

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatización

Asignatura: Fundamentos de Programación

Tema del taller: Resolución de Ejercicios para Codeblock

Docente: Ing. Jenny Ruiz

Integrantes: Mendoza Cortez Christopher Alexander

Fecha: 17/11/2025 Paralelo: 29583

2. DESARROLLO

Transcribir los codigos que estan escritos en pseudocodigo a codigo en C para poder ejecutarlo en Codeblock.

U1T1

Ejercicio 1 (Media de dos valores)

Desarrolle un programa que lea dos números reales desde teclado e imprima su media. los requisitos funcionales son: primero se debe leer un valor real y se guarda en una variable X; a continuación, se lee y almacena el Segundo valor en Y, Finalmente la variable RES recibe la media de ambos valores y los muestra por pantalla.

Codigo

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

```
int main() {
```

```
    // Declaración de variables
```

```
    double X, Y, RES;
```

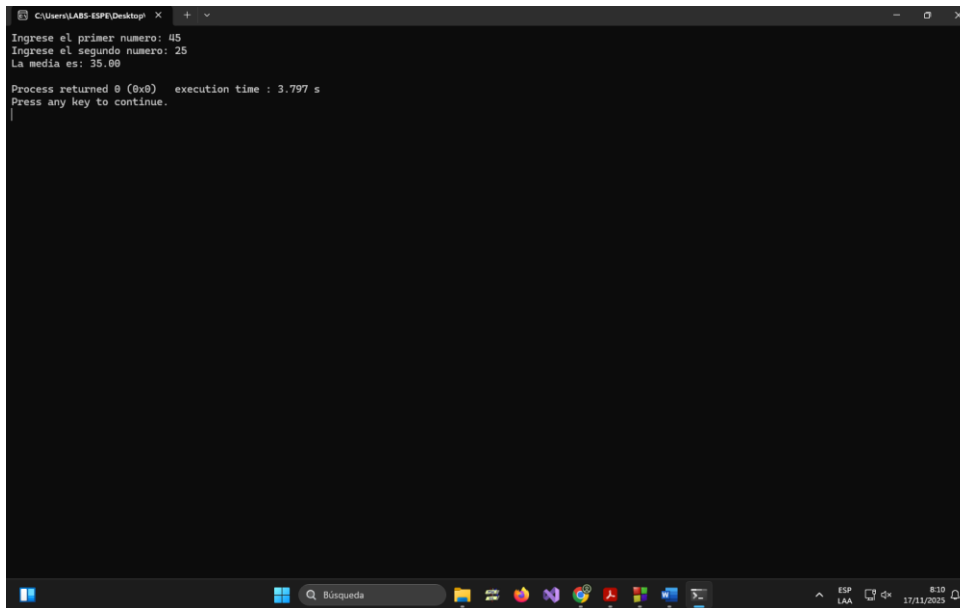
```
    // Lectura de datos
```

```
    printf("Ingrese el primer numero: ");
```

```
    scanf("%lf", &X);
```

```
printf("Ingrese el segundo numero: ");  
  
scanf("%lf", &Y);  
  
// Cálculo de la media  
  
RES = (X + Y) / 2;  
  
// Mostrar resultado  
  
printf("La media es: %.2lf\n", RES);  
  
return 0;  
}
```

Prueba de Escritorio



```
C:\Users\LABS-ESPE\Desktop  
Ingrese el primer numero: 45  
Ingrese el segundo numero: 25  
La media es: 35.00  
Process returned 0 (0x0)   execution time : 3.797 s  
Press any key to continue.
```

Ejercicio 2 (Valor absolute de X al cubo)

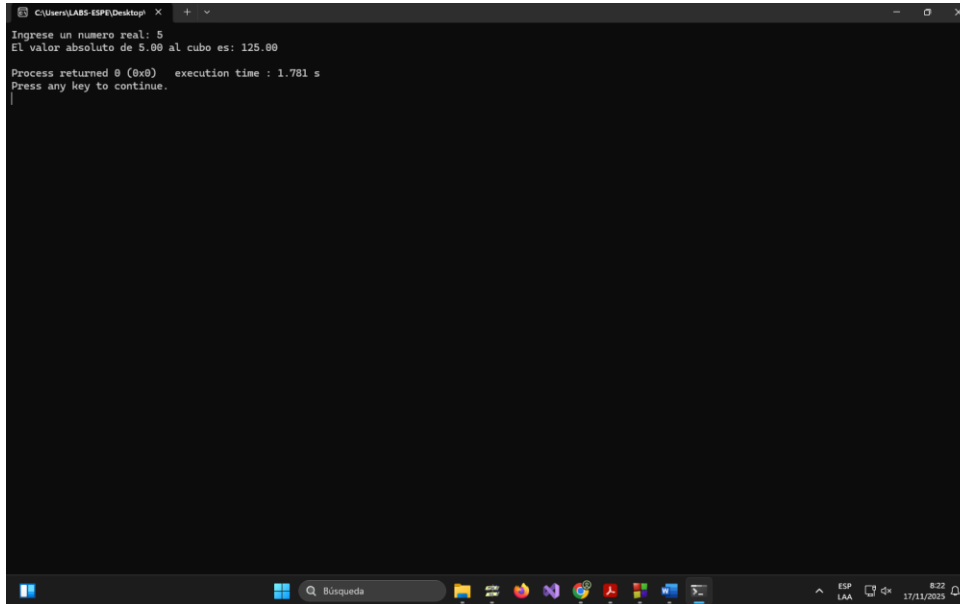
Desarrolle un programa que lea un número real X y escribe por pantalla $|X|^3$

Codigo

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

```
int main() {  
  
    // Declaración  
  
    double X, Va, R;  
  
  
    // Desarrollo  
  
    printf("Ingrese un numero real: ");  
    scanf("%lf", &X);  
  
  
    // Cálculos  
  
    if (X < 0) {  
        Va = -X; // Valor absoluto manual  
    } else {  
        Va = X;  
    }  
  
  
    R = pow(Va, 3); // Elevar al cubo  
  
  
    // Resultados  
  
    printf("El valor absoluto de %.2lf al cubo es: %.2lf\n", X, R);  
  
  
    return 0;  
}
```

Prueba de Escritorio



```
C:\Users\LABS-ESPE\Desktop>
Ingrese un numero real: 5
El valor absoluto de 5.00 al cubo es: 125.00

Process returned 0 (0x0)   execution time : 1.781 s
Press any key to continue.
```

Ejercicio 3 (Divisible)

Desarrolle un programa que lea dos números enteros por teclado y determine si el primero de ellos es divisible por el Segundo, Se mostrará en pantalla el resultado. Utilice el operador modulo % que devuelve el resto de la división.

Codigo

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

```
int main() {
```

```
    // Declaración de variables
```

```
    int N, D, R;
```

```
    // Desarrollo
```

```
    printf("Ingrese el primer numero: ");
```

```
    scanf("%d", &N);
```

```
    printf("Ingrese el segundo numero: ");
```

```
    scanf("%d", &D);
```

```
// Cálculos

R = N % D;

// Resultados

if (R == 0) {

    printf("El numero %d es divisible por %d\n", N, D);

} else {

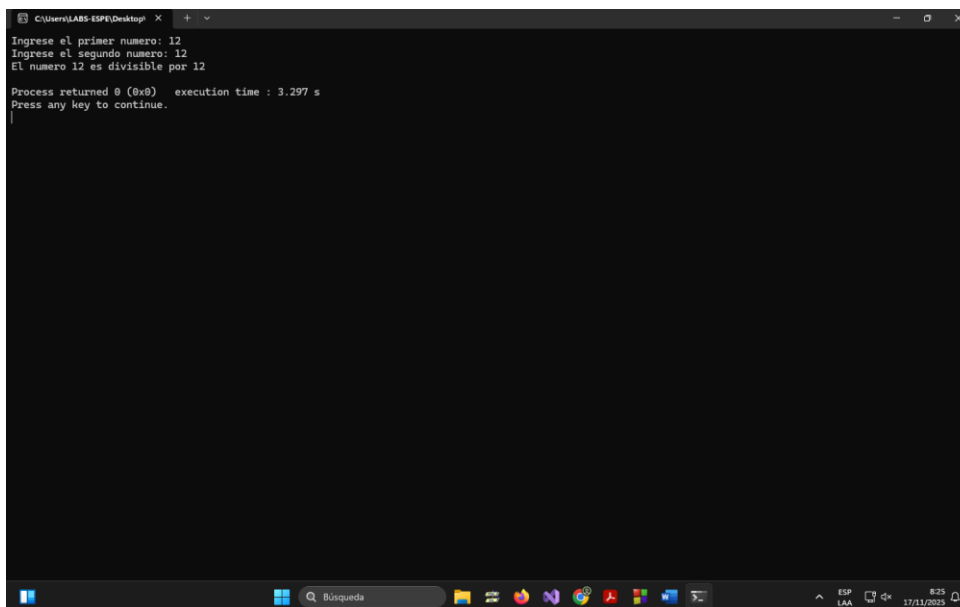
    printf("El numero %d NO es divisible por %d\n", N, D);

}

return 0;

}
```

Prueba de Escritorio



```
C:\Users\LABS-ESPE\Desktop>
Ingrese el primer numero: 12
Ingrese el segundo numero: 12
El numero 12 es divisible por 12

Process returned 0 (0x0)   execution time : 3.297 s
Press any key to continue.
```

Ejercicio 4 (Intervalo)

Desarrolle un programa que lea un número real del teclado y determine si pertenece al intervalo de (0 a10), indicando por pantalla el resultado.

El número X introducido ha de cumplir $X > 0$ y $X \leq 10$ (ambas condiciones a la vez) para pertenecer al intervalo

Código

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

```
int main() {
```

```
    // Declaración
```

```
    double X;
```

```
    int C1, C2, Sol;
```

```
    // Desarrollo
```

```
    printf("Ingrese el numero real: ");
```

```
    scanf("%lf", &X);
```

```
    // Condiciones
```

```
    C1 = (X > 0);
```

```
    C2 = (X <= 10);
```

```
    Sol = C1 && C2;
```

```
    // Resultado
```

```
    if (Sol) {
```

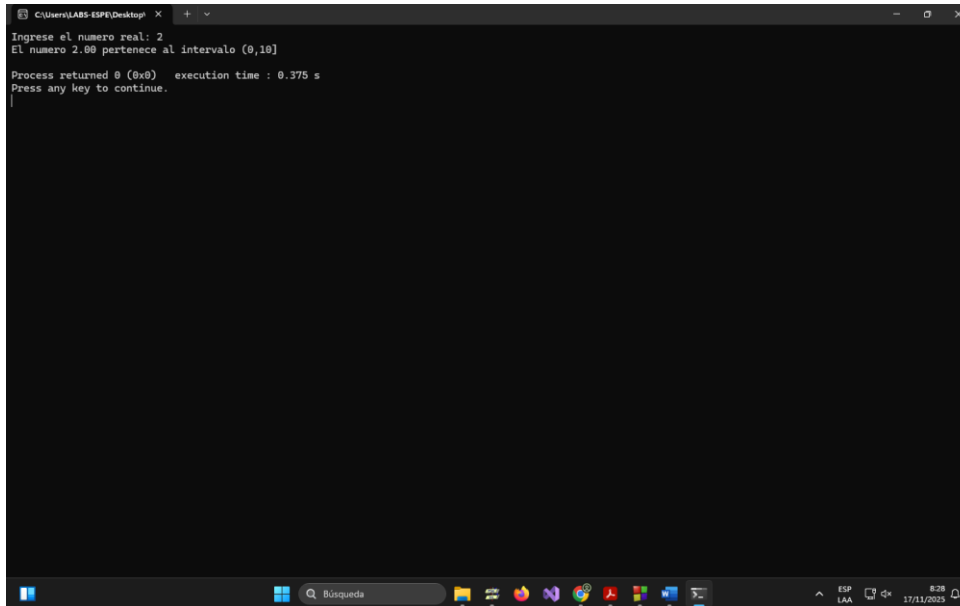
```
        printf("El numero %.2lf pertenece al intervalo (0,10]\n", X);
```

```
    } else {
```

```
        printf("El numero %.2lf NO pertenece al intervalo (0,10]\n", X);
```

```
}  
  
return 0;  
  
}
```

Prueba de Escritorio



Ejercicio 5(conversión de unidades de tiempo)

Desarrolle un programa que lea por teclado un valor entero X e, interpretando este valor como el número de segundos que dura un evento, calcule y muestre por pantalla cuántos días, horas, minutos y segundos representa X.

Codigo

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

```
int main() {  
  
    // Declaración de variables  
  
    int X, D, H, M, S, R;  
  
    // Desarrollo  
  
    printf("Ingrese la cantidad de segundos: ");
```

```
scanf("%d", &X);

// Cálculos

D = X / 86400;    // Dias

R = X % 86400;

H = R / 3600;    // Horas

R = R % 3600;

M = R / 60;      // Minutos

S = R % 60;      // Segundos

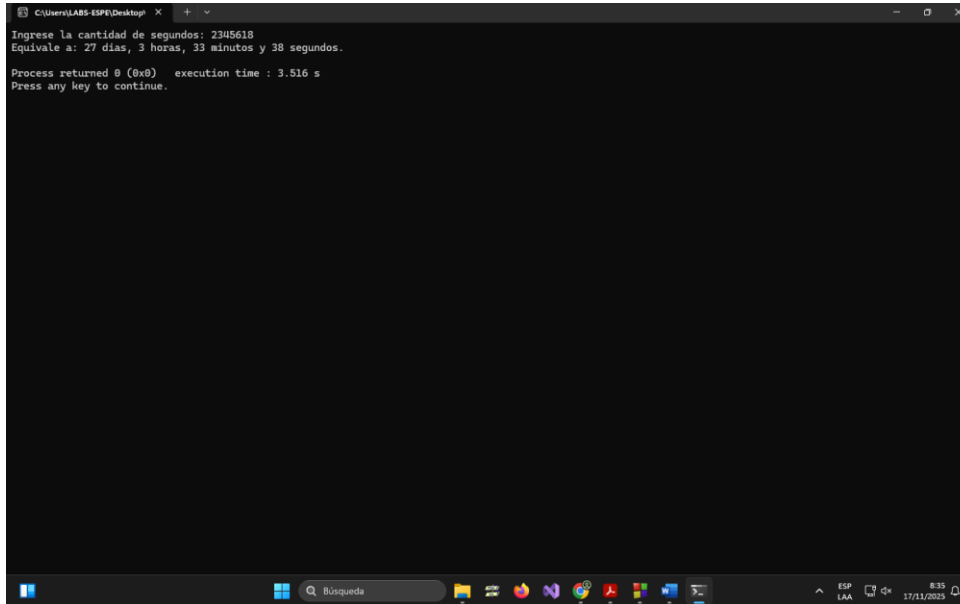
// Resultados

printf("Equivale a: %d dias, %d horas, %d minutos y %d segundos.\n", D, H, M, S);

return 0;

}
```


Prueba de Escritorio



U1T2

Ejercicio 6 Media de números positivos

Desarrolle un programa que lea tres números reales desde teclado y escriba por pantalla la media aritmética únicamente de los números positivos leídos. Puede dar dos opciones A) Empleando una iteración con salida en cabeza y B) Empleando una iteración con salida en cola

Código

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

```
int main() {
```

```
    // Declarar variables
```

```
    double num, sum = 0, media;
```

```
    int cont = 0, x = 1;
```

```
    // Lectura de tres números con iteración
```

```
    do {
```

```
        printf("Ingrese un numero %d: ", x);
```



```
scanf("%lf", &num);

if (num > 0) {

    sum += num;

    cont++;

}

x++;

} while (x <= 3);

// Cálculo y resultado
if (cont > 0) {

    media = sum / cont;

    printf("La media de los numeros positivos es: %.2lf\n", media);

} else {

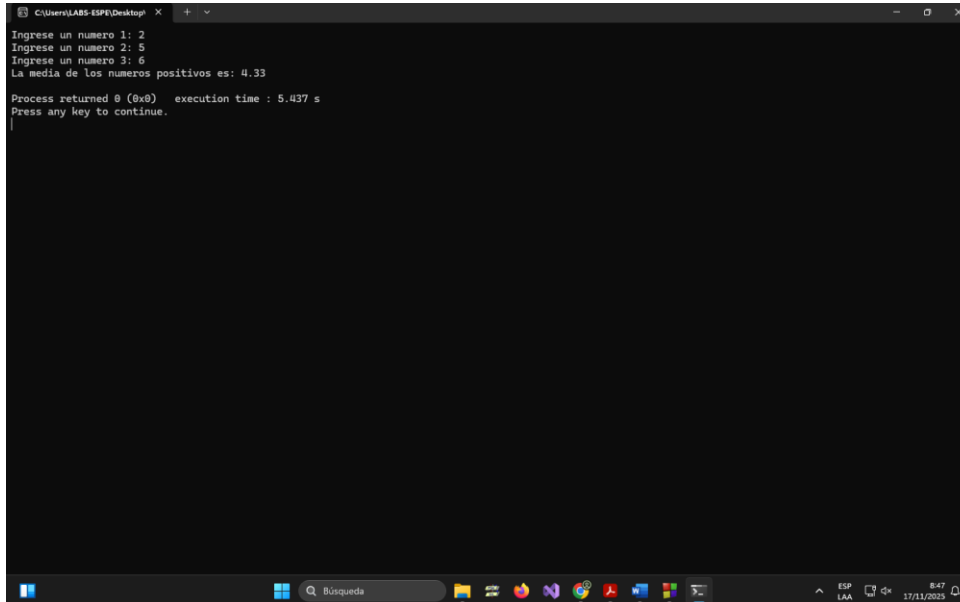
    printf("No se ingresaron numeros positivos.\n");

}

return 0;

}
```

Prueba de Escritorio



```
C:\Users\LAA-ESPE\Desktop>
Ingrese un numero 1: 2
Ingrese un numero 2: 5
Ingrese un numero 3: 6
La media de los numeros positivos es: 4.33

Process returned 0 (0x0)   execution time : 5.437 s
Press any key to continue.
```

Ejercicio 7 Máximo y media de alturas

Desarrolle un programa que lea por teclado un número indeterminado de alturas y que deje de leer cuando se hay introducido una altura negativa. El programa debe calcular e imprimir la media, el máximo y el mínimo de las alturas introducidas

Codigo

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

```
int main() {
```

```
    // Declarar variables
```

```
    double Alt, sum = 0, media, max = -99999, min = 99999;
```

```
    int cont = 0;
```

```
    // Lectura de alturas hasta que se ingrese una negativa
```

```
    do {
```

```
        printf("Ingrese una altura (negativa para terminar): ");
```

```
        scanf("%lf", &Alt);
```



```
if (Alt >= 0) {  
  
    sum += Alt;  
  
    cont++;  
  
    if (Alt > max) {  
  
        max = Alt;  
  
    }  
  
    if (Alt < min) {  
  
        min = Alt;  
  
    }  
  
}  
  
} while (Alt >= 0);  
  
// Cálculo y salida  
  
if (cont > 0) {  
  
    media = sum / cont;  
  
    printf("La media de alturas es: %.2lf\n", media);  
  
    printf("La altura maxima es: %.2lf\n", max);  
  
    printf("La altura minima es: %.2lf\n", min);  
  
} else {  
  
    printf("No se ingresaron alturas válidas.\n");  
  
}  
  
return 0;  
  
}
```



Prueba de Escritorio

```
"C:\Users\Katy Mendoza\Desktop\PROGRAMACION\U1T2-7.exe"
Ingrese una altura (negativa para terminar): 156
Ingrese una altura (negativa para terminar): 166
Ingrese una altura (negativa para terminar): 165
Ingrese una altura (negativa para terminar): -85
La media de alturas es: 162.33
La altura máxima es: 166.00
La altura mínima es: 156.00
Process returned 0 (0x0)   execution time : 7.076 s
Press any key to continue.
```

Ejercicio 8 Raiz Cuadrada Entera

Desarrollo un programa que calcule la raíz cuadrada entera por defecto de un número N positivo dado, que se define como el mayor número entero $R > 0$, tal que R Elevado al cuadrado sea $\leq N$.

Código

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

```
int main() {
```

```
    // Declaración de variables
```

```
    int N, R;
```

```
    // Ingreso de datos
```

```
    printf("Ingrese un numero positivo: ");
```

```
    scanf("%d", &N);
```

```
    // Validación
```

```
    if (N < 0) {
```

```
printf("El numero debe ser positivo.\n");  
  
} else {  
  
    // Inicialización  
  
    R = 1;  
  
  
    // Búsqueda de la raíz cuadrada entera por defecto  
  
    while (R * R <= N) {  
  
        R = R + 1;  
  
    }  
  
  
    R = R - 1;  
  
  
    // Resultado  
  
    printf("La raiz cuadrada entera por defecto de %d es %d\n", N, R);  
  
}  
  
  
return 0;  
}
```



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Prueba de Escritorio

```
"C:\Users\Katy Mendoza\Desktop\PROGRAMACION\UIT2-8.exe"  
Ingrese un numero positivo: 65  
La raíz cuadrada entera por defecto de 65 es 8  
Process returned 0 (0x0)   execution time : 1.661 s  
Press any key to continue.
```

Ejercicio 9 Número primo

Desarrolle un programa que lea por teclado un número entero mayor que 1 y determine si es primo o no

Código

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

```
int main() {
```

```
    // Declaración de variables
```

```
    int N, i, cont;
```

```
    // Ingreso de datos
```

```
    printf("Ingrese un numero entero mayor que 1: ");
```

```
    scanf("%d", &N);
```

```
    // Validación
```

```
    if (N <= 1) {
```

```
printf("El numero debe ser mayor que 1.\n");

} else {

    cont = 0;


    // Verificación de divisores

    for (i = 1; i <= N; i++) {

        if (N % i == 0) {

            cont++;

        }

    }


    // Un numero primo solo tiene 2 divisores: 1 y él mismo

    if (cont == 2) {

        printf("%d es un numero primo.\n", N);

    } else {

        printf("%d no es un numero primo.\n", N);

    }

}

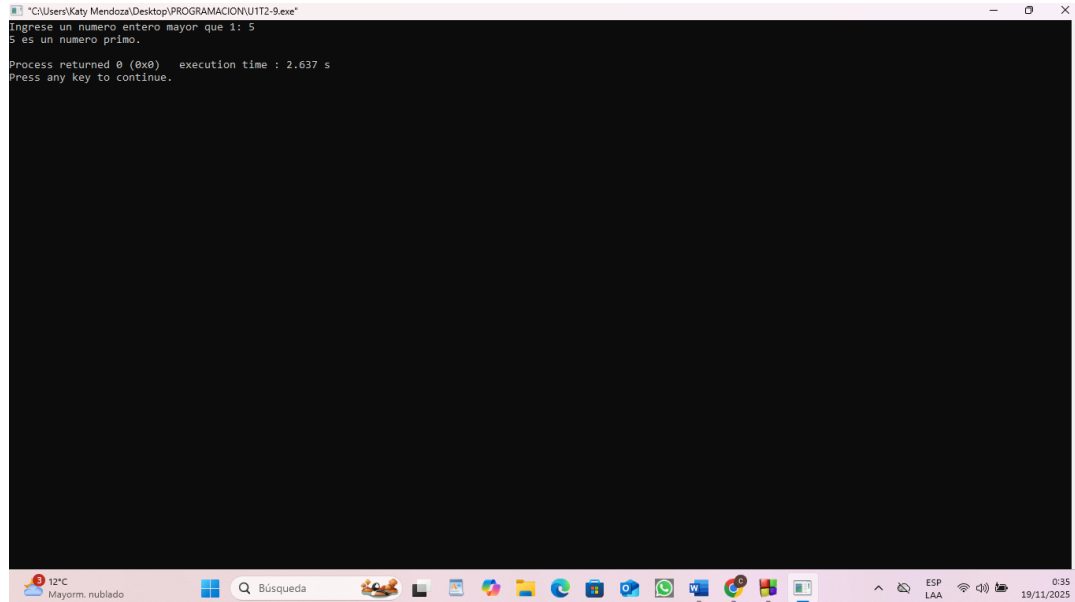

return 0;

}
```




ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Prueba de Escritorio



Ejercicio 10 Números primos comprendidos entre dos números dados

Desarrolle un programa que muestre por pantalla todos los números primos comprendidos entre dos números enteros introducidos por teclado.

Código

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

```
int main() {
```

```
    // Declaración de variables
```

```
    int A, B, i, j, cont;
```

```
    // Ingreso de datos
```

```
    printf("Ingrese el primer numero entero:");
```

```
    scanf("%d", &A);
```

```
    printf("Ingrese el segundo numero entero:");
```

```
    scanf("%d", &B);
```



```
// A tiene que ser menor que B
```

```
if (A > B) {
```

```
    cont = A;
```

```
    A = B;
```

```
    B = cont;
```

```
}
```

```
// Mostrar los numeros primos entre A y B
```

```
printf("Los numeros primos entre %d y %d son:\n", A, B);
```

```
for (i = A; i <= B; i++) {
```

```
    if (i > 1) {
```

```
        cont = 0;
```

```
        for (j = 1; j <= i; j++) {
```

```
            if (i % j == 0) {
```

```
                cont++;
```

```
            }
```

```
        }
```

```
        if (cont == 2) {
```

```
            printf("%d ", i);
```

```
        }
```

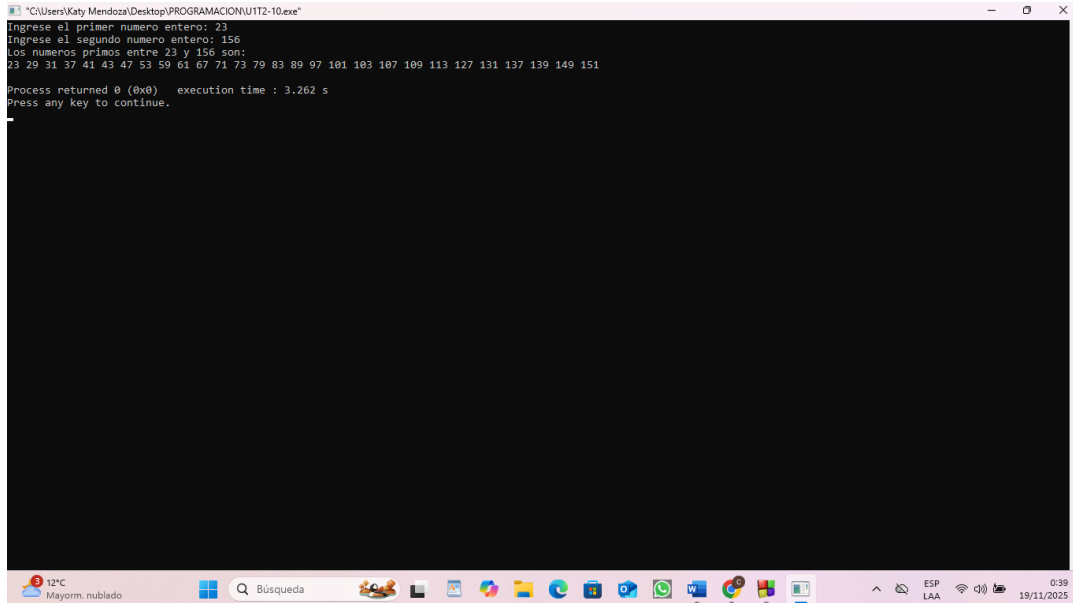
```
    }
```

```
}
```

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

```
return 0;  
}
```

Prueba de Escritorio



```
"C:\Users\Katy Mendoza\Desktop\PROGRAMACION\U1T2-10.exe"  
Ingrese el primer numero entero: 23  
Ingrese el segundo numero entero: 156  
Los numeros primos entre 23 y 156 son:  
23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151  
Process returned 0 (0x0)   execution time : 3.262 s  
Press any key to continue.
```

U1T3

Desarrolle un programa con un MENÚ PRINCIPAL que ofrezca las siguientes opciones:
MENÚ PRINCIPAL

1. Serie Fibonacci
2. Operaciones
 - 2.1 Par o Impar
 - 2.2 Primo
 - 2.3 Retornar
3. Salir

Código

```
#include <stdio.h>  
#include <math.h>  
#include <stdbool.h>  
  
int main() {  
    int opcion, opcion2, n, num, i, a, b, temp;  
    bool primo;  
  
    do {  
        // Menú principal  
        printf("\n===== MENÚ PRINCIPAL =====\n");  
        printf("1. Serie Fibonacci\n");  
        printf("2. Operaciones\n");
```



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

```
printf("3. Salir\n");
printf("Seleccione una opción: ");
scanf("%d", &opcion);

switch (opcion) {
    case 1:
        printf("¿Cuántos términos de la serie Fibonacci quiere
ver?: ");
        scanf("%d", &n);

        if (n > 0) {
            a = 0;
            b = 1;
            printf("Serie Fibonacci:\n");
            for (i = 1; i <= n; i++) {
                printf("%d ", a);
                temp = a + b;
                a = b;
                b = temp;
            }
            printf("\n");
        } else {
            printf("Número no válido. Debe ser positivo.\n");
        }
        break;

    case 2:
        do {
            // Submenú operaciones
            printf("\n--- SUBMENÚ OPERACIONES ---\n");
            printf("1. Verificar Par o Impar\n");
            printf("2. Verificar si es Primo\n");
            printf("3. Regresar al Menú Principal\n");
            printf("Seleccione una opción: ");
            scanf("%d", &opcion2);

            switch (opcion2) {
                case 1:
                    printf("Ingrese un número positivo: ");
                    scanf("%d", &num);
                    if (num > 0) {
                        if (num % 2 == 0)
                            printf("El número %d es PAR.\n",
num);
                        else
                            printf("El número %d es IMPAR.\n",
num);
                    } else {
                        printf("Número no válido. Debe ser
positivo.\n");
                    }
                    break;

                case 2:
                    printf("Ingrese un número positivo: ");
                    scanf("%d", &num);
```



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

```
if (num > 0) {
    if (num == 1) {
        printf("El número NO es primo.\n");
    } else {
        primo = true;
        for (i = 2; i <= (int)sqrt(num);
i++) {
            if (num % i == 0) {
                primo = false;
                break;
            }
        }
        if (primo)
            printf("El número %d es
PRIMO.\n", num);
        else
            printf("El número %d NO es
primo.\n", num);
    }
} else {
    printf("Número no válido. Debe ser
positivo.\n");
}
break;

case 3:
    printf("Regresando al MENÚ
PRINCIPAL...\n");
    break;

default:
    printf("Opción no válida. Intente
nuevamente.\n");
}
} while (opcion2 != 3);
break;

case 3:
    printf("Finalizando el programa... ;Hasta la
próxima!\n");
    break;

default:
    printf("Opción no válida. Intente nuevamente.\n");
}
} while (opcion != 3);

return 0;
}
```



Prueba de Escritorio

```
"C:\Users\Katy Mendoza\Desktop\PROGRAMACION\U1T3.exe"

===== MENU PRINCIPAL =====
1. Serie Fibonacci
2. Operaciones
3. Salir
Seleccione una opcion: 1
¿Cuántos terminos de la serie Fibonacci quiere ver?: 2
Serie Fibonacci:
0 1

===== MENU PRINCIPAL =====
1. Serie Fibonacci
2. Operaciones
3. Salir
Seleccione una opcion: 2

--- SUBMENU OPERACIONES ---
1. Verificar Par o Impar
2. Verificar si es Primo
3. Regresar al Menu Principal
Seleccione una opcion: 1
Ingrese un numero positivo: 35
El numero 35 es IMPAR.

--- SUBMENU OPERACIONES ---
1. Verificar Par o Impar
2. Verificar si es Primo
3. Regresar al Menu Principal
Seleccione una opcion:
```

U1EVAL1CAP1-2

Desarrollar un algoritmo que ingrese un número entero por teclado e identifique si este número es par o impar, y que imprima por pantalla el número es.

Código

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

```
int main() {
```

```
    // Declaración de variables
```

```
    int X, RES;
```

```
    int C = 2;
```

```
    // Entrada de datos
```

```
    printf("Ingrese un valor entero: ");
```

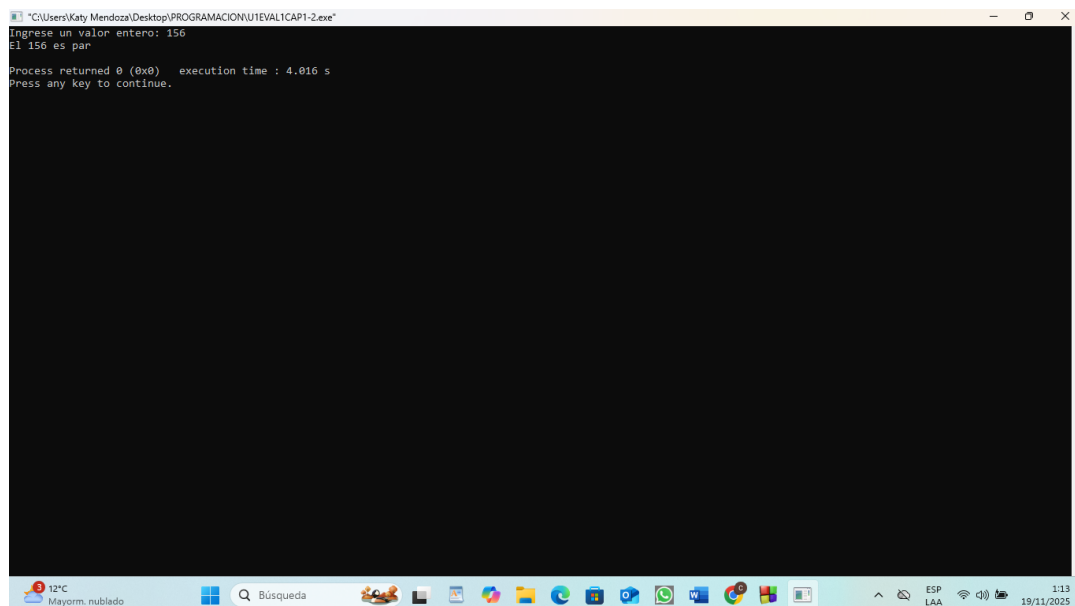
```
    scanf("%d", &X);
```

```
    // Proceso
```

RES = X % C; // Comprobamos si es divisible por 2

```
if (RES == 0) {  
    printf("El %d es par\n", X);  
} else {  
    printf("El %d es impar\n", X);  
}  
  
return 0;  
}
```

Prueba de Escritorio



U1_TALLERES1-2_ALGO

U1-TALLER1

Ejercicio 6-U1T2

Desarrolle un programa que lea tres números reales desde teclado y escribe por pantalla la media aritmética únicamente de los números positivos leídos. Puede dar dos opciones A) Empleando una iteración con salida en cabeza y B) Empleando una iteración con salida en cola



Codigo.

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

```
int main() {
```

```
    // Declarar variables
```

```
    double num, sum = 0, media;
```

```
    int cont = 0, x = 1;
```

```
    // Lectura de tres números con iteración
```

```
    do {
```

```
        printf("Ingrese un numero %d: ", x);
```

```
        scanf("%lf", &num);
```

```
        if (num > 0) {
```

```
            sum += num;
```

```
            cont++;
```

```
        }
```

```
        x++;
```

```
    } while (x <= 3);
```

```
    // Cálculo y resultado
```

```
    if (cont > 0) {
```

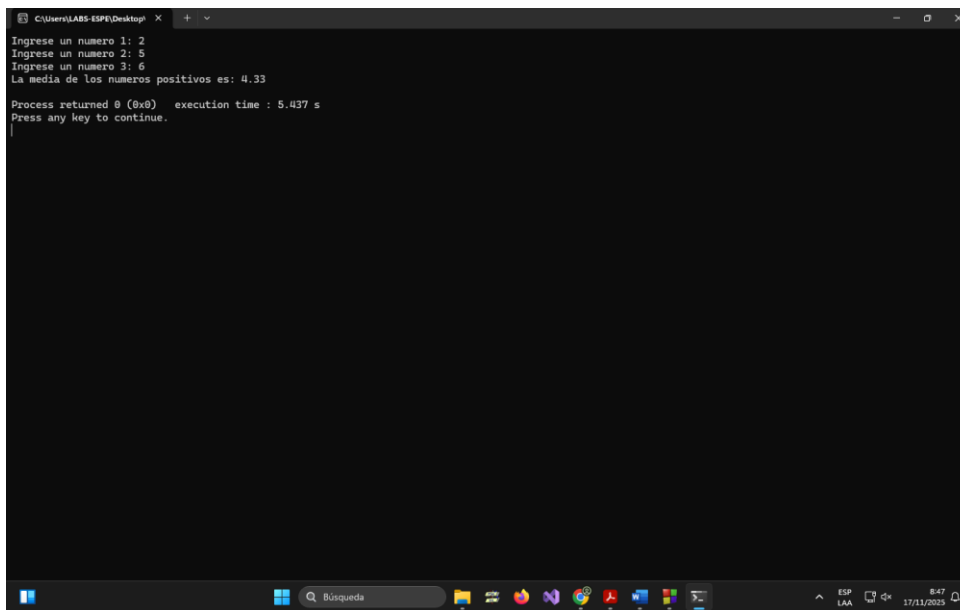
```
        media = sum / cont;
```

```
        printf("La media de los numeros positivos es: %.2lf\n", media);
```



```
} else {  
  
    printf("No se ingresaron numeros positivos.\n");  
  
}  
  
return 0;  
  
}
```

Prueba de Escritorio.



```
C:\Users\LABS-ESPE\Desktop  
Ingrese un numero 1: 2  
Ingrese un numero 2: 5  
Ingrese un numero 3: 6  
La media de los numeros positivos es: 4.33  
Process returned 0 (0x0)   execution time : 5.437 s  
Press any key to continue.
```

U1-TALLER2

EXAMEN

Preguntas y respuestas

Los estudiantes deberán identificar y corregir los errores de sintaxis, lógica o estructura.

Codigo

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

```
int main() {
```

```
    int opcion, marcador = 0;
```

// Presentación

```
printf("Bienvenido a nuestro juego de preguntas y respuestas\n");
```

```
printf("Demuestra tu cultura general y alcanza un gran marcador de puntos\n");
```

// Primera pregunta

```
printf("\nAtencion pregunta: ¿Cual es el lema de la casa Stark de Invernalía?\n");
```

```
printf("1) Se acerca el invierno\n");
```

```
printf("2) Uno para todos y todos para uno\n");
```

```
printf("3) Los Stark siempre pagan sus deudas\n");
```

```
printf("Elige una opcion introduciendo un numero (1, 2 o 3): ");
```

```
scanf("%d", &opcion);
```

// Validar opción

```
if (opcion < 1 || opcion > 3) {
```

```
    printf("No has seguido las reglas del juego. No ganas ningun punto.\n");
```

```
} else {
```

```
    if (opcion == 1) {
```

```
        printf("Correcto. El lema de los Stark es 'Se acerca el invierno'.\n");
```

```
        marcador = marcador + 1;
```

```
    } else {
```

```
        printf("Respuesta incorrecta. La opcion correcta era la 1.\n");
```

```
    }
```

```
}
```

// Segunda pregunta

```
printf("\nAtencion pregunta: ¿Que apodo tenia el autor del Don 'Quijote de la Mancha'?\\n");

printf("1) El hilarante hidalgo\\n");

printf("2) El manco de Lepanto\\n");

printf("3) El potro de Vallecas\\n");

printf("Elige una opcion introduciendo un numero (1, 2 o 3): ");

scanf("%d", &opcion);


if (opcion < 1 || opcion > 3) {

    printf("Opcion no valida.\\n");

} else {

    if (opcion == 2) {

        printf("Enhorabuena.\\n");

        marcador = marcador + 1;

    } else {

        printf("Respuesta incorrecta. Sigue intentando.\\n");

    }

}


// Resultado final

printf("\\nTu marcador final es %d punto(s).\\n", marcador);

if (marcador == 2) {

    printf("!Excelente! Has respondido todas las preguntas correctamente.\\n");

} else {

    printf("Sigue practicando, puedes mejorar tu puntuacion.\\n");

}
```

// Desafío adicional

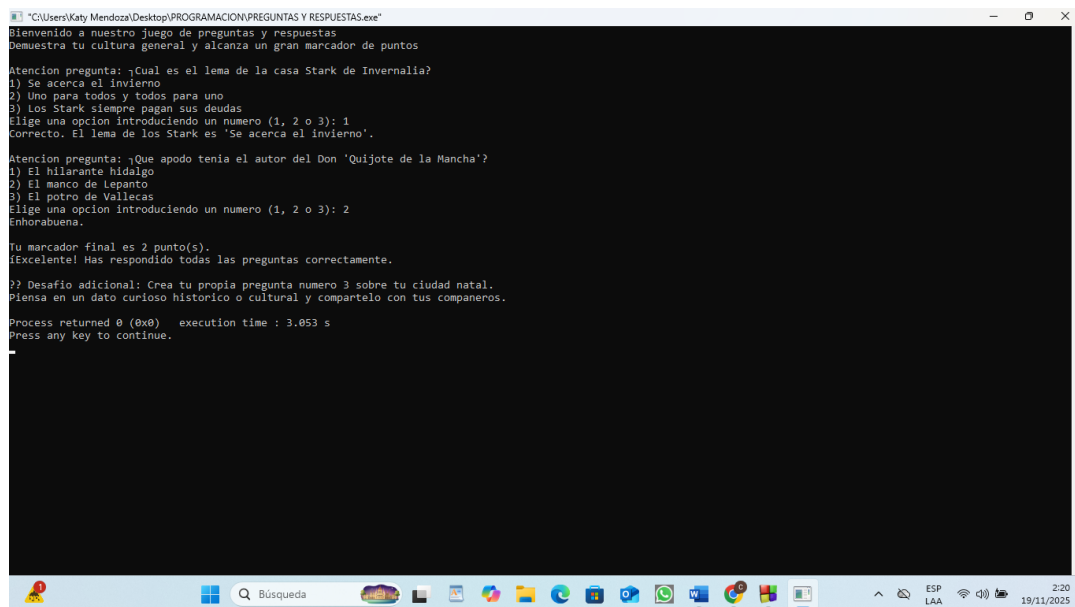
```
printf("\n?? Desafio adicional: Crea tu propia pregunta numero 3 sobre tu ciudad natal.\n");
```

```
printf("Piensa en un dato curioso historico o cultural y compartelo con tus companeros.\n");
```

```
return 0;
```

```
}
```

Prueba de Escritorio



3. CONCLUSIONES

Al realizar estas conversiones de codigo se obtuvo un mejor desempeño al momento de crearlos ya que se realizo una lectura previa a los terminos que se van a utilizar.

Al momento de crear los codigo estar concentrado en lo que se hace para asi evitar errores al momento de ejectarlo.

4. RECOMENDACIONES

Realizar una lectura previa sobre que se va a realizar y los terminos que se utilizaran en el codigo para agilizar el trabajo u desempeño al momento de hacerlo.

5. REFERENCIAS

Referencias

Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language* (2nd ed ed.). Prentice Hall. Obtenido de http://117.250.119.200:8080/jspui/bitstream/123456789/1373/1/%5BKernighan-Ritchie%5DThe_C_Programming_Language.pdf

Sebesta, R. W. (2019). *Concepts of programming languages* (12th ed. ed.). Pearson. Obtenido de https://api.pageplace.de/preview/DT0400.9781292436777_A43586160/preview-9781292436777_A43586160.pdf

Stroustrup, B. (2013). *The C++ programming language* (4th ed ed.). Addison-Wesley. Obtenido de <https://stroustrup.com/4th.html>