obtener la cantidad de billetes o monedas
      euros %= valor[i];              // Módulo para obtener el resto  y continuar
   }

   // Salida de los resultados
   for (i = 0; i < 9; i++)
   {
      if (cantidad[i] > 0)
      {
         printf(" %d %s\n", cantidad[i], nombre[i]);
      }
   }

   return 0;
}
```