

Ejercicios Parcial 1.



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



NOMBRE: GORDILLO MONSERRATE

NRC: 29583

FECHA : 02/ Febrero / 2026

ASIGNATURA: Fundamentos de la programación

CARRERA: Electrónica y automatización

TEMA: EJERCICIOS PRIMER PARCIAL

EJERCICIO 1 (MEDIA DE DOS VALORES)

Tabla De Objetos

objeto	nombre	valor	tipo
valor	x	variable	real
valor	y	variable	real
madia	res	variable	real
constante	c	cte	int

Pseudocódigo

```
inico
//definir variables
definir x,y,res como real
c<- 2
//ingreso de datos
escribir "ingrese el primer numero real"
leer X
//proceso
res <- (x+y)/C
Escribir "la media de: ".x."y ".y." es "res"
fin
```

Código En Codeblobx

```
#include <stdio.h>
int main() {
    float x, y, res;
    float c = 2.0;
```

```

printf("Ingrese el primer numero real: ");
scanf("%f", &x);
printf("Ingrese el segundo numero real: ");
scanf("%f", &y);

res = (x + y) / c;

printf("La media de %.2f y %.2f es: %.2f\n", x, y, res);

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/UQlw9QHRS>

Pantallazos De Prueba

```

main.c
1 #include <stdio.h>
2
3 int main() {
4     float x, y, res;
5     float c = 2.0;
6
Ingresese el primer numero real: 9
Ingresese el segundo numero real: 25
La media de 9.00 y 25.00 es: 17.00
...
Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 2 (DIVISIBLE)

Tabla De Objetos

objeto	nombre	valor	tipo
valor	x	variable	real
valor absoluto	va	variable	real
respuesta	r	variable	real

Pseudocódigo

```

//definir variables
definir x,va,v como real
//ingreso de datos
escribir "ingrese el numero real"
leer X
//proceso
si x<0 entonces

```

```

va <- x
sino
va<- x
finsi
r<-va*va*va
//resultado
Escribir "el valor absoluto de : ".x."al cubo es "R"
fin

```

Código En Codeblob

```

#include <stdio.h>
int main() {
    float x, va, r;
    printf("Ingrese el numero real: ");
    scanf("%f", &x);
    // Proceso para valor absoluto
    if (x < 0) {
        va = -x;
    } else {
        va = x;
    }
    r = va * va * va; // Elevado al cubo
    printf("El valor absoluto de %.2f al cubo es: %.2f\n", x, r);
    return 0;
}

```

Link De GDB Online : https://onlinegdb.com/W_dGkGBB5

Pantallazos De Prueba

```

main.c
2
3 int main() {
4     float x, va, r;
5
6     printf("Ingrese el numero real: ");
7     scanf("%f", &x);
8
9     // Proceso para valor absoluto
10    if (x < 0) {
11        va = -x;
12    } else {
13        va = x;
14    }

```

input

```

Ingresar el numero real: 24
El valor absoluto de 24.00 al cubo es: 13824.00

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 3 (DIVISIBLE)

Tabla De Objetos

objeto	nombre	valor	tipo
numerador	n	variable	int
denominador	d	variable	int
resto	r	0	int

Pseudocódigo

```
inico
//definir variables
definir N,D,R como int
//ingreso de daros
escribir "ingrese el primer valor"
leer N
escribir "ingrese el segundo valor"
leer D
//proceso
R<- N%D
si R<-D entonces
escribir "el número ,“N”, es divisible por”, D,”
sino
escribir "el número ,“N”, no es divisible por”, D,”
finsi
fin
```

Código En Codeblobx

```
#include <stdio.h>
int main() {
    int N, D, R;

    printf("Ingrese el dividendo (N): ");
    scanf("%d", &N);
    printf("Ingrese el divisor (D): ");
    scanf("%d", &D);

    if (D == 0) {
        printf("Error: No se puede dividir por cero.\n");
    } else {
        R = N % D;
        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;
}

```

Link De GDB Online : <https://onlinegdb.com/CnrDUGisj>

Pantallazos De Prueba:

The screenshot shows a code editor window with the file name 'main.c'. The code is as follows:

```

1 #include <stdio.h>
2
3 int main() {
4     int N, D, R;
5
6     printf("Ingrese el dividendo (N): ");
7     scanf("%d", &N);
8     printf("Ingrese el divisor (D): ");
9     scanf("%d", &D);
10
11    if (D == 0) {
12        printf("Error: No se puede dividir por cero.\n");
13    } else {

```

Below the code, there is an 'input' field containing the user input: "Ingrese el dividendo (N): 85" and "Ingrese el divisor (D): 23". The output window shows the program's response: "El numero 85 NO es divisible por 23". At the bottom, it says "...Program finished with exit code 0" and "Press ENTER to exit console".

EJERCICIO 4 (INTERVALO)

Tabla De Objetos

objeto	nombre	valor	tipo
numero	x	variable	real
condicion 1	c1	variable	logico
condicion 2	c2	variable	logico
solucion	pertenece	variable	logico

Pseudocódigo

```

inico
//definir variables
definir x,como real
definir c1,c2,como logico
c1<- (x>0)
c2<- x<=10
pertenece<-c1 y c2

```

```
//ingreso de datos  
escribir "ingrese un numero real"  
leer X  
//proceso  
si pertenece entonces  
escribir "el número pertenece al intervalo (0,10)  
sino  
escribir "el número no pertenece al intervalo (0,10)  
finsi  
fin
```

Código En Codeblob

```
#include <stdio.h>  
int main() {  
    float x;  
    int pertenece; // En C, 1 es verdadero y 0 es falso  
  
    printf("Ingrese un numero real: ");  
    scanf("%f", &x);  
  
    // Evaluamos si esta en el rango (0, 10]  
    if (x > 0 && x <= 10) {  
        pertenece = 1;  
    } else {  
        pertenece = 0;  
    }  
  
    if (pertenece == 1) {  
        printf("El numero %.2f pertenece al intervalo (0, 10]\n", x);  
    } else {  
        printf("El numero %.2f NO pertenece al intervalo (0, 10]\n", x);  
    }  
  
    return 0;  
}
```

Link De GDB Online : <https://onlinedbg.com/aHAq5E2hB>

Pantallazos De Prueba:

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     float x;
5     int pertenece; // En C, 1 es verdadero y 0 es falso
6
7     printf("Ingrese un numero real: ");
8     scanf("%f", &x);
9
10    // Evaluamos si esta en el rango (0, 10]
11    if (x > 0 && x <= 10) {
12        pertenece = 1;
13    }
14
15    printf("El numero %f pertenece al intervalo [%d, %d]\n", x, 0, 10);
16
17    return 0;
18 }
```

```
Ingrese un numero real: 9
El numero 9.00 pertenece al intervalo [0, 10]

...Program finished with exit code 0
Press ENTER to exit console.
```

EJERCICIO 5 (CONVERSIÓN DE UNIDADES DE TIEMPO)

Tabla De Objetos

objeto	nombre	valor	tipo
numero	X	variable	Int
dias	D	variable	Int
horas	H	variable	Int
minutos	M	variable	Int
segundos	S	variable	Int
resultado	R	variable	Int

Pseudocódigo

```
inico
//definir variables
definir x,,r como entero
definir D,H,M,S como entero
//ingreso de datos
escribir "ingrese la cantidad de segundos"
leer X
//proceso
D<-X%86400
R <- X%86400
H <- R%3600
R <- R%3600
M <- R%60
S <- R%60
//Resultado
Escribir "equivale a : ",D,"dias,",H,"horas,",M,"minutos,",S,"segundos
fin
```

Código En Codeblob

```
#include <stdio.h>
int main() {
    int X, R, D, H, M, S;
    printf("Ingrese la cantidad de segundos: ");
    scanf("%d", &X);
    // Calculo de dias
    D = X / 86400;
    R = X % 86400; // Lo que sobra tras sacar los dias
    // Calculo de horas
    H = R / 3600;
    R = R % 3600; // Lo que sobra tras sacar las horas
    // Calculo de minutos y segundos
    M = R / 60;
    S = R % 60;
    printf("Equivale a: %d dias, %d horas, %d minutos, %d segundos\n", D, H, M, S);
    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/Xe-LhAQjS>

Pantallazos De Prueba:

The screenshot shows the online GDB interface with the following details:

- Code Area:**

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int X, R, D, H, M, S;
5
6     printf("Ingrese la cantidad de segundos: ");
7     scanf("%d", &X);
8
9     // Calculo de dias
10    D = X / 86400;
11    R = X % 86400; // Lo que sobra tras sacar los dias
12
13    // Calculo de horas
```
- Input Area:** Shows the user input: "Ingrese la cantidad de segundos: 4567891"
- Output Area:** Shows the program output: "Equivale a: 52 dias, 20 horas, 51 minutos, 31 segundos"
- Status Bar:** Shows "...Program finished with exit code 0" and "Press ENTER to exit console."

EJERCICIO 6 MEDIA DE NÚMEROS POSITIVOS

Tabla De Objetos

objeto	nombre	valor	tipo
valor	num	variable	real

suma	sum	variable	real
media	media	variable	real
contador	cont	variable	Int
interaccion	x	variable	Int

Pseudocódigo

```

inico
Algoritmo MediaPositivos_Mientras
    Definir num, suma, contador, i Como Real
    suma <- 0
    contador <- 0
    i <- 1
    Mientras i <= 3 Hacer
        Escribir "Ingrese el número ", i, ":""
        Leer num
        Si num > 0 Entonces
            suma <- suma + num
            contador <- contador + 1
        SiNo
            // No se hace nada si es negativo o cero
        FinSi
        i <- i + 1
    FinMientras
    Si contador > 0 Entonces
        Escribir "La media de los positivos es: ", suma / contador
    SiNo
        Escribir "No se ingresaron números positivos."
    FinSi
FinAlgoritmo

```

Código En Codeblox

```

#include <stdio.h>

int main() {
    float num, suma = 0;
    int contador = 0, i = 1;

    while (i <= 3) {
        printf("Ingrese el numero %d: ", i);
        scanf("%f", &num);
    }
}

```


Pseudocódigo

Algoritmo Alturas_Max_Min_Media

Definir altura, suma, max, min, contador Como Real

suma <- 0

contador <- 0

max <- 0

min <- 9999 // Valor inicial alto para comparar

Escribir "Ingrese altura (negativo para terminar):"

Leer altura

Mientras altura >= 0 Hacer

 contador <- contador + 1

 suma <- suma + altura

 Si altura > max Entonces

 max <- altura

 FinSi

 Si altura < min Entonces

 min <- altura

 FinSi

 Escribir "Ingrese siguiente altura:"

 Leer altura

FinMientras

Si contador > 0 Entonces

 Escribir "Media: ", suma / contador

 Escribir "Máximo: ", max

 Escribir "Mínimo: ", min

SiNo

 Escribir "No se ingresaron alturas válidas."

FinSi

FinAlgoritmo

Código En Codeblox

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

```
int main() {
```

```
    float altura, suma = 0, max = 0, min = 999.0;
```

```
int contador = 0;

printf("Ingrese altura (valor negativo para salir): ");
scanf("%f", &altura);

while (altura >= 0) {
    contador++;
    suma += altura;

    if (altura > max) {
        max = altura;
    } else { /* No es maximo */ }

    if (altura < min) {
        min = altura;
    } else { /* No es minimo */ }

    printf("Ingrese siguiente altura: ");
    scanf("%f", &altura);
}

if (contador > 0) {
    printf("\nResultados:\n");
    printf("Media: %.2f\n", suma / contador);
    printf("Maximo: %.2f\n", max);
    printf("Minimo: %.2f\n", min);
} else {
    printf("No se ingresaron datos.\n");
}
return 0;
}
```

Link De GDB Online : <https://onlined gdb.com/KXxhSDPz0m>

Pantallazos De Prueba:


```

Mientras divisor <= n Hacer
    Si n % divisor = 0 Entonces
        contador_divisores <- contador_divisores + 1
    FinSi
    divisor <- divisor + 1
FinMientras

Si contador_divisores = 2 Entonces
    Escribir "El número ", n, " es primo."
SiNo
    Escribir "El número ", n, " no es primo."
FinSi
SiNo
    Escribir "Número no válido."
FinSi
FinAlgoritmo

```

Código En Codeblox

```

#include <stdio.h>

int main() {
    int n, i = 1, divisores = 0;
    printf("Ingrese un numero entero mayor que 1: ");
    scanf("%d", &n);

    if (n > 1) {
        while (i <= n) {
            if (n % i == 0) {
                divisores++;
            } else { /* No es divisor */
                i++;
            }
        }

        if (divisores == 2) {
            printf("El numero %d es primo.\n", n);
        } else {
            printf("El numero %d no es primo.\n", n);
        }
    } else {
        printf("Entrada invalida.\n");
    }
}

```



```

divisores <- 0
j <- 1
// Verificamos si 'i' es primo
Mientras j <= i Hacer
    Si i % j = 0 Entonces
        divisores <- divisores + 1
    FinSi
    j <- j + 1
FinMientras

Si divisores = 2 Entonces
    Escribir i
FinSi

i <- i + 1
FinMientras
FinAlgoritmo

```

Código En Codeblox

```

#include <stdio.h>
int main() {
    int A, B, i, j, divisores;
    printf("Ingrese Inicio del rango (A): ");
    scanf("%d", &A);
    printf("Ingrese fin del rango (B): ");
    scanf("%d", &B);

    printf("Primos entre %d y %d:\n", A, B);

    i = A;
    while (i <= B) {
        divisores = 0;
        j = 1;
        // Comprobar si 'i' es primo
        while (j <= i) {
            if (i % j == 0) {
                divisores++;
            }
            j++;
        }
    }
}

```

```

if (divisores == 2) {
    printf("%d ", i);
}
i++;
}
printf("\n");
return 0;
}

```

Link de GDB online: <https://onlinegdb.com/r4cfG3D5I>

Pantallazos De Prueba:

```

main.c
5     printf("Ingrese inicio del rango (A): ");
6     scanf("%d", &A);
7     printf("Ingrese fin del rango (B): ");
8     scanf("%d", &B);
9
10    printf("Primos entre %d y %d:\n", A, B);
11
12    i = A;
13    while (i <= B) {
14        divisores = 0;
15        j = 1;
16        // Comprobar si 'i' es primo
17        while (j <= i) {
            .
            .
        }
    }
}

Ingrese inicio del rango (A): 8
Ingrese fin del rango (B): 16
Primos entre 8 y 16:
11 13

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 11 ESTRUCTURAS REPETITIVAS

Tabla De Objetos

objeto	nombre	valor	tipo
opcion	opcion	variable	entero
numero de teminod	n	variable	entero
numero ingresado	num	variable	entero
variable de control	i	variable	entero
primer numero	a	variable	entero
segundo numero	b	variable	entero
variable auxiliar	temp	variable	entero
primo o no	primo	variable	entero

Pseudocódigo

```
Proceso MenuPrincipal
Definir opcion, n, num, i, a, b, temp Como Entero
Definir primo Como Logico
Repetir
    Escribir "===== MENÚ PRINCIPAL ====="
    Escribir "1. Serie Fibonacci"
    Escribir "2. Operaciones"
    Escribir "3. Salir"
    Leer opcion
    Segun opcion Hacer
        1:
            Escribir "¿Cuántos términos de la serie Fibonacci quiere ver?:"
            " Leer n
            Si n > 0 Entonces
                a <- 0
                b <- 1
                Escribir "Serie Fibonacci:
                " Para i <- 1 Hasta n Con Paso 1 Hacer
                    Escribir a
                    temp <- a + b
                    a <- b
                    b <- temp
                FinPara
            SiNo Escribir "Número no valido. Debe ser un número positivo."
            FinSi
        2:
            Repetir
                Escribir "--- SUBMENÚ OPERACIONES ---"
                Escribir "2.1 Verificar Par o Impar"
                Escribir "2.2 Verificar si es Primo"
                Escribir "2.3 Regresar al Menú Principal"
                Leer opcion
                Segun opcion Hacer
                    1, 2.1:
                        Escribir "Ingrese un número positivo:
                        " Leer num Si num > 0 Entonces
                            Si num MOD 2 = 0 Entonces
                                Escribir "El número ", num, " es PAR."
                            SiNo
```

Escribir "El número ", num, " es IMPAR."

FinSi

SiNo Escribir "Número no valido. Debe ser un número positivo."

FinSi

2, 2.2:

Escribir "Ingrese un número positivo: "

Leer num

Si num > 0 Entonces

Si num = 1 Entonces

Escribir "El número NO es primo."

SiNo

primo <- Verdadero

Para i <- 2 Hasta Trunc(Raiz(num)) Hacer

Si num MOD i = 0 Entonces

primo <- Falso

FinSi

FinPara

Si primo Entonces

Escribir "El número ", num, " es PRIMO."

SiNo

Escribir "El número ", num, " NO es primo."

FinSi

FinSi

SiNo Escribir "Número no valido. Debe ser un número positivo."

FinSi

3, 2.3:

Escribir "Regresando al MENÚ PRINCIPAL..."

De Otro Modo: Escribir "Opción no valida. Intente nuevamente."

FinSegun

Hasta Que opcion = 3 O opcion = 2.3

3:

Escribir "Finalizando el programa... ¡Hasta la proxima!"

De Otro Modo: Escribir "Opción no valida. Intente nuevamente."

FinSegun

Hasta Que opcion = 3

FinProceso

Código En Codeblox

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

```

int main() {
    int opcion, n, num, i, a, b, temp;
    bool primo;

    do {
        printf("\n===== MENU PRINCIPAL =====\n");
        printf("1. Serie Fibonacci\n");
        printf("2. Operaciones\n");
        printf("3. Salir\n");
        printf("Elija una opcion: ");
        scanf("%d", &opcion);

        switch(opcion) {
            case 1:
                printf("¿Cuantos terminos de Fibonacci quiere ver?: ");
                scanf("%d", &n);
                if (n > 0) {
                    a = 0; b = 1;
                    printf("Serie: ");
                    for (i = 1; i <= n; i++) {
                        printf("%d ", a);
                        temp = a + b;
                        a = b;
                        b = temp;
                    }
                    printf("\n");
                } else {
                    printf("Numero no valido.\n");
                }
                break;

            case 2:
                do {
                    printf("\n--- SUBMENU OPERACIONES ---\n");
                    printf("1. Verificar Par o Impar\n");
                    printf("2. Verificar si es Primo\n");
                    printf("3. Regresar al Menu Principal\n");
                    scanf("%d", &opcion);

                    if (opcion == 1) {

```

```

        printf("Ingrese numero: ");
        scanf("%d", &num);
        if (num % 2 == 0) printf("Es PAR.\n");
        else printf("Es IMPAR.\n");
    } else if (opcion == 2) {
        printf("Ingrese numero: ");
        scanf("%d", &num);
        if (num <= 1) {
            printf("No es primo.\n");
        } else {
            primo = true;
            for (i = 2; i <= sqrt(num); i++) {
                if (num % i == 0) primo = false;
            }
            if (primo) printf("Es PRIMO.\n");
            else printf("No es PRIMO.\n");
        }
    }
} while (opcion != 3);
opcion = 0; // Reset para no salir del menu principal
break;
}
} while (opcion != 3);

return 0;
}

```

Link De GDB Online : <https://onlined gdb.com/7X45nyhAPR>

Pantallazos De Prueba:

```

----- MENU PRINCIPAL -----
1. Serie Fibonacci
2. Operaciones
3. Salir
Elija una opcion: 1
;Cuantos terminos de Fibonacci quiere ver?: 5
Serie: 0 1 1 2 3

----- SUBMENU OPERACIONES ---
1. Verificar Par o Impar
2. Verificar si es Primo
3. Regresar al Menu Principal
1
Ingrese numero: 8
< Es PAR.

----- SUBMENU OPERACIONES ---
1. Verificar Par o Impar
2. Verificar si es Primo
3. Regresar al Menu Principal
2
Ingrese numero: 5
Es PRIMO.

----- SUBMENU OPERACIONES ---
1. Verificar Par o Impar
2. Verificar si es Primo
3. Regresar al Menu Principal
3

```

EJERCICIO 12 (MEDIA DE DOS VALORES)

Tabla De Objetos

objeto	nombre	valor	tipo
valor	x	variable	entero
constante	c	2	entero
resultado	res	variable	entero

Pseudocódigo

```
Algoritmo Par_o_Impar
    // Definir variables
    Definir x Como Entero

    // Ingreso de datos
    Escribir "Ingrese el valor entero:"
    Leer x

    // Proceso y Resultado
    Si x % 2 == 0 Entonces
        Escribir "El numero ", x, " es Par"
    SiNo
        Escribir "El numero ", x, " es Impar"
    FinSi
FinAlgoritmo
```

Código En Codeblox

```
#include <stdio.h>
int main() {
    // Definir variables
    int x;

    // Ingreso de datos
    printf("Ingrese el valor entero: ");
    scanf("%d", &x);

    // Proceso y Resultado
    if (x % 2 == 0) {
        printf("El numero %d es Par\n", x);
```

```

} else {
    printf("El numero %d es Impar\n", x);
}

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/EEIhl8aDT>

Pantallazos De Prueba:

```

main.c
1 #include <stdio.h>
2
3 int main() {
4     // Definir variables
5     int x;
6
7     // Ingreso de datos
8     printf("Ingrese el valor entero: ");
9     scanf("%d", &x);
10
11    // Proceso y Resultado
12    if (x % 2 == 0) {
13        printf("El numero %d es Par\n", x);
14    } else {
15        printf("El numero %d es Impar\n", x);
16    }
17
18
Ingrese el valor entero: 54
El numero 54 es Par

Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 13 JUEGO DE PREGUNTAS

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
opc	opcion	Variable	Entero
marca	marcador	Variable	Entero

Pseudocódigo

```

Proceso Juego_Preguntas
// Variables de control
Definir opcion, marcador Como Entero
marcador = 0
// Presentación
Escribir "Bienvenido a nuestro juego de preguntas y respuestas"
Escribir "Demuestra tu cultura general y alcanza un gran marcador de puntos"
// Primera pregunta
Escribir "Atención pregunta: ¿Cuál es el lema de la casa Stark de Invernia?"
Escribir "1) Se acerca el invierno"
Escribir "2) Uno para todos y todos para uno"

```

Escribir "3) Los Stark siempre pagan sus deudas"
Escribir "Elige una opción introduciendo un número (1, 2 o 3):"
Leer opcion
// Validar la opción ingresada
Si opcion < 1 Y opcion > 3Entonces
 Escribir "No has seguido las reglas del juego No ganas ningún punto"
Sino
 // Verificar ANSWER
 Si opcion = 1
 Escribir "Correcto El lema de los Stark es 'Se acerca el invierno'"
 marcador == marcador + 1
 SiNo
 Escribir "Respuesta incorrecta La opción correcta era la 1."
 FinSi
 FinSi
// Segunda pregunta
Escribir "Atención, pregunta: ¿Qué apodo tenía el autor del Don 'Quijote de la Mancha'?"
Escribir "1) El hilarante hidalgo"
Escribir "2) El manco de Lepanto"
Escribir "3) El potro de Vallecás"
Escribir "Elige una opción introduciendo un número (1, 2 o 3):"
Leer opción
Si opcion < 1 Y opcion > 3 Entonces
 Escribir "Opción no válida"
 SiNo
 Si opcion == 2 Entonces
 Escribir "Enhorabuena"
 marcador <- marcador + 1
 SiNo
 Escribir "Respuesta incorrecta sigue intentando"
 FinSi
 FinSi
//Resultado Final
Escribir "Tu marcador final es " marcador, " punto(s)"
Si marcador == 2
 Escribir "¡Excelente! Has respondido todas las preguntas correctamente."
Sino
 Escribir "Sigue practicando puedes mejorar tu puntuacion"
FinSi

Escribir ""?? Desafio adicional Crea tu propia pregunta numero 3 sobre tu ciudad natal"
Escribir "Piensa en un dato curioso histórico o cultural y compártelo con tus compañeros"
Fin Proceso

Código En Codeblobx

```
#include <stdio.h>
int main() {
    int opcion, marcador = 0;

    printf("Bienvenido al juego de preguntas\n");

    // Pregunta 1
    printf("\n1. ¿Cuál es el lema de la casa Stark?\n");
    printf("1) Se acerca el invierno\n2) Uno para todos\n3) Siempre pagan deudas\n");
    scanf("%d", &opcion);

    if (opcion < 1 || opcion > 3) {
        printf("Opción no válida. 0 puntos.\n");
    } else if (opcion == 1) {
        printf("¡Correcto!\n");
        marcador++;
    } else {
        printf("Incorrecto.\n");
    }

    // Pregunta 2
    printf("\n2. ¿Apodo del autor del Quijote?\n");
    printf("1) El hidalgo\n2) El manco de Lepanto\n3) El potro\n");
    scanf("%d", &opcion);

    if (opcion < 1 || opcion > 3) {
        printf("Opción no válida.\n");
    } else if (opcion == 2) {
        printf("¡Enhorabuena!\n");
        marcador++;
    } else {
        printf("Incorrecto.\n");
    }

    printf("\nMarcador final: %d puntos.\n", marcador);
    if (marcador == 2) printf("¡Excelente desempeño!\n");
```

```

else printf("Sigue practicando.\n");

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/45zho14Lv>

Pantallazos De Prueba :

```

Bienvenido al juego de preguntas
1. ¿Cuál es el lema de la casa Stark?
1) Se acerca el invierno
2) Uno para todos
3) Siempre pagan deudas
1
;Correcto!

2. ¿Apodo del autor del Quijote?
1) El hidalgo
2) El manco de Lepanto
3) El potro
3
Incorrecto.

Marcador final: 1 puntos.
Sigue practicando.

...Program finished with exit code 0
Press ENTER to exit console

```

EJERCICO 14 PIZZA

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
radio	radio	Variable	Real
comensales	comensales	Variable	Real
porciones	porciones	Variable	Real
area	area	Variable	Real
area_persona	area_persona	Variable	Real
porc_persona	porc_persona	Variable	Real

Pseudocódigo

Algoritmo PizzaReparto

Definir radio, comensales, porciones, area, area_persona, porc_persona Como Real

Escribir "Radio de la pizza (cm):"

Leer radio

Escribir "Número de comensales:"

Leer comensales

Escribir "Número de porciones totales:"

Leer porciones

```
area <- 3.14159 * (radio * radio)
area_persona <- area / comensales
porc_persona <- porciones / comensales
```

```
Escribir "Area total: ", area, " cm2"
Escribir "A cada uno le toca: ", area_persona, " cm2 y ", porc_persona, " porciones."
FinAlgoritmo
```

Código En Codeblox

```
#include <stdio.h>
int main() {
    float radio, comensales, porciones, area;

    printf("Radio de la pizza (cm): ");
    scanf("%f", &radio);
    printf("Comensales: ");
    scanf("%f", &comensales);
    printf("Porciones totales: ");
    scanf("%f", &porciones);

    area = 3.14159 * (radio * radio);

    printf("\n--- RESULTADOS ---\n");
    printf("a) Area total de la pizza: %.2f cm2\n", area);
    printf("b) Por comensal: %.2f cm2 y %.2f porciones\n", (area/comensales),
    (porciones/comensales));

    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/ARaG7GBVi>

Pantallazos De Prueba

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     float radio, comensales, porciones, area;
5
6     printf("Radio de la pizza (cm): ");
7     scanf("%f", &radio);
8     printf("Comensales: ");
9     scanf("%f", &comensales);
10    printf("Porciones totales: ");
11    scanf("%f", &porciones);
12
13    area = 3.14159 * (radio * radio);
14
Radio de la pizza (cm): 12
Comensales: 8
Porciones totales: 4
--- RESULTADOS ---
a) Area total de la pizza: 452.39 cm2
b) Por comensal: 56.55 cm2 y 0.50 porciones

...Program finished with exit code 0
Press ENTER to exit console.
```

EJERCICIO 15 FUSION NUMERO CALORÍAS EJERCICIO 14

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
media	media	Variable	Real
comio	comio	Variable	Real
exceso	exceso	Variable	Real
calorias	calorias	Variable	Real
minutos	minutos	Variable	Real

Pseudocódigo

Algoritmo GlotonCalorias

// Datos del EJERCICIO 14

Definir comio, media, exceso, calorías, minutos Como Real

media <- porciones / comensales

Escribir "¿Cuantas porciones comio el gloton?"

Leer comio

Si comio > media Entonces

exceso <- comio - media

calorías <- exceso * 250

minutos <- calorías / 11

Escribir "Debe correr ", minutos, " minutos para quemar el exceso."

SiNo

Escribir "No comio mas que la media."

FinSi

FinAlgoritmo

Código En Codeblox

```
#include <stdio.h>
int main() {
    float radio, comensales, porciones, comio, media, exceso, calorias, minutos;
    // Reutilizamos datos del anterior
    printf("Total porciones de la pizza: ");
    scanf("%f", &porciones);
    printf("Numero de comensales: ");
    scanf("%f", &comensales);
    media = porciones / comensales;
    printf("¿Cuantas porciones engullo el comensal gloton?: ");
    scanf("%f", &comio);
    if (comio > media) {
        exceso = comio - media;
        calorias = exceso * 250;
        minutos = calorias / 11;
        printf("\nEl gloton comio %.2f porciones de mas.\n", exceso);
        printf("Calorias extra: %.2f kcal\n", calorias);
        printf("Debe hacer footing durante %.2f minutos.\n", minutos);
    } else {
        printf("Nadie comio mas que la media.\n");
    }
}

return 0;
}
```

Link de GDB online: <https://onlinegdb.com/TafFPbGFv7>

Pantallazos De Prueba :

```
main.c
2
3 int main() {
4     float radio, comensales, porciones, comio, media, exceso, calorias, minutos;
5
6     // Reutilizamos datos del anterior
7     printf("Total porciones de la pizza: ");
8     scanf("%f", &porciones);
9     printf("Numero de comensales: ");
10    scanf("%f", &comensales);
11    media = porciones / comensales;
12
13
Total porciones de la pizza: 18
Numero de comensales: 9
¿Cuantas porciones engullo el comensal gloton?: 8
El gloton comio 6.00 porciones de mas.
Calorias extra: 1500.00 kcal
Debe hacer footing durante 136.36 minutos.

...Program finished with exit code 0
Press ENTER to exit console.
```

Ejercicios Parcial 2.



ESPE



UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

NOMBRE: GORDILLO MONSERRATE

NRC: 29583

FECHA : 02/ Febrero / 2026

ASIGNATURA: Fundamentos de la programación

CARRERA: Electrónica y automatización

TEMA: EJERCICIOS SEGUNDO PARCIAL

EJERCICIO 1 ALGORITMO SWITCH VOCALES Y CONSONANTES

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
caracter	car	variable	Carácter

Pseudocódigo

Algoritmo ClasificarCaracter

Definir car Como Caracter

Escribir "Ingrese un caracter:"

Leer car

Segun car Hacer

'a', 'e', 'i', 'o', 'u':

Escribir "El caracter "", car, "" es una VOCAL MINUSCULA."

'A', 'E', 'I', 'O', 'U':

Escribir "El caracter "", car, "" es una VOCAL MAYUSCULA."

'0','1','2','3','4','5','6','7','8','9':

Escribir "El caracter "", car, "" es un NUMERO."

De Otro Modo:

Si car >= 'a' Y car <= 'z' Entonces

Escribir "Es una CONSONANTE MINUSCULA."

SiNo

Si car >= 'A' Y car <= 'Z' Entonces

Escribir "Es una CONSONANTE MAYUSCULA."

SiNo

Escribir "Es un SIMBOLO."

FinSi

FinSi

FinSegun

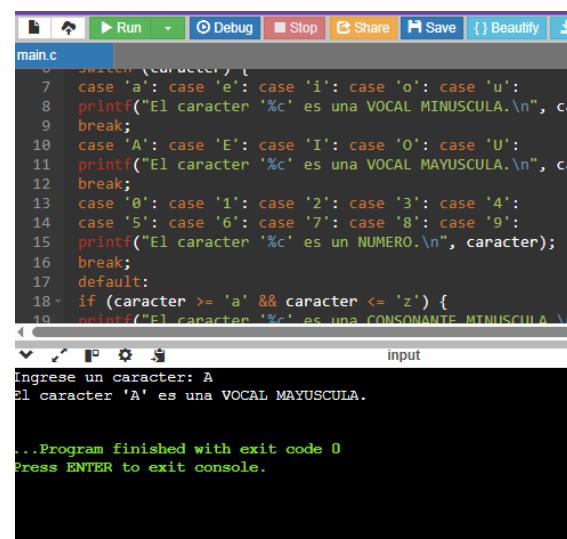
FinAlgoritmo

Código En Codeblox

```
#include <stdio.h>
int main() {
    char caracter;
    printf("Ingrese un caracter: ");
    scanf(" %c", &caracter);
    switch (caracter) {
        case 'a': case 'e': case 'i': case 'o': case 'u':
            printf("El caracter '%c' es una VOCAL MINUSCULA.\n", caracter);
            break;
        case 'A': case 'E': case 'I': case 'O': case 'U':
            printf("El caracter '%c' es una VOCAL MAYUSCULA.\n", caracter);
            break;
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            printf("El caracter '%c' es un NUMERO.\n", caracter);
            break;
        default:
            if (caracter >= 'a' && caracter <= 'z') {
                printf("El caracter '%c' es una CONSONANTE MINUSCULA.\n", caracter);
            } else if (caracter >= 'A' && caracter <= 'Z') {
                printf("El caracter '%c' es una CONSONANTE MAYUSCULA.\n", caracter);
            } else {
                printf("El caracter '%c' es un SIMBOLO.\n", caracter);
            }
            break;
    }
    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/otAuOPKAd>

Pantallazos De Prueba :



```
main.c
  6     switch (caracter) {
  7         case 'a': case 'e': case 'i': case 'o': case 'u':
  8             printf("El caracter '%c' es una VOCAL MINUSCULA.\n", caracter);
  9             break;
 10         case 'A': case 'E': case 'I': case 'O': case 'U':
 11             printf("El caracter '%c' es una VOCAL MAYUSCULA.\n", caracter);
 12             break;
 13         case '0': case '1': case '2': case '3': case '4':
 14         case '5': case '6': case '7': case '8': case '9':
 15             printf("El caracter '%c' es un NUMERO.\n", caracter);
 16             break;
 17         default:
 18             if (caracter >= 'a' && caracter <= 'z') {
 19                 printf("El caracter '%c' es una CONSONANTE MINUSCULA.\n", caracter);
 20             } else if (caracter >= 'A' && caracter <= 'Z') {
 21                 printf("El caracter '%c' es una CONSONANTE MAYUSCULA.\n", caracter);
 22             } else {
 23                 printf("El caracter '%c' es un SIMBOLO.\n", caracter);
 24             }
 25             break;
 26     }
 27     return 0;
 28 }
```

Input: Ingrese un caracter: A
Output: El caracter 'A' es una VOCAL MAYUSCULA.

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 2

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
caracter	car	variable	Carácter

Pseudocódigo

Algoritmo ClasificadorAnidado

 Definir car Como Caracter

 Escribir "Ingrese caracter: "

 Leer car

 Si car >= 'A' Y car <= 'Z' Entonces

 Segun car Hacer

 'A', 'E', 'I', 'O', 'U': Escribir "Es vocal mayuscula"

 De Otro Modo: Escribir "Es consonante mayuscula"

 FinSegun

 Sino

 Si car >= 'a' Y car <= 'z' Entonces

 Segun car Hacer

 'a', 'e', 'i', 'o', 'u': Escribir "Es vocal minuscula"

 De Otro Modo: Escribir "Es consonante minuscula"

 FinSegun

 Sino

 Si car >= '0' Y car <= '9' Entonces

 Escribir "Es un numero"

 Sino

 Escribir "Es un simbolo"

 FinSi

 FinSi

FinAlgoritmo

Código de codeblox

```
#include <stdio.h>
void main()
{
char car;
printf("Ingrese caracter: ");
scanf("%c",&car);
if (car>='A' && car<='Z')
{
switch(car)
{
case 'A': case 'E': case 'I': case 'O': case 'U':
printf("Es vocal mayuscula\n");
```

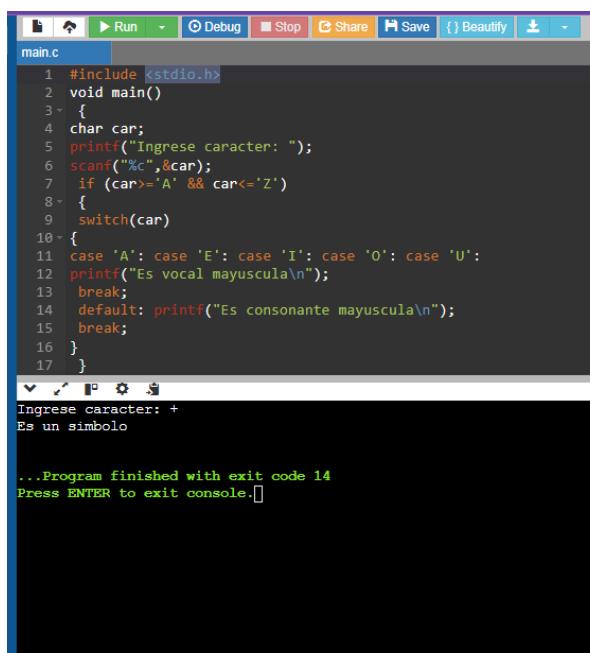
```

break;
default: printf("Es consonante mayuscula\n");
break;
}
}
else
{
if (car>='a' && car<='z')
{
switch(car)
{
case 'a': case 'e': case 'i':
case 'o': case 'u': printf("Es vocal minuscula\n");
break;
default: printf("Es consonante minuscula\n");
break;}
}
else
{
if (car>='0' && car<='9')
printf("Es un numero\n");
else
printf("Es un simbolo\n");
}
}
}
}

```

Link De GDB Online : <https://onlinegdb.com/fwGxp1m6d>

Pantallazos De Prueba



The screenshot shows a web-based C debugger interface. The top bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The code editor window contains the provided C program. The terminal window below shows the program's output when run with the character '+' as input.

```

main.c
1 #include <stdio.h>
2 void main()
3 {
4     char car;
5     printf("Ingrese caracter: ");
6     scanf("%c",&car);
7     if (car>='A' && car<='Z')
8     {
9         switch(car)
10    {
11         case 'A': case 'E': case 'I': case 'O': case 'U':
12             printf("Es vocal mayuscula\n");
13             break;
14         default: printf("Es consonante mayuscula\n");
15             break;
16     }
17 }

```

Ingrese caracter: +
Es un simbolo

...Program finished with exit code 14
Press ENTER to exit console.[]

EJERCICIO 4

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
caracter	car	variable	Carácter

Pseudocódigo

Algoritmo VerificarAcento

 Definir car Como Caracter

 Escribir "Ingrese un caracter:"

 Leer car

 Segun car Hacer

 'á','é','í','ó','ú':

 Escribir "Es una vocal acentuada."

 De Otro Modo:

 Escribir "No es una vocal acentuada."

 FinSegun

FinAlgoritmo

Código En Codeblox

```
#include <stdio.h>
#include<wchar.h>
#include <locale.h>
void main()
{
setlocale(LC_ALL, "");
wchar_t wcar;
wprintf(L"Ingrese un caracter: ");
wscanf(L"%lc", &wcar);
switch(wcar)
{
case L'á': case L'é': case L'i': case L'o': case L'u':
wprintf(L"Es una vocal acentuada.\n");
break;
default: wprintf(L"No es una vocal acentuada .\n");
break;
}
}
```

Link De GDB Online : <https://onlinegdb.com/o1aOW3uhS>

Pantallazos De Prueba :

```

1 #include <stdio.h>
2 #include<wchar.h>
3 #include <locale.h>
4 void main()
5 {
6     setlocale(LC_ALL, "");
7     wchar_t wcar;
8     wprintf(L"Ingresa un caracter: ");
9     wscanf(L"%c", &wcar);
10    switch(wcar)
11    {
12        case L'á': case L'é': case L'í': case L'ó': case L'ú':
13            wprintf(L"Es una vocal acentuada.\n");
14        break;
15        default: wprintf(L"No es una vocal acentuada .\n");
16    }
17 }
```

Ingrese un caracter: ó
Es una vocal acentuada.

...Program finished with exit code 24
Press ENTER to exit console.

EJERCICIO 5

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
caracter	car	variable	Carácter

Pseudocódigo

Algoritmo ColoresRGB

Definir car Como Caracter

Escribir "Ingrese caracter (R, G, B):"

Leer car

Segun car Hacer

'R': Escribir "El color es Rojo."

'G': Escribir "El color es Verde."

'B': Escribir "El color es Azul."

FinSegun

FinAlgoritmo

Código En Codeblox

```

#include <stdio.h>
void main()
{
char car;
{
printf("Ingrese caracter: ");
scanf("%c",&car);
switch(car)
{
case 'R':
```

```

printf("El color es Rojo.\n");
break;
case 'G': printf("El color es Verde.\n");
break;
case 'B': printf("El color es Azul .\n");
break;
}
}
}

```

Link De GDB Online : <https://onlinegdb.com/PMtFQK4bJ>

Pantallazos De Prueba :

```

main.c
1 #include <stdio.h>
2 void main()
3 {
4     char car;
5     {
6         printf("Ingrese caracter: ");
7         scanf("%c",&car);
8         switch(car)
9         {
10            case 'R':
11                printf("El color es Rojo.\n");
12                break;
13            case 'G': printf("El color es Verde.\n");
14                break;
15            case 'B': printf("El color es Azul .\n");
16        break;
17     }
18 }
19

```

EJERCICIO 6

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
Número del día	num	variable	Entero

Pseudocódigo

Algoritmo DiasSemana

Definir num Como Entero

Escribir "Ingresa un numero (1-7):"

Leer num

Segun num Hacer

- 1: Escribir "Lunes"
- 2: Escribir "Martes"
- 3: Escribir "Miercoles"
- 4: Escribir "Jueves"
- 5: Escribir "Viernes"

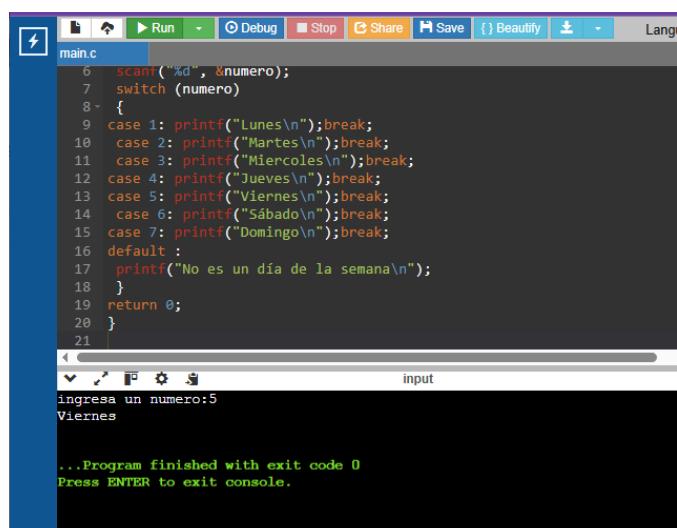
6: Escribir "Sabado"
 7: Escribir "Domingo"
 De Otro Modo: Escribir "No es un dia de la semana"
 FinSegun
 FinAlgoritmo

Código En Codeblox:

```
#include <stdio.h>
int main()
{
int numero;
printf("ingresa un numero:");
scanf("%d", &numero);
switch (numero)
{
case 1: printf("Lunes\n");break;
case 2: printf("Martes\n");break;
case 3: printf("Miercoles\n");break;
case 4: printf("Jueves\n");break;
case 5: printf("Viernes\n");break;
case 6: printf("Sábado\n");break;
case 7: printf("Domingo\n");break;
default :
printf("No es un día de la semana\n");
}
return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/Stxk78UyO>

Pantallazos De Prueba :



```
main.c
1  #include <stdio.h>
2  int main()
3  {
4      int numero;
5      printf("ingresa un numero:");
6      scanf("%d", &numero);
7      switch (numero)
8      {
9          case 1: printf("Lunes\n");break;
10         case 2: printf("Martes\n");break;
11         case 3: printf("Miercoles\n");break;
12         case 4: printf("Jueves\n");break;
13         case 5: printf("Viernes\n");break;
14         case 6: printf("Sábado\n");break;
15         case 7: printf("Domingo\n");break;
16         default :
17             printf("No es un dia de la semana\n");
18     }
19     return 0;
20 }
```

input
ingresa un numero:5
Viernes

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 7

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
op_mp	Opc. Menú Principal	1,2,3	Entero
op_sub	Opc. Submenú Expresiones	1 al 7	Entero
res_bool	Resultado booleano	Verdadero / Falso	Lógico
a_d	Coeficiente 'a'	$\neq 0$	Real
b_d	Coeficiente 'b'	Entrada	Real
c_d	Coeficiente 'c'	Entrada	Real
d_d	Discriminante	$b^2 - 4ac$	Real
x1, x2	Raíces de la ecuación	Resultado	Real

Pseudocódigo

Algoritmo MenuMatematico

Definir op_mp, op_sub Como Entero

Definir res_bool Como Logico

Definir a_d, b_d, c_d, d_d, x1, x2 Como Real

Repetir

Escribir "===== Menu Principal ====="

Escribir "1. Resolver Expresiones"

Escribir "2. ECUACIONES DE 2º GRADO"

Escribir "3. Salir"

Leer op_mp

Segun op_mp Hacer

1:

Repetir

Escribir "==== Menu Expresiones ==="

Escribir "1 a 6: Evaluar Expresiones"

Escribir "7. Regresar"

Leer op_sub

Si op_sub >= 1 Y op_sub <= 6 Entonces

Segun op_sub Hacer

1: res_bool <- $(3*2 + 8/8) \leftrightarrow (4*8 - 2)$

2: res_bool <- $(6*(3+9)/2) \geq (9-3)$

3: res_bool <- $(5+2*10) < (10/2+5)$

4: res_bool <- $(14/(3-1)+3*3) == (14-3)$

5: res_bool <- $(12*4-4) \leq (4+12/2)$

6: res_bool <- $(7+5*2) \leftrightarrow (7*5-10)$

FinSegun

Si res_bool Entonces Escribir "Verdadero" SiNo Escribir "Falso" FinSi

```

    FinSi
    Hasta Que op_sub == 7
2:
    Repetir
        Escribir "Ingrese valor a (distinto de 0):"
        Leer a_d
        Hasta Que a_d <> 0
        Escribir "Ingrese b:"
        Leer b_d
        Escribir "Ingrese c:"
        Leer c_d

        d_d <- b_d^2 - 4*a_d*c_d
        Si d_d < 0 Entonces
            Escribir "Raices imaginarias"
        SiNo
            x1 <- (-b_d + rc(d_d)) / (2*a_d)
            x2 <- (-b_d - rc(d_d)) / (2*a_d)
            Escribir "Raiz 1: ", x1, " Raiz 2: ", x2
        FinSi
    FinSegun
    Hasta Que op_mp == 3
FinAlgoritmo

```

Código En Codeblox

```

#include <stdio.h>
#include<stdlib.h>
#include <math.h>
int main()
{
    int opcion_mp, opcion_sub;
    int resultado;
    int i, j, k, m, n, p, a, b, c, x, y, u, v, q, r;
    double a_d, b_d, c_d, d_d, x1_d, x2_d;
    do
    {
        printf("===== Menu Principal =====\n");
        printf("1. Resolver Expresiones\n");
        printf("2. ECUACIONES DE 2º GRADO\n");
        printf("3. Salir\n");
        printf("===== ======\n");
        printf("Ingrese una opcion: ");
        if (scanf("%d", &opcion_mp) != 1) {
            printf("Error: Entrada no valida. Ingrese un numero.\n");
            while (getchar() != '\n');
            opcion_mp = 0;
            continue;
        }

```

```

while (getchar() != '\n');
switch(opcion_mp)
{ case 1:
do
{
printf("===== Menu Expresiones =====\n");
printf("1. Expresion 1\n");
printf("2. Expresion 2\n");
printf("3. Expresion 3\n");
printf("4. Expresion 4\n");
printf("5. Expresion 5\n");
printf("6. Expresion 6\n");
printf("7. Regresar al Menu Principal\n");
printf("=====*\n");
printf("Seleccione la expresion a evaluar: ");
if (scanf("%d", &opcion_sub) != 1)
{
printf("Entrada no valida, ingrese un numero.\n");
while (getchar() != '\n');
opcion_sub = 0;
continue;
}
while (getchar() != '\n');
if (opcion_sub >= 1 && opcion_sub <= 6)
{
switch (opcion_sub)
{
case 1: i = 4; j = 2; k = 8;
resultado = (3 * j + 8 / k) != (i * k - j);
printf("Expresion 1\n");
break;
case 2:
m = 6; n = 3; p = 9;
resultado = (m * (n + p) / 2) >= (p - n);
printf(" Expresion 2\n");
break;
case 3:
a = 5; b = 2; c = 10;
resultado = (a + b * c) < (c / b + a);
printf("Expresion 3 \n");
break;
case 4: x = 14; y = 3;
resultado = (x / (y - 1) + 3 * y) == (x - y);
printf("Expresion 4 \n");
break;
case 5: u = 12; v = 4;
resultado = (u * v - 4) <= (v + u / 2);
printf(" Expresion 5 \n");
}
}
}

```

```

break;
case 6: q = 7; r = 5;
resultado = (q + r * 2) != (q * r - 10);
printf("Expresion 6\n");
break;
}
printf("Resultado de la expresion: ");
if (resultado){
printf("Verdadero\n");
} else {
printf("Falso\n");
}
printf("Presione enter para continuar...\n");
getchar();
}
else if (opcion_sub == 7)
{
printf("Regresando al Menu Principal\n");
}
else
{
printf("Opcion no valida. Seleccione otra.\n");
}
}
while(opcion_sub != 7);
break;
case 2:
printf("== Ecuacion de 2º Grado ==\n");
do {
printf("Ingresa el valor de a: ");
scanf("%lf", &a_d);
while (getchar() != '\n'); }
while (a_d == 0.0);
printf("Ingresa el valor de b: ");
scanf("%lf", &b_d);
while (getchar() != '\n');
printf("Ingresa el valor de c: ");
scanf("%lf", &c_d);
while (getchar() != '\n');
d_d = b_d * b_d - 4.0 * a_d * c_d; if (d_d < 0.0)
{
printf(Resultado: Raices imaginarias.\n");
} else {
x1_d = (-b_d + sqrt(d_d)) / (2.0 * a_d); x2_d = (-b_d - sqrt(d_d)) / (2.0 * a_d);
printf("Resultado:\n");
printf("Raiz 1: %.4f\n", x1_d);
printf("Raiz 2: %.4f\n", x2_d);
}

```

```

printf("Presione enter para continuar...\n");
getchar();
break;
case 3: printf("Saliendo del programa.\n");
break;
default: printf("Opcion no valida. Por favor, ingrese 1, 2 o 3.\n");
printf("Presione enter para continuar...\n");
getchar();
break;
}
}
while(opcion_mp != 3);
return 0;
}

```

Link De GDB Online : <https://onlined gdb.com/FPZ2HLRMy>

Pantallazos De Prueba :

```

Ingrese una opcion: 1
==== Menu Expresiones ====
1. Expresion 1
2. Expresion 2
3. Expresion 3
4. Expresion 4
5. Expresion 5
6. Expresion 6
7. Regresar al Menu Principal
_____
Seleccione la expresion a evaluar: 5
Expresion 5
Resultado de la expresion: Falso
Presione enter para continuar...
==== Menu Expresiones ====
1. Expresion 1
2. Expresion 2
3. Expresion 3
4. Expresion 4
5. Expresion 5
6. Expresion 6
7. Regresar al Menu Principal
_____
Seleccione la expresion a evaluar: 7
Regresando al Menu Principal
===== Menu Principal =====
1. Resolver Expresiones
2. ECUACIONES DE 2o GRADO
3. Salir
_____
Ingrese una opcion: 2
    Ecuacion de 2o Grado ===
Ingresa el valor de a: 9
Ingresa el valor de b: 5
Ingresa el valor de c: 1
Resultado: Raices imaginarias.
Presione enter para continuar...

```

EJERCICIO 8

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
op	Opción del menú	1,2,3	Entero
fil	Cantidad de filas	> 0	Entero
i	Contador de filas (bucle externo)	1	Entero
j	Contador de columnas (bucle interno)	1	Entero
n	Límite de la serie	Entrada	Entero

suma	Acumulador de la serie	variable	Real
------	------------------------	----------	------

Pseudocódigo

Algoritmo FiguraYSerie

Definir op, fil, i, j, n Como Entero

Definir suma Como Real

Repetir

 Escribir "1. Figura | 2. Serie | 3. Salir"

 Leer op

 Segun op Hacer

 1:

 Escribir "Filas:"

 Leer fil

 Para i <- 1 Hasta fil Hacer

 Para j <- 1 Hasta fil Hacer

 Si i==1 o i==fil o j==1 o j==fil o i==j Entonces

 Escribir Sin Saltar "1"

 SiNo

 Escribir Sin Saltar " "

 FinSi

 FinPara

 Escribir ""

 FinPara

 2:

 Escribir "Valor n:"

 Leer n

 suma <- 0

 Para i <- 1 Hasta n Hacer

 suma <- suma + i^2

 Escribir Sin Saltar i, "^2 "

 FinPara

 Escribir "Total: ", suma

 FinSegun

 Hasta Que op == 3

FinAlgoritmo

Código En Codeblox

```
#include <stdio.h>
#include <math.h>
int main() {
int op;
int fil, i, j;
int n;
double res;
do {
printf("\n==== MENU PRINCIPAL ====\n");
```

```

printf("1) Figura con 1's (cuadrado)\n");
printf("2) Serie numerica\n");
printf("3) Salir\n");
printf("Elige una opcion: ");
scanf("%d", &op);
switch (op) { case 1: do { printf("\nIngresa el numero de filas (mayor que 0): ");
    scanf("%d", &fil);
} while (fil <= 0);
for (i = 1; i <= fil; i++) {
for (j = 1; j <= fil; j++) {
if (i == 1 || i == fil || j == 1 || j == fil || i == j) {
printf("1");
} else {
printf(" ");
}
}
printf("\n");
}
break;
case 2:
do {
printf("\nIngresa un valor para n (mayor que 0): ");
scanf("%d", &n);
} while (n <= 0);
res = 0;
printf("\nSerie: ");
for (i = 1; i <= n; i++) { res = res + i * i;
printf("%d^2", i); if (i < n) {
printf("+");
}
}
printf("\nResultado de la serie = %.0f\n", res);
break;
case 3: printf("\nSaliendo del programa...\n");
break;
default:
printf("\nOpcion no valida. Intenta de nuevo.\n");
}
} while (op != 3);
return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/qZk6TYVL8>

Pantallazos De Prueba :

```

== MENU PRINCIPAL ==
1) Figura con 1's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion: 1
Ingresa el numero de filas (mayor que 0): 5
11111
11 1
1 1 1
1 1 1
11111

== MENU PRINCIPAL ==
1) Figura con 1's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion: 2
Ingresa un valor para n (mayor que 0): 4
Serie: 1^2+2^2+3^2+4^2
Resultado de la serie = 30

== MENU PRINCIPAL ==
1) Figura con 1's (cuadrado)
2) Serie numerica
3) Salir
Elige una opcion: 3
Saliendo del programa...
...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 9 menú de figuras y serie

Tabla De Objetos

Objeto	Nombre	Valor	Tipo
op	Opción del menú	1,2,3	Entero
fil	Número de filas para la figura	> 0	Entero
n	Cantidad de términos de la serie	> 0	Entero
i	Índice de filas / Término actual	0 o 1	Entero
j	Índice de columnas	variable	Entero
res	Resultado de la suma de potencias	variable	Real

Pseudocódigo

Algoritmo Figura Y Series Potencias

Definir op, fil, n, i, j Como Entero

Definir res Como Real

Repetir

 Escribir "1. Figura | 2. Serie | 3. Salir"

 Leer op

 Segun op Hacer

 1:

 Repetir

 Escribir "Ingrese numero de filas:"

 Leer fil

 Hasta Que fil > 0

 Para i <- 0 Hasta fil-1 Hacer

 Para j <- 0 Hasta fil-1 Hacer

```

    Si i==0 o i==fil-1 o j==0 o j==fil-1 Entonces
        Escribir Sin Saltar "1"
    SiNo
        Escribir Sin Saltar " "
    FinSi
    FinPara
    Escribir ""
FinPara
2:
res <- 0
Escribir "Numero de terminos:"
Leer n
Para i <- 1 Hasta n Hacer
    res <- res + i^i
    Escribir Sin Saltar i, "^", i, " "
FinPara
Escribir "Resultado: ", res
FinSegun
Hasta Que op == 3
FinAlgoritmo

```

Código En Codeblox

```

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

int main() {
    int op, n, fil;
    float res = 0, i_float;

    do {

        res = 0;

        printf("MENU\n");
        printf("1. Figura (Cuadro Vacio)\n");
        printf("2. Serie (Suma de potencias)\n");
        printf("3. Salir\n");
        printf("Ingrese opcion: ");

        if (scanf("%d", &op) != 1) {
            while(getchar() != '\n');
            op = 0;
        }

        switch (op) {
            case 1:

```

```

do {
    printf("Ingrese numero de filas (lado del cuadrado): ");
    scanf("%d", &fil);
} while (fil <= 0);

printf("\n");

for (int i = 0; i < fil; i++) {

    for (int j = 0; j < fil; j++) {

        if (i == 0 || i == fil - 1 || j == 0 || j == fil - 1)
            printf("1 ");
        else
            printf(" ");
    }
    printf("\n");
}

printf("\nPresione Enter para continuar...");
getchar(); getchar();
break;

```

case 2:

```

do {
    printf("Ingrese numero de terminos: ");
    scanf("%d", &n);
} while (n <= 0);

printf("Serie: ");
for (int k = 1; k <= n; k++) {
    i_float = (float)k;

    printf("%.0f^%.0f", i_float, i_float);

    if (k < n) printf(" + ");

    res = res + pow(i_float, i_float);
}
printf("\nEl resultado es: %.2f\n", res);

printf("\nPresione Enter para continuar...");
getchar(); getchar();
break;

```

case 3:

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

```

        break;

default:
    printf("Opcion no valida\n");
    printf("Presione Enter para continuar...");
    getchar(); getchar();
    break;
} // fin del switch

system("cls");

} while (op != 3);

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/NR-J7eldt>

Pantallazos De Prueba

```

] MENU
1. Figura (Cuadro Vacio)
2. Serie (Suma de potencias)
3. Salir
Ingrese opcion: 1
Ingrese numero de filas (lado del cuadrado): 8

1 1 1 1 1 1 1 1
1           1
1           1
1           1
1           1
1           1
1           1
1           1
1 1 1 1 1 1 1 1

Presione Enter para continuar...
sh: 1: cls: not found
MENU
1. Figura (Cuadro Vacio)
2. Serie (Suma de potencias)
3. Salir
Ingrese opcion: 2
Ingrese numero de terminos: 9
Serie: 1^1 + 2^2 + 3^3 + 4^4 + 5^5 + 6^6 + 7^7 + 8^8 + 9^9
El resultado es: 405071328.00

Presione Enter para continuar...

```

Ejercicios Parcial 3.



ESPE



UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

NOMBRE: GORDILLO MONSERRATE

NRC: 29583

FECHA : 02/ Febrero / 2026

ASIGNATURA: Fundamentos de la programación

CARRERA: Electrónica y automatización

TEMA: EJERCICIOS TERCER PARCIAL

EJERCICIO 1

Tabla De Objetos

Objeto	Tipo	Descripción
num	Entero	Contador de números encontrados.
i	Entero	Número evaluando
n	Entero	Parámetro
res	Entero	Acumulador de los divisores
salida	Entero	condicion

Pseudocódigo

Funcion Especial(N)

Res <- 0

Para I <- 1 Hasta N-1 Con Paso 1 Hacer

Si N Mod I == 0 Entonces

 Res <- Res + I

Fin Si

Fin Para

Si Res == N Entonces

 Retornar 1

Sino

 Retornar 0

Fin Si

Fin Funcion

Algoritmo Principal

Num <- 0

I <- 1

```

Mientras Num < 4 Hacer
    Si Esperfecto(I) == 1 Entonces
        Escribir I, " Es Perfecto"
        Num <- Num + 1
    Fin Si
    I <- I + 1
Fin Mientras
Fin

```

Código En Codeblox

```

#include<stdio.h>
int esperfecto(int);
int main(void) /* antes: void main(void) o void principal(vacio) */
{
    int num=0,i=1;
    while(num<4)
    {
        if(esperfecto(i))
        {
            printf("\n%d es perfecto",i);
            num++;
        }
        i++;
    }
    return 0; /* agregado para terminar correctamente en C estándar */
}

int esperfecto(int n)
{
    int i,res=0, salida=0;
    for(i=1; i<n; i++)
    {
        if(n%i == 0)
        {
            res=res+i;
        }
    }
    if(res == n)
        salida=1;
    return(salida);
}

```

Link de GDB online: <https://www.onlinegdb.com/L1L50PL2R>

Pantallazos De Prueba

```

1 #include<stdio.h>
2
3 int esperfecto(int);
4
5 int main(void) /* antes: void main(void) o void principal(vacio) */
6 {
7     int num=9,i=1;
8     while(num<=i)
9     {
10        if(esperfecto(i))
11        {
12            printf("%d es perfecto\n",i);
13        }
14    }
15 }

```

6 es perfecto
28 es perfecto
496 es perfecto
8128 es perfecto

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 2

Tabla De Objetos

Objeto	Tipo	Descripción
Inicio	Entero	Límite inferior del rango.
fin	Entero	Límite superior del rango.
resultado	Estructura	Contiene primos y entero
n	Entero	Número a evaluar esprimo.
limite	Entero	Raíz cuadrada de n

Pseudocódigo

Estructura Resultado

Primos[1000]: Entero

Cantidad: Entero

Fin Estructura

Funcion Esprimo(N)

Si N <= 1 Retornar Falso

Si N == 2 Retornar Verdadero

Si N Mod 2 == 0 Retornar Falso

Límite <- Raiz_cuadrada(N)

Para I <- 3 Hasta Limite Con Paso 2 Hacer

 Si N Mod I == 0 Retornar Falso

 Fin Para

 Retornar Verdadero

Fin Funcion

Algoritmo Principal

Escribir "Inicio Del Rango: "

```
Leer Inicio  
Escribir "Fin Del Rango: "  
Leer Fin
```

```
Res.Cantidad <- 0  
Para I <- Inicio Hasta Fin Hacer  
    Si Esprimo(I) Entonces  
        Res.Primos[Res.Cantidad] <- I  
        Res.Cantidad <- Res.Cantidad + 1  
    Fin Si  
Fin Para
```

```
Escribir "Vector De Primos: ", Res.Primos  
Escribir "Total: ", Res.Cantidad  
Fin
```

Código En Codeblox

```
#include <stdio.h>  
#include <stdbool.h>  
#include <math.h>  
  
// Estructura simple para almacenar primos  
struct Resultado {  
    int primos[1000]; // Tamaño fijo para simplificar  
    int cantidad;  
};  
  
// Función esprimo exactamente como se pide  
bool esprimo(int n) {  
    if (n <= 1) return false;  
    if (n == 2) return true;  
    if (n % 2 == 0) return false;  
  
    int limite = sqrt(n);  
    for (int i = 3; i <= limite; i += 2) {  
        if (n % i == 0) return false;  
    }  
  
    return true;  
}  
  
int main() {  
    int Inicio, fin;  
  
    printf("=====\\n");
```

```

printf("BUSCADOR DE NUMEROS PRIMOS\n");
printf("=====\\n\\n");

// Leer rango
printf("Inicio del rango: ");
scanf("%d", &Inicio);
printf("Fin del rango: ");
scanf("%d", &fin);

// Crear estructura para resultados
struct Resultado resultado;
resultado.cantidad = 0;

// Buscar primos
for (int i = Inicio; i <= fin; i++) {
    if (esprimo(i)) {
        resultado.primos[resultado.cantidad] = i;
        resultado.cantidad++;
    }
}

// Mostrar vector de primos
printf("\nVector de numeros primos:\\n");
printf("[");
for (int i = 0; i < resultado.cantidad; i++) {
    printf("%d", resultado.primos[i]);
    if (i < resultado.cantidad - 1) printf(", ");
}
printf("]\\n");

// Resumen
printf("\\nTotal: %d numeros primos\\n", resultado.cantidad);

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/hLdBangC4>

Pantallazos De Prueba

```

47     }
48 }
49 // Mostrar vector de primos
50 printf("\nVector de numeros primos:\n");
51 printf("[" );
52 for (int i = 0; i < resultado.cantidad; i++) {
53     printf("%d", resultado.primos[i]);
54     if (i < resultado.cantidad - 1) printf(", ");
55 }
56 printf("]\n");
57 }
58 // Resumen
59 printf("\nTotal: %d numeros primos\n", resultado.cantidad);
60

```

BUSCADOR DE NUMEROS PRIMOS

=====

Inicio del rango: 4
Fin del rango: 12

Vector de numeros primos:
[5, 7, 11]

Total: 3 numeros primos

...Program finished with exit code 0
Press ENTER to exit console.[]

EJERCICIO 3

Tabla De Objetos

Objeto	Tipo	Descripción
vec[13]	Entero	Almacena los números del 3 al 15.
fact[13]	Real	Almacena los resultados factoriales.
num	Entero	Parámetro de la función calc_fact.
j	Entero	Contador para el bucle

Pseudocódigo

Funcion Calc_fact(Num)

```

F <- 1
Si Num >= 0 Entonces
    Para J <- 2 Hasta Num Hacer
        F <- F * J
    Fin Para
Fin Si
Retornar F
Fin Funcion

```

Algoritmo Principal

```

Vec <- {3,4,5,6,7,8,9,10,11,12,13,14,15}
Para I <- 0 Hasta 12 Hacer
    Resultado <- Calc_fact(Vec[I])
    Escribir "El Factorial De ", Vec[I], " Es ", Resultado
Fin Para
Fin

```

Código En Codeblox

```
#include <stdio.h>
#define N 13

float calc_fact(int);

int main(void) /* antes: void main(void) */
{
    /* Vector con los valores de 3 a 15, para replicar tu salida */
    int vec[N] = {3,4,5,6,7,8,9,10,11,12,13,14,15};
    float fact[N] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1};
    int i;

    for(i=0; i<N; i++)
    {
        fact[i] = calc_fact(vec[i]);
        printf("\nEL factorial de %d es %1.0f", vec[i], fact[i]);
    }

    /* Pausa estándar (sin conio.h), compatible con OnlineGDB */
    printf("\n\nPresione ENTER para finalizar...");
    getchar(); /* consume posible '\n' */
    getchar(); /* espera ENTER */

    return 0; /* final correcto */
}

float calc_fact(int num)
{
    int j;
    float fact = 1;
    if(num >= 0)
    {
        for(j = 2; j <= num; j++)
        {
            fact = fact * j;
        }
    }
    return fact;
}
```

Link De GDB Online : <https://www.onlinegdb.com/3cpfsRDqX>

Pantallazos De Prueba

```
1
2 #include <stdio.h>
3 #define N 13
4
5 float calc_fact(int n)
6 {
7     if (n == 1) return 1;
8     else return n * calc_fact(n - 1);
9 }
10
11 int main()
12 {
13     int n = N;
14     float fact;
15
16     fact = calc_fact(n);
17
18     printf("EL factorial de %d es %f\n", n, fact);
19
20     n--;
21
22     while (n > 0) {
23         fact = calc_fact(n);
24         printf("EL factorial de %d es %f\n", n, fact);
25         n--;
26     }
27
28     return 0;
29 }
```

```
EL factorial de 3 es 6
EL factorial de 4 es 24
EL factorial de 5 es 120
EL factorial de 6 es 720
EL factorial de 7 es 5040
EL factorial de 8 es 40320
EL factorial de 9 es 362880
EL factorial de 10 es 3628800
EL factorial de 11 es 39916800
EL factorial de 12 es 479001600
EL factorial de 13 es 6227020800
EL factorial de 14 es 87178289152
EL factorial de 15 es 1307674279936

Presione ENTER para finalizar...□
```

EJERCICIO 4

Tabla De Objetos

Objeto	Tipo	Descripción
FactData	Estructura	Contiene ny fact .
datos[15]	Arreglo de Estructura	registros para cada número del 1 al 15.
n	Entero	Número actual para el cálculo

Pseudocódigo

```
Funcion Calc_fact(N)
    Si N <= 1 Entonces
        Retornar 1
    Sino
        Retornar N * Calc_fact(N - 1)
    Fin Si
Fin Funcion

Algoritmo Principal
    Para I <- 0 Hasta 14 Hacer
        Datos[I].N <- I + 1
        Datos[I].Fact <- Calc_fact(Datos[I].N)
    Fin Para
    // Mostrar Resultados...
Fin
```

Código En Codeblox

```
#include <stdio.h>
struct FactData {
    int n;
    long long fact;
};

long long calc_fact(int n) {
    if (n <= 1) return 1;
    return n * calc_fact(n - 1);
}

int main() {
    struct FactData datos[15];

    printf("=====\\n");
    printf("FACTORIALES RECURSIVOS\\n");
    printf("=====\\n\\n");

    printf("Vector original:\\n");
    for (int i = 0; i < 15; i++) {
        datos[i].n = i + 1;
        printf("%d ", datos[i].n);
    }

    printf("\\n\\nCalculando factoriales...\\n\\n");

    for (int i = 0; i < 15; i++) {
        datos[i].fact = calc_fact(datos[i].n);
    }

    printf("Resultados:\\n");
    printf("-----\\n");
    for (int i = 0; i < 15; i++) {
        printf("%2d! = %lld\\n", datos[i].n, datos[i].fact);
    }

    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/9od0s09bCG>

Pantallazos De Prueba :

```

28     datos[i].fact = calc_fact(datos[i].n);
29 }
30
31
32 //-----//
33
34 #include <iostream>
35 #include <vector>
36
37 using namespace std;
38
39
40 int main()
41 {
42     vector<int> datos;
43
44     datos.push_back(1);
45     datos.push_back(2);
46     datos.push_back(3);
47     datos.push_back(4);
48     datos.push_back(5);
49     datos.push_back(6);
50     datos.push_back(7);
51     datos.push_back(8);
52     datos.push_back(9);
53     datos.push_back(10);
54     datos.push_back(11);
55     datos.push_back(12);
56     datos.push_back(13);
57     datos.push_back(14);
58     datos.push_back(15);
59
60     cout << "Vector original:" << endl;
61     for (int i = 0; i < datos.size(); i++)
62         cout << datos[i] << " ";
63
64     cout << endl << "Calculando factorial..." << endl;
65
66     for (int i = 0; i < datos.size(); i++)
67     {
68         cout << "-----" << endl;
69         cout << i << "!" << " = ";
70         cout << datos[i].fact << endl;
71     }
72
73     cout << endl << "...Program finished with exit code 0" << endl;
74     cout << endl << "Press ENTER to exit console." << endl;
75
76     return 0;
77 }

```

EJERCICIO 5

Tabla De Objetos

Objeto	Tipo	Descripción
per[10]	Estructura	almacena los datos de hasta 10 personas.
n	Entero	Cantidad de personas a ingresar.
suma	Entero	Acumulador para calcular la media de edad.
max	Entero	Almacena el número máximo de repeticiones de un nombre.
mas_repetido	Cadena	Almacena el nombre que más veces aparece.
rep	Entero	Campo de la estructura para contar repeticiones individuales.

Pseudocódigo

Algoritmo Registropersonas

Leer N

Suma <- 0

Para I <- 0 Hasta N-1 Hacer

 Leer Per[I].Nom, Per[I].Edad, Per[I].Ciu

 Suma <- Suma + Per[I].Edad

 Per[I].Rep <- 1

Fin Para

// Calcular Repeticiones

Max <- 0

Para I <- 0 Hasta N-1 Hacer

 Para J <- I+1 Hasta N-1 Hacer

 Si Per[I].Nom == Per[J].Nom Entonces

 Per[I].Rep <- Per[I].Rep + 1

 Per[J].Rep <- Per[J].Rep + 1

```

    Fin Si
    Fin Para
    Si Per[i].Rep > Max Entonces
        Max <- Per[i].Rep
        Mas_repetido <- Per[i].Nom
    Fin Si
    Fin Para

    Escribir "Media Edad: ", Suma / N
    Escribir "Nombre Más Repetido: ", Mas_repetido
Fin

```

Código En Codeblox

```

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

struct p {
    char nom[50];
    int edad;
    char ciu[50];
    int rep;
};

int main() {
    struct p per[10];
    int n, i, j, suma = 0, max = 0;
    char mas_repetido[50];

    printf("Cuantas personas? ");
    scanf("%d", &n);
    getchar();

    // Entrada
    for(i = 0; i < n; i++) {
        printf("\nPersona %d:\n", i+1);
        printf("Nombre: "); fgets(per[i].nom, 50, stdin);
        per[i].nom[strlen(per[i].nom)-1] = '\0';

        printf("Edad: "); scanf("%d", &per[i].edad);
        getchar();
        suma += per[i].edad;

        printf("Ciudad: "); fgets(per[i].ciu, 50, stdin);
        per[i].ciu[strlen(per[i].ciu)-1] = '\0';
        per[i].rep = 1;
    }
}

```

```

// Calcular repeticiones
for(i = 0; i < n; i++) {
    for(j = i+1; j < n; j++) {
        if(strcmp(per[i].nom, per[j].nom) == 0) {
            per[i].rep++;
            per[j].rep++;
        }
    }
    if(per[i].rep > max) {
        max = per[i].rep;
        strcpy(mas_repetido, per[i].nom);
    }
}

// Salida
printf("\nDATOS:\n");
for(i = 0; i < n; i++) {
    printf("%s - %d años - %s (rep: %d)\n",
           per[i].nom, per[i].edad, per[i].ciu, per[i].rep);
}

printf("\nMedia edad: %.1f\n", (float)summa/n);
printf("Nombre mas repetido: %s (%d veces)\n", mas_repetido, max);

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/aXRZchkd7>

Pantallazos De Prueba :

```

main.c
28     suma += per[i].edad;
Cuantas personas? 2
Persona 1:
Nombre: CARMEN
Edad: 12
Ciudad: QUITO

Persona 2:
Nombre: EMILIO
Edad: 16
Ciudad: CUENCA

DATOS:
CARMEN - 12 años - QUITO (rep: 1)
EMILIO - 16 años - CUENCA (rep: 1)

Media edad: 14.0
Nombre mas repetido: CARMEN (1 veces)

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 6

Tabla De Objetos

Objeto	Tipo	Descripción
alfabeto[27]	Estructura	Almacena el código y la matriz de cada letra.
mi_matriz[8][8]	Entero (Matriz)	Matriz de entrada que se desea reconocer.
coincidencias	Entero	Contador de bits iguales entre dos matrices.
similitud	Real	Porcentaje de coincidencia
max_similitud	Real	El porcentaje más alto encontrado hasta el momento.

Pseudocódigo

```

Funcion Busca_caracter(Mp[8][8], Tab_let[27])
    Max_similitud <- 0
    Mejor_letra <- '?'
    Para L <- 0 Hasta 26 Hacer
        Coincidencias <- 0
        Para I <- 0 Hasta 7 Hacer
            Para J <- 0 Hasta 7 Hacer
                Si Mp[I][J] == Tab_let[L].Mptos[I][J] Entonces
                    Coincidencias <- Coincidencias + 1
                Fin Si
            Fin Para
        Fin Para
        Similitud <- Coincidencias / 64.0
        Si Similitud > Max_similitud Entonces
            Max_similitud <- Similitud
            Mejor_letra <- Tab_let[L].Cod_ascii
        Fin Si
    Fin Para
    Retornar Mejor_letra
Fin Funcion

```

Código En Codeblox

```

#include <stdio.h>
struct letras {
    char cod_ASCII;
    int mptos[8][8];
};

char busca_caracter(int mp[8][8], struct letras tab_let[27]) {
    float max_similitud = 0;
    char mejor_letra = '?';

    for (int l = 0; l < 27; l++) {

```

```

int coincidencias = 0;

for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        if (mp[i][j] == tab_let[l].mptos[i][j]) {
            coincidencias++;
        }
    }
}

float similitud = (float)coincidencias / 64.0;

if (similitud > max_similitud) {
    max_similitud = similitud;
    mejor_letra = tab_let[l].cod_ASCII;
}
}

return mejor_letra;
}

int main() {
    struct letras alfabeto[27];
    int mi_matriz[8][8];

    alfabeto[0].cod_ASCII = 'A';
    int ejemplo_A[8][8] = {
        {0,0,0,1,1,0,0,0},
        {0,0,1,0,0,1,0,0},
        {0,1,0,0,0,0,1,0},
        {1,0,0,0,0,0,0,1},
        {1,1,1,1,1,1,1,1},
        {1,0,0,0,0,0,0,1},
        {1,0,0,0,0,0,0,1},
        {1,0,0,0,0,0,0,1}
    };

    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            alfabeto[0].mptos[i][j] = ejemplo_A[i][j];
        }
    }

    alfabeto[1].cod_ASCII = 'B';
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {

```

```

        alfabeto[1].mptos[i][j] = 0;
    }
}

for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        mi_matriz[i][j] = ejemplo_A[i][j];
    }
}

mi_matriz[3][4] = 0;
mi_matriz[5][5] = 1;

char resultado = busca_caracter(mi_matriz, alfabeto);

printf("RECONOCIMIENTO DE CARACTERES\n");
printf("=====\\n\\n");

printf("Matriz de entrada:\\n");
for (int i = 0; i < 8; i++) {
    printf(" ");
    for (int j = 0; j < 8; j++) {
        printf("%d ", mi_matriz[i][j]);
    }
    printf("\\n");
}

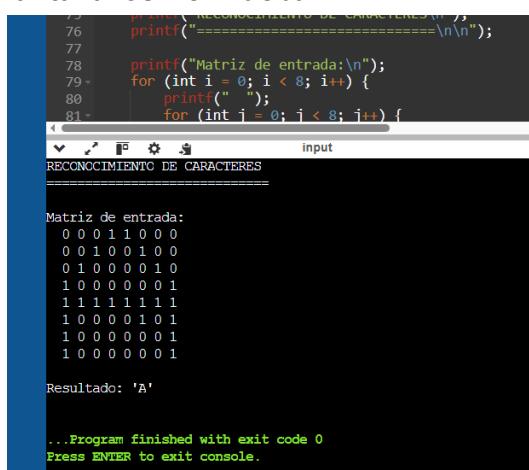
printf("\\nResultado: '%c'\\n", resultado);

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/9hk6YOX8q8>

Pantallazos De Prueba:



```

75
76 printf("RECONOCIMIENTO DE CARACTERES\\n");
77
78 printf("Matriz de entrada:\\n");
79 for (int i = 0; i < 8; i++) {
80     printf(" ");
81     for (int j = 0; j < 8; j++) {
82         printf("%d ", mi_matriz[i][j]);
83     }
84     printf("\\n");
85 }

86 printf("\\nResultado: '%c'\\n", resultado);

87
88 return 0;
89 }

RECONOCIMIENTO DE CARACTERES
=====
Matriz de entrada:
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
1 0 0 0 0 1 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1

Resultado: 'A'

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 7

Tabla De Objetos

Objeto	Tipo	Descripción
Punto	Estructura	Almacena coordenadas reales x e y.
Poligono	Estructura	Contiene el número de vértices y un arreglo de Punto.
per	Real	Acumulador de longitud total de los lados.
area	Real	Resultado del cálculo.
dx, dy	Real	Diferencias de distancia entre dos puntos consecutivos.

Pseudocódigo

Funcion Areapoligono(P)

 Area <- 0

 Para I <- 0 Hasta P.Nvert - 1 Hacer

 J <- (I + 1) Mod P.Nvert

 Area <- Area + (P.Vert[I].X * P.Vert[J].Y)

 Area <- Area - (P.Vert[J].X * P.Vert[I].X)

 Fin Para

 Retornar Abs(Area) / 2.0

Fin Funcion

Algoritmo Principal

Pol <- Leerpoligono()

 Escribir "Perímetro: ", Perimetropoligono(Pol)

 Escribir "Área: ", Areapoligono(Pol)

Fin

Código En Codeblox

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

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

```
#define MAX_VERTICES 100
```

```
typedef struct {
```

```
    float x, y;
```

```
} Punto;
```

```
typedef struct {
```

```
    int nvert;
```

```
    Punto vert[MAX_VERTICES];
```

```
} Poligono;
```

```

// Leer polígono
Poligono leerPoligono() {
    Poligono p;

    do {
        printf("Número de vértices (3-%d): ", MAX_VERTICES);
        scanf("%d", &p.nvert);
    } while (p.nvert < 3 || p.nvert > MAX_VERTICES);

    printf("Introduce coordenadas (x y) de cada vértice:\n");
    for (int i = 0; i < p.nvert; i++) {
        printf("Vértice %d: ", i + 1);
        scanf("%f %f", &p.vert[i].x, &p.vert[i].y);
    }

    return p;
}

// Calcular perímetro
float perimetroPoligono(Poligono p) {
    float per = 0;

    for (int i = 0; i < p.nvert; i++) {
        int j = (i + 1) % p.nvert;
        float dx = p.vert[j].x - p.vert[i].x;
        float dy = p.vert[j].y - p.vert[i].y;
        per += sqrt(dx*dx + dy*dy);
    }

    return per;
}

// Calcular área (método del determinante)
float areaPoligono(Poligono p) {
    float area = 0;

    for (int i = 0; i < p.nvert; i++) {
        int j = (i + 1) % p.nvert;
        area += p.vert[i].x * p.vert[j].y;
        area -= p.vert[j].x * p.vert[i].y;
    }

    return fabs(area) / 2.0;
}

// Mostrar polígono
void mostrarPoligono(Poligono p) {
    printf("\nPolígono de %d vértices:\n", p.nvert);
}

```

```

        for (int i = 0; i < p.nvert; i++) {
            printf(" V%d: (%.2f, %.2f)\n", i+1, p.vert[i].x, p.vert[i].y);
        }
    }

int main() {
    printf("== Analisis DE Poligonos ==\n\n");

    Poligono pol = leerPoligono();

    mostrarPoligono(pol);

    float per = perimetroPoligono(pol);
    float area = areaPoligono(pol);

    printf("\nRESULTADOS:\n");
    printf("Perimetro: %.2f unidades\n", per);
    printf("Area:      %.2f unidades cuadradas\n", area);

    return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/iK0bMlloCD>

Pantallazos De Prueba

```

input
== Analisis DE Poligonos ==
Número de vertices (3-100): 3
Introduce coordenadas (x y) de cada vertice:
Vertice 1: 8.2
8.2
Vertice 2: 9.6
9.6
Vertice 3: 4.5
2.2}

Poligono de 3 vertices:
V1: (8.20, 8.20)
V2: (9.60, 9.60)
V3: (4.50, 2.20)

RESULTADOS:
Perimetro: 18.02 unidades
Area:      1.61 unidades cuadradas

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 8

Tabla De Objetos

Objeto	Tipo	Descripción
men	Estructura	Contiene las cadenas game_over y se_acabo.
m	Puntero	Referencia a la estructura men en la función prn_cad.

fin	Estructura	Instancia que contiene el mensaje original "GAME OVER".
n	Entero	Cantidad de caracteres a imprimir en cada iteración.

Pseudocódigo

Funcion Prn_cad(M: Puntero A Estructura, N: Entero)

I <- 0

Mientras I < N Hacer

 Escribir M->Game_over[I] Sin Saltar Línea

 I <- I + 1

Fin Mientras

 Saltar Linea

 M->Se_acabo <- "Insert Coin"

Fin Funcion

Algoritmo Principal

 Fin.Game_over <- "Game Over"

 Para I <- 0 Hasta 9 Hacer

 Llamar Prn_cad(Fin, I + 1)

 Fin Para

 Escribir Fin.Se_acabo

Fin

Código En Codeblox

```
// problema4_4.c
```

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

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

```
#define N 10
```

```
#define M 12
```

```
struct mensaje {
```

```
    char game_over[N];
```

```
    char se_acabo[M];
```

```
};
```

```
typedef struct mensaje men;
```

```
void prn_cad(men *m, int n) {
```

```
    int i = 0;
```

```
    while (i < n) {
```

```
        printf("%c", m->game_over[i]);
```

```
        i++;
```

```
}
```

```
    printf("\n"); // Para que se vea cada línea como en la figura
```

```
    strcpy(m->se_acabo, "INSERT COIN");
```

```

}

int main(void) {
    men fin = { "GAME OVER", "" };
    int i;
    for (i = 0; i < N; i++)
        prn_cad(&fin, i + 1); // i+1 para imprimir G, GA, ..., GAME OVER

    printf("%s\n", fin.se_acabo);
    return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/zJL9xbzEj>

Pantallazos De Prueba

```

n.c
8- struct mensaje {
9-     char game_over[N];
0-     char se_acabo[M];
1- };
2- typedef struct mensaje men;
3-
4- void prn_cad(men *m, int n) {
5-     int i = 0;
6-     while (i < n) {
7-         printf("%c", m->game_over[i]);
8-         if (i == 3) {
9-             printf("GA");
10-            i++;
11-        }
12-        if (i == 6) {
13-            printf("GAME");
14-            i++;
15-        }
16-        i++;
17-    }
18- }
19-
20- int main(void) {
21-     men fin = { "GAME OVER", "" };
22-     int i;
23-     for (i = 0; i < N; i++)
24-         prn_cad(&fin, i + 1); // i+1 para imprimir G, GA, ..., GAME OVER
25-
26-     printf("%s\n", fin.se_acabo);
27-     return 0;
28- }

G
GA
GAM
GAME
GAME
GAME O
GAME OV
GAME CVE
GAME OVER
GAME OVER
INSERT COIN

...Program finished with exit code 0
Press ENTER to exit console. []

```

EJERCICIO 9

Tabla De Objetos

Objeto	Tipo	Descripción
frase	Cadena	Almacena el texto ingresado por el usuario.
letras	Entero	Puntero que apunta a la variable que acumula el total de letras.
espacios	Entero	Puntero que apunta a la variable que acumula los espacios.
cadena	Carácter	Se utiliza para recorrer la frase posición por posición.

Pseudocódigo

Funcion Contarletrasespacios(Cadena: Puntero, Letras: Puntero, Espacios: Puntero)

*Letras <- 0

*Espacios <- 0

Mientras *Cadena != '\0' Hacer

```

Si Es_letra(*Cadena) Entonces
    *Letras <- *Letras + 1
Sino Si *Cadena == '' Entonces
    *Espacios <- *Espacios + 1
Fin Si
Cadena <- Cadena + 1 // Mover El Puntero Al Siguiente Carácter
Fin Mientras
Fin Funcion

```

Algoritmo Principal

```

Escribir "Ingrese Una Frase: "
Leer Frase
Llamar Contarletrasespacios(Frase, Letras, Espacios)
Escribir "Letras: ", Letras
Escribir "Espacios: ", Espacios
Fin

```

Código En Codeblox

```

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

// Función que recibe un puntero a cadena y cuenta letras y espacios
void contarLetrasEspacios(char *cadena, int *letras, int *espacios) {
    *letras = 0;
    *espacios = 0;

    while (*cadena != '\0') {
        if (isalpha(*cadena)) { // Usando isalpha() de ctype.h
            (*letras)++;
        } else if (*cadena == ' ') {
            (*espacios)++;
        }
        cadena++;
    }
}

int main() {
    char frase[100];
    int letras, espacios;

    printf("Ingrese una frase: ");
    fgets(frase, sizeof(frase), stdin);

    // Eliminar el salto de línea

```

```

frase[strcspn(frase, "\n")] = 0;

// Llamar a la función con puntero
contarLetrasEspacios(frase, &letras, &espacios);

printf("\n--- Resultados ---\n");
printf("Frase: \"%s\"\n", frase);
printf("Letras: %d\n", letras);
printf("Espacios: %d\n", espacios);
printf("Total caracteres: %zu\n", strlen(frase));

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/vCAHSQfIH>

Pantallazos De Prueba :

```

1.c
2.     cadena++;
3. }
4.
5. int main()
6. {
7.     char frase[100];
8.
9.     printf("Ingrese una frase: ");
10.    scanf("%s", frase);
11.
12.    contarLetrasEspacios(frase, &letras, &espacios);
13.
14.    printf("\n--- Resultados ---\n");
15.    printf("Frase: \"%s\"\n", frase);
16.    printf("Letras: %d\n", letras);
17.    printf("Espacios: %d\n", espacios);
18.    printf("Total caracteres: %zu\n", strlen(frase));
19.
20.    return 0;
21. }

Ingrese una frase: EL PERR LOCO

--- Resultados ---
Frase: "EL PERRO LOCO"
Letras: 11
Espacios: 2
Total caracteres: 14

...Program finished with exit code 0
Press ENTER to exit console.

```

EJERCICIO 10

Tabla De Objetos

Objeto	Tipo	Descripción
frase	Cadena	Texto original ingresado.
sinEspacios	Cadena	Texto procesado sin espacios y en minúsculas.
Inicio	Carácter	Apunta al primer carácter de la cadena.
fin	Carácter	Apunta al último carácter de la cadena.

Pseudocódigo

Fucion Espalindromo(Cadena: Cadena)
 Inicio <- Dirección Del Primer Carácter De Cadena
 Fin <- Dirección Del Último Carácter De Cadena
 Mientras Inicio < Fin Hacer
 Si *Inicio != *Fin Entonces

```

    Retornar 0 // No Es Palíndromo
Fin Si
Inicio <- Inicio + 1
Fin <- Fin - 1
Fin Mientras
Retornar 1 // Es Palíndromo
Fin Funcion

```

Algoritmo Principal

```

Leer Frase
Sinespacios <- Eliminarespaciessimulas(Frase)
Si Espalindromo(Sinespacios) == 1 Entonces
    Escribir "Es Palíndromo"
Sino
    Escribir "No Es Palíndromo"
Fin Si
Fin

```

Código En Codeblox

```

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

// Función para quitar espacios de una cadena
void eliminarEspacios(char *destino, const char *origen) {
    char *d = destino;
    const char *s = origen;

    while (*s != '\0') {
        if (*s != ' ') {
            *d = tolower(*s); // Convertir a minúscula
            d++;
        }
        s++;
    }
    *d = '\0'; // Terminar la cadena
}

// Función para verificar palíndromo usando punteros
int esPalindromo(const char *cadena) {
    const char *Inicio = cadena;
    const char *fin = cadena + strlen(cadena) - 1;

    while (Inicio < fin) {
        if (*Inicio != *fin) {
            return 0; // No es palíndromo
        }
        Inicio++;
    }
}

```

```

        fin--;
    }
    return 1; // Es palíndromo
}

int main() {
    char frase[100];
    char sinEspacios[100];

    printf("Ingrese una frase: ");
    fgets(frase, sizeof(frase), stdin);

    // Eliminar el salto de línea
    frase[strcspn(frase, "\n")] = 0;

    // Crear una versión sin espacios y en minúsculas
    eliminarEspacios(sinEspacios, frase);

    printf("\nFrase original: \"%s\"\n", frase);
    printf("Frase sin espacios: \"%s\"\n", sinEspacios);

    // Verificar si es palíndromo
    if (esPalindromo(sinEspacios)) {
        printf("La frase ES un palíndromo (ignorando espacios)\n");
    } else {
        printf("La frase NO es un palíndromo\n");
    }

    return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/sg4lhav3a6>

Pantallazos De Prueba :

```

8     printf("\nFrase original: \"%s\"\n", frase);
9     printf("Frase sin espacios: \"%s\"\n", sinEspacios);
0
1     // Verificar si es palíndromo
2     if (esPalindromo(sinEspacios)) {
3         printf("La frase ES un palíndromo (ignorando espacios)\n");
4     } else {
5         printf("La frase NO es un palíndromo\n");
6     }
7
8     return 0;
9
0
1

```

Ingrese una frase: EL GATO VOLADOR

Frase original: "EL GATO VOLADOR"
Frase sin espacios: "elgatovolador"
La frase NO es un palíndromo

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 11

Tabla De Objetos

Objeto	Tipo	Descripción
nombre	Cadena	Almacena el nombre.
apellido	Cadena	Almacena el apellido.
nombreCompleto	Cadena	Destino de la unión de ambas cadenas.
d	Carácter	Puntero de destino para escribir en nombreCompleto.
s	Carácter	Puntero de origen para leer desde nombre o apellido.

Pseudocódigo

Funcion Unircadenas(Destino, Cadena1, Cadena2)

```
D <- Inicio De Destino
S <- Inicio De Cadena1
// Copiar Primera Cadena
Mientras *S != '\0' Hacer
    *D <- *S
    D <- D + 1, S <- S + 1
Fin Mientras
// Agregar Espacio Intermedio
*D <- ''
D <- D + 1
// Copiar Segunda Cadena
S <- Inicio De Cadena2
Mientras *S != '\0' Hacer
    *D <- *S
    D <- D + 1, S <- S + 1
Fin Mientras
*D <- '\0' // Fin De Cadena
Fin Funcion
```

Código En Codeblox

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

// Función para concatenar dos cadenas con un espacio entre ellas
void unirCadenas(char *destino, const char *cadena1, const char *cadena2) {
    char *d = destino;
    const char *s = cadena1;

    // Copiar primera cadena
    while (*s != '\0') {
        *d = *s;
        d++;
        s++;
    }

    // Agregar espacio intermedio
    *d = ' ';
    d++;

    // Copiar segunda cadena
    while (*s != '\0') {
        *d = *s;
        d++;
        s++;
    }

    // Terminar la cadena resultante
    *d = '\0';
}
```

```

        d++;
        s++;
    }

// Agregar espacio
*d = ' ';
d++;

// Resetear puntero para segunda cadena
s = cadena2;

// Copiar segunda cadena
while (*s != '\0') {
    *d = *s;
    d++;
    s++;
}

// Terminar la cadena
*d = '\0';
}

int main() {
    char nombre[50];
    char apellido[50];
    char nombreCompleto[100];

    printf("Ingrese su nombre: ");
    fgets(nombre, sizeof(nombre), stdin);
    nombre[strcspn(nombre, "\n")] = 0;

    printf("Ingrese su apellido: ");
    fgets(apellido, sizeof(apellido), stdin);
    apellido[strcspn(apellido, "\n")] = 0;

    // Usar función personalizada
    unirCadenas(nombreCompleto, nombre, apellido);

    printf("\n--- Resultado ---\n");
    printf("Nombre: %s\n", nombre);
    printf("Apellido: %s\n", apellido);
    printf("Nombre completo: \"%s\"\n", nombreCompleto);

    return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/e-Ax7vBVGR>

Pantallazos De Prueba :

```
4     fgets(apellido, sizeof(apellido), stdin);
5     apellido[strcspn(apellido, "\n")] = 0;
6
7     // Usar función personalizada
8     unirCadenas(nombreCompleto, nombre, apellido);
9
10    printf("\n--- Resultado ---\n");
11    printf("Nombre: %s\n", nombre);
12    printf("Apellido: %s\n", apellido);
13    printf("Nombre completo: \"%s\"\n", nombreCompleto);
14
```

Ingrese su nombre: CARMEZA
Ingrese su apellido: GONZALES

--- Resultado ---
Nombre: CARMEZA
Apellido: GONZALES
Nombre completo: "CARMEZA GONZALES"

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 12

Tabla De Objetos

Objeto	Tipo	Descripción
texto	Cadena	El texto completo donde se buscará.
palabra	Cadena	La palabra que se desea encontrar.
posicion	Carácter	Puntero donde inicia la palabra.
indice	Entero	Resultado de resta de punteros

Pseudocódigo

Algoritmo BuscarPalabra

 Texto <- "Hola Muchachos Bienvenidos"

 Palabra <- "Muchachos"

// Strstr Busca La Subcadena Y Devuelve Un Puntero

 Posicion <- Buscar_subcadena(Texto, Palabra)

 Si Posicion != Nulo Entonces

 // La Resta De Punteros Da La Distancia Entre Ellos

 Indice <- Posicion - Inicio De Texto

 Escribir "Aparece En La Posición: ", Indice

 Sino

 Escribir "No Se Encontró"

 Fin Si

Fin

Código En Codeblox

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

int main() {
    char texto[] = "hola muchachos bienvenidos";
    char palabra[] = "muchachos";
    char *posicion;

    posicion = strstr(texto, palabra);

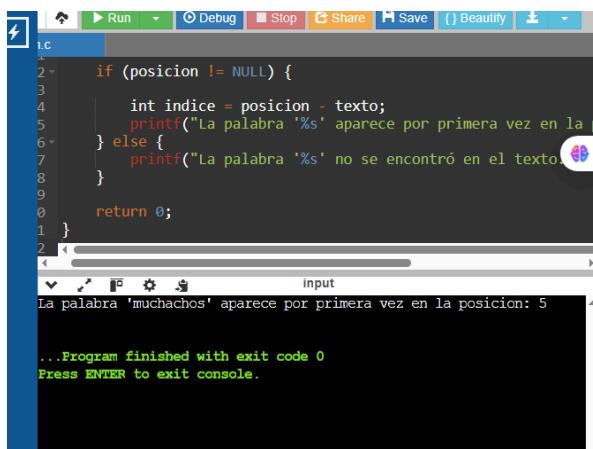
    if (posicion != NULL) {

        int indice = posicion - texto;
        printf("La palabra '%s' aparece por primera vez en la posicion: %d\n", palabra, indice);
    } else {
        printf("La palabra '%s' no se encontró en el texto.\n", palabra);
    }

    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/1JfI7SZP>

Pantallazos De Prueba :



The screenshot shows the Codeblox IDE interface. The code editor displays a C program that uses the `strstr` function to find the first occurrence of the word "muchachos" in the string "hola muchachos bienvenidos". The output window shows the result: "La palabra 'muchachos' aparece por primera vez en la posicion: 5". The status bar at the bottom indicates the program finished with exit code 0.

EJERCICIO 13

Tabla De Objetos

Objeto	Tipo	Descripción
cadena	Cadena	Almacena el texto original.
invertida	Cadena	Almacena el texto al revés.

longitud	Entero	Cantidad total de caracteres de la cadena original.
i	Entero	Índice que recorre la cadena original desde el final.
j	Entero	Índice que recorre la cadena invertida desde el inicio.

Pseudocódigo

Algoritmo Invertircadena

Leer Cadena

Longitud <- Longitud_de(Cadena)

J <- 0

Para I <- Longitud - 1 Hasta 0 Con Paso -1 Hacer

 Invertida[J] <- Cadena[I]

 J <- J + 1

Fin Para

 Invertida[J] <- '\0' // Carácter De Fin De Cadena

 Escribir "Original: ", Cadena

 Escribir "Invertida: ", Invertida

Fin

Código En Codeblox

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

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

```
int main() {
    char cadena[100];
    char invertida[100];
    int i, j, longitud;

    printf("Ingrese una cadena: ");
    fgets(cadena, sizeof(cadena), stdin);
    cadena[strcspn(cadena, "\n")] = 0;

    longitud = strlen(cadena);

    // Invertir la cadena
    j = 0;
    for (i = longitud - 1; i >= 0; i--) {
        invertida[j] = cadena[i];
        j++;
    }
    invertida[j] = '\0'; // Terminar la cadena invertida

    printf("\nResultados:\n");
    printf("Original: \"%s\"\n", cadena);
    printf("Invertida: \"%s\"\n", invertida);
```

```

        printf("Longitud: %d caracteres\n", longitud);

    return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/nKxZ6nJKfH>

Pantallazos De Prueba :

```

1.c
1     invertida[j] = '\0'; // Terminar la cadena invertida
2
3     printf("\nResultados:\n");
4     printf("Original: \"%s\"\n", cadena);
5     printf("Invertida: \"%s\"\n", invertida);
6     printf("Longitud: %d caracteres\n", longitud);

7
8     return 0;
9 }
0
1

```

Ingrese una cadena: 892

Resultados:
Original: "892"
Invertida: "298"
Longitud: 3 caracteres

...Program finished with exit code 0
Press ENTER to exit console.

EJERCICIO 14

Tabla De Objetos

Objeto	Tipo	Descripción
texto	Cadena	Frase ingresada por el usuario.
vocales	Entero	Contador acumulativo de vocales encontradas.
letra	Carácter	Variable temporal para evaluar el carácter actual del bucle.
i	Entero	Índice para recorrer el arreglo de caracteres.

Pseudocódigo

Algoritmo Contarvocales

Leer Texto

Vocales <- 0

Para I <- 0 Hasta Longitud_de(Texto) - 1 Hacer

 Letra <- Texto[I]

 Si Letra == 'A' O Letra == 'A' O Letra == 'E' ... (Repetir Con I, O, U) Entonces

 Vocales <- Vocales + 1

 Fin Si

Fin Para

Escribir "Total De Vocales: ", Vocales
Fin

Código En Codeblobx

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

int main() {
    char texto[100];
    int vocales = 0;
    int i;

    printf("Ingrese un texto: ");
    fgets(texto, sizeof(texto), stdin);
    texto[strcspn(texto, "\n")] = 0;

    // Contar vocales
    for (i = 0; texto[i] != '\0'; i++) {
        char letra = texto[i];

        if (letra == 'a' || letra == 'A' ||
            letra == 'e' || letra == 'E' ||
            letra == 'i' || letra == 'I' ||
            letra == 'o' || letra == 'O' ||
            letra == 'u' || letra == 'U') {
            vocales++;
        }
    }

    printf("\nResultados:\n");
    printf("Texto: \"%s\"\n", texto);
    printf("Total de vocales: %d\n", vocales);
    printf("Longitud total: %d caracteres\n", (int)strlen(texto));

    return 0;
}
```

Link De GDB Online : <https://onlinegdb.com/PD8cjRvRK>

Pantallazos De Prueba :

```

1 | #include <cs50.h>
2 | #include <stdio.h>
3 |
4 | {
5 |     int i;
6 |     int count = 0;
7 |     char texto[10];
8 |     for (i = 0; i < 10; i++) {
9 |         if (isvowel(texto[i])) {
10|             count++;
11|         }
12|     }
13|     printf("\nResultados:\n");
14|     printf("Texto: \"%s\"\n", texto);
15|     printf("Total de vocales: %d\n", count);
16|     printf("Longitud total: %d caracteres\n", (int)strlen(texto));
17|
18|     return 0;
19| }

```

Ingrese un texto: CARRITOS

Resultados:

Texto: "CARRITOS"

Total de vocales: 3

Longitud total: 8 caracteres

...Program finished with exit code 0

Press ENTER to exit console.

EJERCICIO 15

Tabla De Objetos

Objeto	Tipo	Descripción
Estudiante	Estructura	Contiene id, apellidos, nombres y edad.
f	Archivo	Puntero al archivo físico estudiantes.txt.
linea	Cadena	Búfer para leer cada renglón del archivo.
op	Entero	Opción seleccionada en el menú principal.
temp	Archivo	Puntero a un archivo temporal usado para borrar o actualizar.

Pseudocódigo

Algoritmo Crud_estudiantes

Repetir

 Escribir "1. Agregar, 2. Listar, 3. Consultar, 4. Actualizar, 5. Eliminar, 0. Salir"

 Leer Op

 Segun Op Hacer

 Caso 1: // Agregar

 Leer Datos Del Estudiante

 Abrir Archivo En Modo "A" (Añadir)

 Escribir Datos En Archivo Con Formato "Id;Nom;Ape;Edad"

 Cerrar Archivo

 Caso 2: // Listar

 Abrir Archivo En Modo "R" (Lectura)

 Mientras Haya Lineas En Archivo Hacer

 Leer Linea Y Mostrar En Pantalla

 Fin Mientras

 Cerrar Archivo

Caso 5: // Eliminar (Lógica De Archivo Temporal)

Leer Id_a_eliminar

Abrir Archivo Original En "R"

Abrir Archivo Temporal En "W"

Mientras Haya Lineas En Original Hacer

 Si Id_linea != Id_a_eliminar Entonces

 Escribir Linea En Archivo Temporal

 Fin Si

Fin Mientras

Cerrar Ambos

Borrar Original, Renombrar Temporal Como Original

Fin Segun

Hasta Que Op == 0

Fin

Código En Codeblox

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

/* =====
   Configuración del archivo
   ===== */
#define ARCHIVO_EST "estudiantes.txt"

/* =====
   Modelo de datos (estructura)
   ===== */
typedef struct {
    char id[20];
    char apellidos[50];
    char nombres[50];
    int edad;
} Estudiante;

/* =====
   Utilidades de entrada segura
   ===== */

/* Elimina el '\n' que deja fgets (si existe) */
void limpiarNuevaLinea(char *s) {
    s[strcspn(s, "\n")] = '\0';
}
```

```

/* Lee una cadena con etiqueta usando fgets */
void leerCadena(const char *etiqueta, char *dest, int tam) {
    printf("%s", etiqueta);
    if (fgets(dest, tam, stdin) == NULL) {
        /* Si hay error/EOF, dejar cadena vacía */
        dest[0] = '\0';
        return;
    }
    limpiarNuevaLinea(dest);
}

/* Lee un entero de forma robusta usando fgets + strtol */
int leerEntero(const char *etiqueta) {
    char buf[64];
    long val;
    char *endptr;
    for (;;) {
        printf("%s", etiqueta);
        if (fgets(buf, sizeof(buf), stdin) == NULL) {
            /* EOF: devolver 0 por defecto */
            return 0;
        }
        val = strtol(buf, &endptr, 10);
        if (endptr != buf) {
            return (int)val;
        }
        printf("Entrada invalida. Intente de nuevo.\n");
    }
}

/* =====
   Parser de línea a estructura
   ===== */
/* Convierte "id;apellidos;nombres;edad" a Estudiante.
   Retorna 1 si se pudo parsear, 0 si la línea es inválida. */
int parsearEstudiante(const char *linea, Estudiante *e) {
    /* %[^;] lee hasta ';' */
    int n = sscanf(linea, "%19[^;];%49[^;];%49[^;];%d",
                   e->id, e->apellidos, e->nombres, &e->edad);
    return (n == 4);
}

/* =====
   Verificación de ID duplicado
   ===== */
int existId(const char *idBuscado) {
    FILE *f = fopen(ARCHIVO_EST, "r");

```

```

if (f == NULL) return 0; /* Si no existe el archivo, no hay IDs guardados */

char linea[256];
Estudiante e;

while (fgets(linea, sizeof(linea), f)) {
    if (parsearEstudiante(linea, &e) && strcmp(e.id, idBuscado) == 0) {
        fclose(f);
        return 1;
    }
}
fclose(f);
return 0;
}

/* =====
Operación: Crear (Aregar)
===== */
void agregarEstudiante(void) {
    Estudiante nuevo;

    leerCadena("Ingrese ID: ", nuevo.id, sizeof(nuevo.id));
    if (nuevo.id[0] == '\0') { printf("ID vacio.\n"); return; }

    if (existId(nuevo.id)) {
        printf("Ya existe un estudiante con ese ID.\n");
        return;
    }

    leerCadena("Ingrese apellidos: ", nuevo.apellidos, sizeof(nuevo.apellidos));
    leerCadena("Ingrese nombres: ", nuevo.nombres, sizeof(nuevo.nombres));
    nuevo.edad = leerEntero("Ingrese edad: ");

    FILE *f = fopen(ARCHIVO_EST, "a");
    if (f == NULL) {
        printf("Error: no se pudo abrir/crear el archivo.\n");
        return;
    }

    /* Formato con ';' por campo (una línea por estudiante) */
    fprintf(f, "%s;%s;%s;%d\n", nuevo.id, nuevo.apellidos, nuevo.nombres, nuevo.edad);
    fclose(f);

    printf("Estudiante agregado correctamente.\n");
}

/* =====
Operación: Leer (Listar)
===== */

```

```

=====
void listarEstudiantes(void) {
    FILE *f = fopen(ARCHIVO_EST, "r");
    if (f == NULL) {
        printf("No hay datos aun (archivo inexistente).\n");
        return;
    }

    char linea[256];
    Estudiante e;

    printf("\n%-10s %-20s %-5s\n", "ID", "APELLOS", "NOMBRES", "EDAD");
    printf("-----\n");

    while (fgets(linea, sizeof(linea), f)) {
        if (parsearEstudiante(linea, &e)) {
            printf("%-10s %-20s %-5d\n",
                   e.id, e.apellidos, e.nombres, e.edad);
        }
    }

    fclose(f);
}

/* =====
Operación: Buscar por ID
===== */
int buscarPorId(const char *idBuscado, Estudiante *salida) {
    FILE *f = fopen(ARCHIVO_EST, "r");
    if (f == NULL) return 0;

    char linea[256];
    Estudiante e;

    while (fgets(linea, sizeof(linea), f)) {
        if (parsearEstudiante(linea, &e) && strcmp(e.id, idBuscado) == 0) {
            *salida = e;
            fclose(f);
            return 1;
        }
    }

    fclose(f);
    return 0;
}

void consultarEstudiante(void) {
    char id[20];

```

```

Estudiante e;

leerCadena("Ingrese ID a buscar: ", id, sizeof(id));
if (buscarPorId(id, &e)) {
    printf("Encontrado: %s %s (Edad: %d)\n", e.nombres, e.apellidos, e.edad);
} else {
    printf("No se encontro el ID.\n");
}
}

/* =====
Operación: Actualizar
(reescritura con archivo temporal)
===== */

void actualizarEstudiante(void) {
    char idObjetivo[20];
    leerCadena("Ingrese ID a actualizar: ", idObjetivo, sizeof(idObjetivo));

    FILE *f = fopen(ARCHIVO_EST, "r");
    if (f == NULL) {
        printf("No existe el archivo.\n");
        return;
    }

    FILE *temp = fopen("tmp.txt", "w");
    if (temp == NULL) {
        fclose(f);
        printf("No se pudo crear archivo temporal.\n");
        return;
    }

    char linea[256];
    Estudiante e;
    int actualizado = 0;

    while (fgets(linea, sizeof(linea), f)) {
        if (parsearEstudiante(linea, &e) && strcmp(e.id, idObjetivo) == 0) {
            /* Solicitar nuevos datos */
            leerCadena("Nuevos apellidos: ", e.apellidos, sizeof(e.apellidos));
            leerCadena("Nuevos nombres: ", e.nombres, sizeof(e.nombres));
            e.edad = leerEntero("Nueva edad: ");

            fprintf(temp, "%s;%s;%s;%d\n", e.id, e.apellidos, e.nombres, e.edad);
            actualizado = 1;
        } else {
            /* Reescribir la línea (si tiene formato válido) */
            Estudiante cpy;
            if (parsearEstudiante(linea, &cpy)) {

```

```

        fprintf(temp, "%s;%s;%s;%d\n", cpy.id, cpy.apellidos, cpy.nombres, cpy.edad);
    }
}
}

fclose(f);
fclose(temp);

remove(ARCHIVO_EST);
rename("tmp.txt", ARCHIVO_EST);

if (actualizado) printf("Registro actualizado.\n");
else      printf("No se encontro el ID.\n");
}

/* =====
Operación: Eliminar (baja física)
===== */
void eliminarEstudiante(void) {
char idEliminar[20];
leerCadena("Ingrese ID a eliminar: ", idEliminar, sizeof(idEliminar));

FILE *f = fopen(ARCHIVO_EST, "r");
if (f == NULL) {
    printf("No existe el archivo.\n");
    return;
}

FILE *temp = fopen("tmp.txt", "w");
if (temp == NULL) {
    fclose(f);
    printf("No se pudo crear archivo temporal.\n");
    return;
}

char linea[256];
Estudiante e;
int eliminado = 0;

while (fgets(linea, sizeof(linea), f)) {
    if (parsearEstudiante(linea, &e)) {
        if (strcmp(e.id, idEliminar) != 0) {
            fprintf(temp, "%s;%s;%s;%d\n", e.id, e.apellidos, e.nombres, e.edad);
        } else {
            eliminado = 1;
        }
    }
}
}

```

```

fclose(f);
fclose(temp);

remove(ARCHIVO_EST);
rename("tmp.txt", ARCHIVO_EST);

if (eliminado) printf("Registro eliminado.\n");
else      printf("No se encontro el ID.\n");
}

/* =====
Menú principal
=====
int menu(void) {
    char opcion[16];
    printf("\n==== CRUD Estudiantes (TXT) ===\n");
    printf("1. Agregar\n");
    printf("2. Listar\n");
    printf("3. Consultar por ID\n");
    printf("4. Actualizar\n");
    printf("5. Eliminar\n");
    printf("0. Salir\n");
    printf("Seleccione una opcion: ");
    if (fgets(opcion, sizeof(opcion), stdin) == NULL) return 0;
    return atoi(opcion);
}

int main(void) {
    int op;
    do {
        op = menu();
        switch (op) {
            case 1: agregarEstudiante(); break;
            case 2: listarEstudiantes(); break;
            case 3: consultarEstudiante(); break;
            case 4: actualizarEstudiante(); break;
            case 5: eliminarEstudiante(); break;
            case 0: printf("Saliendo...\n"); break;
            default: printf("Opcion invalida.\n"); break;
        }
    } while (op != 0);
    return 0;
}

```

Link De GDB Online : <https://onlined gdb.com/5Qozc4JExC>

Pantallazos De Prueba :

```
--- CRUD Estudiantes (TXT) ---
1. Agregar
2. Listar
3. Consultar por ID
4. Actualizar
5. Eliminar
0. Salir

Seleccione una opcion: 1
Ingresar apellidos: GORUILLO CARGUA
Ingresar nombres: MONSEERAT ANAH
Ingresar edad: 12
Estudiante agregado correctamente.

--- CRUD Estudiantes (TXT) ---
1. Agregar
2. Listar
3. Consultar por ID
4. Actualizar
5. Eliminar
0. Salir

Seleccione una opcion: 2
ID      APELLIDOS      NOMBRES      EDAD
100562645  GORUILLO CARGUA  MONSEERAT ANAH  12

--- CRUD Estudiantes (TXT) ---
1. Agregar
2. Listar
3. Consultar por ID
4. Actualizar
5. Eliminar
0. Salir

Seleccione una opcion: 3
Ingresar ID a buscar: 100562645
Encontrado: MONSEERAT ANAH GORUILLO CARGUA (Edad: 12)

--- CRUD Estudiantes (TXT) ---
1. Agregar
2. Listar
3. Consultar por ID
4. Actualizar
5. Eliminar
0. Salir

Seleccione una opcion:
```

EJERCICIO 16

tabla de valores

Objeto	Tipo	Descripción
nombres[5][15]	Cadena	Almacena 5 nombres de hasta 15 caracteres.
edad[5]	Entero	Almacena las edades correspondientes a cada nombre.
aux	Cadena	Variable para intercambiar nombres durante el ordenamiento.
auxi	Entero	Variable temporal para intercambiar edades.
pos	Entero	Guarda el índice donde se encontró a la persona buscada.
sn	Carácter	Almacena la respuesta 'S' o 'N' para confirmar cambios.

Pseudocódigo

Algoritmo Gestiónpersonas

// 1. Ingreso De Datos

Para I <- 0 Hasta 4 Hacer

 Leer Nombres[I], Edad[I]

Fin Para

// 2. Ordenamiento Burbuja (Alfabético)

Para I <- 0 Hasta 3 Hacer

 Para J <- I + 1 Hasta 4 Hacer

 Si Comparar(Nombres[I], Nombres[J]) > 0 Entonces

 Intercambiar(Nombres[I], Nombres[J])

 Intercambiar(Edad[I], Edad[J])

 Fin Si

Fin Para

Fin Para

// 3. Búsqueda

Escribir "Persona A Buscar: "

Leer Buscar

Pos <- -1

Para I <- 0 Hasta 4 Hacer

Si Nombres[I] == Buscar Entonces

 Pos <- I

 Romper Bucle

Fin Si

Fin Para

// 4. Modificación Opcional

Si Pos != -1 Entonces

 Escribir "Desea Cambiar La Información (S/N)?"

 Leer Sn

 Si Sn == 'S' Entonces

 Leer Nueva_opcion (1.Nombre O 2.Edad)

 Actualizar Datos En Nombres[Pos] O Edad[Pos]

 Re-Ordenar Lista

 Fin Si

 Fin Si

Fin

Código En Codeblox

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

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

```
#include <conio.h>
```

```
#define TAM 5
```

```
#define NOM 15
```

```
void main()
```

```
{
```

```
    // Arreglos y variables como en tu estructura
```

```
    char nombres[TAM][NOM];
```

```
    char aux[NOM];      // variable temporal para intercambiar los nombres
```

```
    int edad[TAM];
```

```
    char buscar[NOM];
```

```
    int auxi;
```

```
    int cambia;        // opción de cambio (1=Nombre, 2=Edad)
```

```
    char sn;          // respuesta S/N
```

```

int i, j;

// Etiquetas en palabras para "Persona UNO ... CINCO"
const char *etiquetas[TAM] = {"UNO", "DOS", "TRES", "CUATRO", "CINCO"};

// ===== INGRESO DE LA INFORMACIÓN =====
for (i = 0; i < TAM; i++)
{
    printf("Persona %s\n", etiquetas[i]);
    printf("Nombre: ");
    scanf("%s", nombres[i]);
    printf("Edad: ");
    scanf("%d", &edad[i]);
}

// ===== IMPRESIÓN (SIN ORDENAR) =====
printf("\n");
printf("No.\tNombre\tEdad\n");
printf("--\t-----\t---\n");
for (i = 0; i < TAM; i++)
{
    printf("%d\t%s\t%d\n", (i + 1), nombres[i], edad[i]);
}

// ===== ORDENAR ALFABÉTICAMENTE POR NOMBRE =====
for (i = 0; i < TAM - 1; i++)
{
    for (j = i + 1; j < TAM; j++)
    {
        if (strcmp(nombres[i], nombres[j]) > 0)
        {
            // Intercambiar nombres
            strcpy(aux, nombres[i]);
            strcpy(nombres[i], nombres[j]);
            strcpy(nombres[j], aux);

            // Intercambiar edades en la misma posición
            auxi = edad[i];
            edad[i] = edad[j];
            edad[j] = auxi;
        }
    }
}

// ===== IMPRESIÓN (ORDENADA) =====
printf("\n");
printf("No.\tNombre\tEdad\n");
printf("--\t-----\t---\n");

```

```

for (i = 0; i < TAM; i++)
{
    printf("%d\t%s\t%d\n", (i + 1), nombres[i], edad[i]);
}

// ===== BÚSQUEDA =====
printf("\nPersona a buscar: ");
scanf("%s", buscar);

int pos = -1;
for (i = 0; i < TAM; i++)
{
    if (strcmp(nombres[i], buscar) == 0)
    {
        pos = i;
        break;
    }
}

if (pos == -1)
{
    printf("No existe la persona '%s'.\n", buscar);
    printf("\nPresione una tecla para salir... ");
    getch();
    return;
}

printf("La edad de %s es %d\n", nombres[pos], edad[pos]);

// ===== CONFIRMAR CAMBIO (S/N) =====
do
{
    printf("Desea cambiar la informacion (S/N): ");
    scanf(" %c", &sn);
    if (sn >= 'a' && sn <= 'z') sn -= 32;
} while (sn != 'S' && sn != 'N');

if (sn == 'S')
{
    // Elegir qué cambiar
    do
    {
        printf("1. Nombre\n");
        printf("2. Edad\n");
        printf("Opcion: ");
        scanf("%d", &cambia);
    } while (cambia != 1 && cambia != 2);
}

```

```

if (cambia == 1)
{
    printf("Nuevo nombre: ");
    scanf("%s", nombres[pos]);
}
else
{
    printf("Nueva edad: ");
    scanf("%d", &edad[pos]);
}

// Reordenar alfabéticamente después del cambio
for (i = 0; i < TAM - 1; i++)
{
    for (j = i + 1; j < TAM; j++)
    {
        if (strcmp(nombres[i], nombres[j]) > 0)
        {
            strcpy(aux, nombres[i]);
            strcpy(nombres[i], nombres[j]);
            strcpy(nombres[j], aux);

            auxi = edad[i];
            edad[i] = edad[j];
            edad[j] = auxi;
        }
    }
}

// IMPRESIÓN FINAL (ORDENADA)
printf("\n");
printf("No.\tNombre\tEdad\n");
printf("--\t-----\t---\n");
for (i = 0; i < TAM; i++)
{
    printf("%d\t%s\t%d\n", (i + 1), nombres[i], edad[i]);
}
else //
{
    printf("No se realizaron cambios.\n");
}

printf("\nPresione una tecla para salir... ");
getch();
}

```

Link De GDB Online : <https://onlinedb.com/OdUlveLx8>

Pantallazos De Prueba

```
Persona UNO
Nombre: CAMILA
Edad: 10
Persona DOS
Nombre: JUAN
Edad: 10
Persona TRES
Nombre: PETRA
Edad: 12
Persona CUATRO
Nombre: OTTO
Edad: 10
Persona CINCO
Nombre: LEO
Edad: 10

No. Nombre Edad
1 CAMILA 10
2 JUAN 10
3 PETRA 12
4 OTTO 10
5 LEO 10

No. Nombre Edad
1 CAMILA 10
2 JUAN 10
3 LEO 20
4 OTTO 10
5 PETRA 12

Personas a buscar: LEO
La edad de LEO es 10
Desea cambiar la informacion (S/N): S
1. Nombre
2. Edad
3. Salir
2
Nueva edad: 20

No. Nombre Edad
1 CAMILA 10
2 JUAN 10
3 LEO 20
4 OTTO 10
5 PETRA 12
```

EJERCICIO 17

Tabla De Objetos

Objeto	Tipo	Descripción
numeros[5]	Entero (Arreglo)	Almacena el historial de intentos del jugador.
numeroSecreto	Entero	El número aleatorio generado (1 a 100).
maxIntentos	Entero	Límite de oportunidades definido por el usuario (1 a 5).
intentosRealizados	Entero	Contador de cuántas veces ha disparado el jugador.
i	Entero	Índice para recorrer el arreglo de historial.

Pseudocódigo

Algoritmo Juegoadivinanza

Numerosecreto <- Generar_aleatorio(1, 100)

Repetir

 Escribir "Cantidad De Intentos (1-5): "

 Leer Maxintentos

 Hasta Que Maxintentos >= 1 Y Maxintentos <= 5

 Para I <- 0 Hasta Maxintentos - 1 Hacer

 Intentosrealizados <- Intentosrealizados + 1

 Escribir "Intento ", I + 1, ":"

Leer Numeros[1]

```
Si Numeros[I] == NumeroSecreto Entonces
    Escribir "¡Correcto!"
    Romper Bucle
Sino Si Numeros[I] < NumeroSecreto Entonces
    Escribir "Demasiado Bajo"
Sino
    Escribir "Demasiado Alto"
Fin Si
Fin Para
```

```
// Resumen Final
Escribir "Número Secreto Era: ", NumeroSecreto
Para I <- 0 Hasta Intentosrealizados - 1 Hacer
    Escribir "Intento ", I + 1, ":", Numeros[I]
Fin Para
Fin
```

Código En Codeblox

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

int main() {
    int numeros[5];
    int numeroSecreto;
    int maxIntentos;
    int intentosRealizados = 0;
    int i;

    srand(time(NULL));
    numeroSecreto = rand() % 100 + 1;

    do {
        printf("Ingrese la cantidad de intentos (1 a 5): ");
        scanf("%d", &maxIntentos);

        if (maxIntentos < 1 || maxIntentos > 5) {
            printf("Error: El numero debe estar entre 1 y 5.\n");
        }
    } while (maxIntentos < 1 || maxIntentos > 5);
```

```

printf("\n¡Comienza el juego! Adivina el numero entre 1 y 100.\n\n");

// --- BUCLE DE JUEGO ---
for (i = 0; i < maxIntentos; i++) {
    intentosRealizados++;
    printf("Intento %d: ", i + 1);
    scanf("%d", &numeros[i]);

    if (numeros[i] == numeroSecreto) {
        printf(">> ¡Correcto, adivinaste el numero!\n");
        break;
    } else if (numeros[i] < numeroSecreto) {
        printf(">> Demasiado bajo.\n");
    } else {
        printf(">> Demasiado alto.\n");
    }
}

printf("\n=====\\n");
printf(" RESUMEN DE PARTIDA\\n");
printf("=====\\n");
printf("Numero secreto: %d\\n", numeroSecreto);

for (i = 0; i < intentosRealizados; i++) {
    printf("Intento %d: %d\\n", i + 1, numeros[i]);
}

if (numeros[intentosRealizados-1] != numeroSecreto) {
    printf("\\n¡Te quedaste sin intentos! Mas suerte la proxima.\n");
}

return 0;
}

```

Link De GDB Online : <https://onlinegdb.com/9PaR9nCJn>

Pantallazos De Prueba

```

main.c
50     printf("Intento %d: ", i + 1);
51     scanf("%d", &numeros[i]);
52
53     if (numeros[i] == numeroSecreto) {
54         printf(">> ¡Correcto, adivinaste el numero!\n");
55         break;
56     } else if (numeros[i] < numeroSecreto) {
57         printf(">> Demasiado bajo.\n");
58     } else {
59         printf(">> Demasiado alto.\n");
60     }
61 }

;Comienza el juego! Adivina el numero entre 1 y 100.

Intento 1: 12
>> Demasiado alto.
Intento 2: 5
>> Demasiado alto.

=====
RESUMEN DE PARTIDA
=====

Número secreto: 3
Intento 1: 12
Intento 2: 5

;Te quedaste sin intentos! Mas suerte la proxima.

```

EJERCICIO 18

Tabla De Objetos

Objeto	Tipo	Descripción
Intento	Estructura	Almacena el valor ingresado y una cadena con el resultado.
intentos[10]	Arreglo de Estructura	Vector que guarda el historial de hasta 10 intentos.
min, max	Entero	Definen los límites del rango de búsqueda.
numeroSecreto	Entero	Número aleatorio generado dentro del rango configurado.
usados	Entero	Contador de intentos válidos realizados.

Pseudocódigo

Estructura Intento

 Valor: Entero

 Resultado: Cadena

Fin Estructura

Algoritmo Juegoadivinanzaavanzado

// 1. Configuración Del Rango

 Escribir "Ingrese Mínimo Y Máximo: "

 Leer Min, Max

 Si Min >= Max Entonces Terminar Con Error

// 2. Preparar Secreto E Intentos

 NumeroSecreto <- Generar_aleatorio(Min, Max)

 Escribir "Máximo De Intentos (1-10): "

Leer Maxintentos

Usados <- 0

Gano <- Falso

// 3. Bucle Principal Del Juego (Rf03, Rf04)

Mientras Usados < Maxintentos Y Gano == Falso Hacer

 Escribir "Intento ", Usados + 1

 Leer Valor

// Validaciones (Rf03)

 Si Valor < Min O Valor > Max Entonces

 Escribir "Error: Fuera De Rango"

 Continuar Mientras

 Fin Si

 Si Esduplicado(Intentos, Usados, Valor) Entonces

 Escribir "Error: Ya Ingresaste Este Número"

 Continuar Mientras

 Fin Si

// Registro Y Lógica (Rf04, Rf05)

 Intentos[Usados].Valor <- Valor

 Si Valor == Numerosecreto Entonces

 Intentos[Usados].Resultado <- "Correcto"

 Gano <- Verdadero

 Sino Si Valor < Numerosecreto Entonces

 Intentos[Usados].Resultado <- "Menor Que El Secreto"

 Sino

 Intentos[Usados].Resultado <- "Mayor Que El Secreto"

 Fin Si

 Usados <- Usados + 1

Fin Mientras

// 4. Salida De Datos (Rf05)

 Llamar Imprimirresumen(Min, Max, Numerosecreto, Intentos, Usados)

Fin

Código En Codeblox

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

#define MAX_INTENTOS 10
#define TAM_TEXTO 30

typedef struct {
    int valor;
    char resultado[TAM_TEXTO]; // "Correcto", "Menor que el secreto", "Mayor que el
    secreto"
} Intento;

// --- Prototipos ---
int leerEntero(const char *mensaje);
int leerEnteroEnRango(const char *mensaje, int min, int max);
int esDuplicado(const Intento intentos[], int usados, int valor);
void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int
maxIntentos);

int main(void) {
    int min, max;
    int maxIntentos;
    int numeroSecreto;
    Intento intentos[MAX_INTENTOS];
    int usados = 0;

    // -----
    // RF01: Configuración del rango
    // -----
    printf("==> CONFIGURACION DEL RANGO ==>\n");
    min = leerEntero("Ingrese el minimo del rango: ");
    max = leerEntero("Ingrese el maximo del rango: ");

    if (min >= max) {
        printf("ERROR: El minimo debe ser menor que el maximo.\n");
        return 1;
    }

    // Inicializar RNG y secreto
    srand((unsigned)time(NULL));
    numeroSecreto = (rand() % (max - min + 1)) + min;

    // -----
    // RF02: Configuración de intentos
```

```

// -----
printf("\n==== CONFIGURACION DE INTENTOS ====\n");
maxIntentos = leerEnteroEnRango("Seleccione el maximo de intentos (1 a 10): ", 1,
MAX_INTENTOS);

printf("\n==== COMIENZA EL JUEGO ====\n");

// -----
// RF03: VALIDACION DE RANGO Y DUPLICADOS
// -----
while (usados < maxIntentos) {
    int valor;
    printf("\nIntento %d de %d. Ingrese un numero: ", usados + 1, maxIntentos);
    if (scanf("%d", &valor) != 1) {
        // Limpiar buffer en caso de entrada no numérica
        int c; while ((c = getchar()) != '\n' && c != EOF) {}
        printf("ERROR: Entrada no valida. Intente nuevamente.\n");
        continue;
    }

    // Validar rango (RF03)
    if (valor < min || valor > max) {
        printf("ERROR: Numero fuera del rango [%d - %d]\n", min, max);
        continue;
    }

    // Validar duplicado (RF03)
    if (esDuplicado(intentos, usados, valor)) {
        printf("ERROR: Ya ingresaste este numero antes.\n");
        continue;
    }
}

// -----
//RF04:REGISTRO DE INTENTO DE MATRIZ
// -----
intentos[usados].valor = valor;

// Calcular resultado textual para RF05
if (valor == numeroSecreto) {
    strncpy(intentos[usados].resultado, "Correcto", TAM_TEXTO - 1);
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';
} else if (valor < numeroSecreto) {
    strncpy(intentos[usados].resultado, "Menor que el secreto", TAM_TEXTO - 1);
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';
} else {
    strncpy(intentos[usados].resultado, "Mayor que el secreto", TAM_TEXTO - 1);
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';
}

```

```

if (valor == numeroSecreto) {
    printf("¡FELICIDADES! Adivinaste el numero secreto.\n");
    usados++; // se cuenta el intento ganador
    break;
} else if (valor < numeroSecreto) {
    printf("El numero secreto es MAYOR.\n");
} else {
    printf("El numero secreto es MENOR.\n");
}

usados++;
}

// -----
// RF05: Resumen detallado en tabla
// -----
imprimirResumen(min, max, numeroSecreto, intentos, usados, maxIntentos);

if (usados == maxIntentos && (usados == 0 || intentos[usados - 1].valor != numeroSecreto)) {
    printf("\nNo lograste adivinar el numero. ¡Sigue intentando!\n");
}

return 0;
}

int leerEntero(const char *mensaje) {
    int v;
    for (;;) {
        printf("%s", mensaje);
        if (scanf("%d", &v) == 1) {
            return v;
        }
        // limpiar buffer
        int c; while ((c = getchar()) != '\n' && c != EOF) {}
        printf("Entrada no valida. Intente nuevamente.\n");
    }
}

int leerEnteroEnRango(const char *mensaje, int min, int max) {
    int v;
    for (;;) {
        printf("%s", mensaje);
        if (scanf("%d", &v) == 1 && v >= min && v <= max) {
            return v;
        }
    }
}

```

```

int c; while ((c = getchar()) != '\n' && c != EOF) {}
printf("ERROR: Debe estar entre %d y %d.\n", min, max);
}
}

int esDuplicado(const Intento intentos[], int usados, int valor) {
for (int i = 0; i < usados; ++i) {
if (intentos[i].valor == valor) {
return 1;
}
}
return 0;
}

void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int maxIntentos) {
printf("\n=====\\n");
printf(" RESUMEN DETALLADO DEL JUEGO\\n");
printf("=====\\n");
printf("Rango elegido: [%d - %d]\\n", min, max);
printf("Numero secreto: %d\\n", secreto);
printf("Intentos permitidos: %d\\n", maxIntentos);
printf("Intentos realizados: %d\\n", usados);
printf("Intentos restantes: %d\\n\\n", maxIntentos - usados);

// Tabla
printf("-----\\n");
printf("%-10s %-15s %-20s\\n", "Intento", "Valor", "Resultado");
printf("-----\\n");
for (int i = 0; i < usados; ++i) {
printf("%-10d %-15d %-20s\\n", i + 1, intentos[i].valor, intentos[i].resultado);
}
printf("-----\\n");
}
}

```

Link De GDB Online : <https://onlinegdb.com/AoiP49PZ1G>

Pantallazos De Prueba

```

==> ./input
==== CONFIGURACION DEL RANGO ====
Ingrese el minimo del rango: 4
Ingrese el maximo del rango: 10
==== CONFIGURACION DE INTENTOS ====
Selecciona el maximo de intentos (1 a 10): 3
==== COMIENZA EL JUEGO ====
Intento 1 de 3. Ingrese un numero: 8
El numero secreto es MAYOR.
Intento 2 de 3. Ingrese un numero: 9
¡FELICIDADES! Adivinaste el numero secreto.

RESUMEN DETALLADO DEL JUEGO
=====
Rango elegido: [4 - 10]
Número secreto: 9
Intentos permitidos: 3
Intentos realizados: 2
Intentos restantes: 1

Intento     Valor      Resultado
1           8          Menor que el secreto
2           9          Correcto

...Program finished with exit code 0
Press ENTER to exit console.

```