

freddy Bekla Cedillo Soria

D 17 M 10 A 24

Scribe®

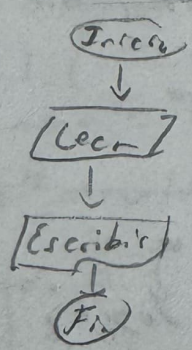
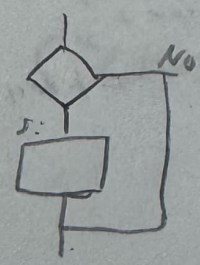
Nva Cuale = 322116362

Grupo = 22 fundamentos de programación
Semre 2

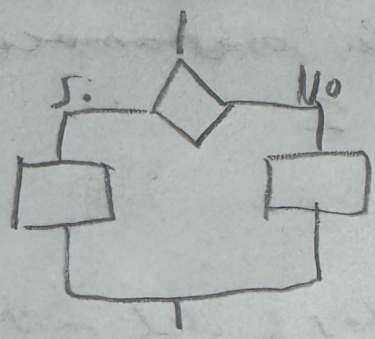
1: ¿Que es un arreglo?

Es una secuencia de datos del mismo tipo que ocupan un lugar contiguo en la memoria; el arreglo se pronuncia sucesivamente 0, 1, 2, 3, 4... (de 0 a n-1 donde n es tamaño del arreglo)
Su sintaxis es: tipo del elemento del Arreg
nombre del Arreg [num de elementos];

2: Completar

Estructura	Diagrama	Pseudocódigo	Codificación
Secuencial	 <pre> graph TD Inicio([Inicio]) --> Leer[Leer] Leer --> Escribir[Escribir] Escribir --> Fin([Fin]) </pre>	<p>Leer escribir</p>	<pre> } scanf("%d", &d); printf("%d", d); } </pre>
Condicional Simple		<p>Inicio Leer a, b Si a > b entonces Instrucciones Fin Si Fin</p>	<pre> if (a > b) { instrucciones } </pre>

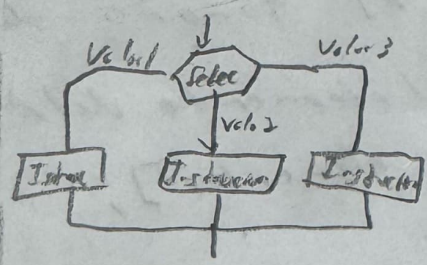
Condiciona
doble



Inicio
Leer gb
Si a > b entonces
Instrucciones 1
Si No
Instrucciones 2
Fin Si
Fin

IF (a > b)
{
 Instrucciones 1
}
else {
 Instrucciones 2
}
}

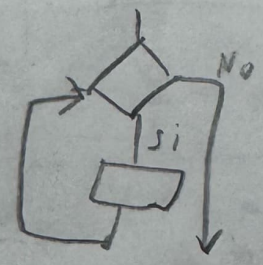
Condiciona
Múltiple



Inicio
Leer opción
Seguir opción Mas
Caso 1:
Instrucciones
Caso 2:
Instrucciones
Caso 3:
Instrucciones por
defecto
Fin Seguir
Fin

Switch (Variable)
{
 Case Valor 1
 Instrucciones
 Case Valor 2
 Instrucciones
 Case Valor 3
 Instrucciones
 default:
 Instrucciones
}

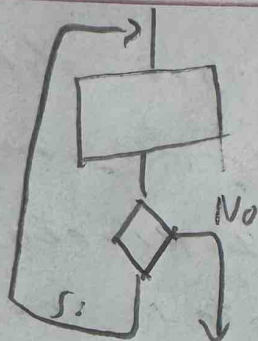
Iterativa
Mientras



Mientras que
Expresión lógica
Instrucciones
Fin del Mientras

while (expresión)
{
 Instrucciones
}

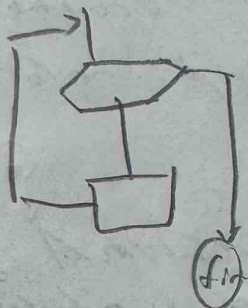
Iterativa
hacer Mientras



Hacer
Instrucción
Mientras Expresión
Logica)

do
{
Instrucción
}
while [Expresión];

Iterativa
Para



Para (valor inicial
condición incremento
decremento)
Instrucciones
Fin Para

for (valor. inicial;
condición; incremento
/decremento)
{
Instrucciones
}

3: Realiza un ejemplo de condicional doble utilizando el condicional ternario.

Condición ? Si Se cumple : Si No Se cumple

include <stdio.h>

int main () {

int numero = 10;

char *resultado;

resultado = (numero > 0) ? "Positivo" : (numero < 0) ?

"negativo" : "Cero";

printf ("El numero es : %s\n", resultado);

}

return 0;

}

4: Realizar el análisis (Requerir datos de entrada, salida, restricciones y procesos), diagrama de flujo o pseudocódigo, programa y pruebas de escritorio para los siguientes incisos:

a: Leer un arreglo de tipo carácter con los siguientes elementos e imprimirlo de forma inversa.

b: Leer dos vectores vector A y vector B de 8 elementos de tipo entero y obtener el vector C que almacene la suma del vector A + vector B e imprimir cada uno de los vectores al final

c: Realizar un programa que obtenga la siguiente numeración y la siguiente posición (utiliza estructura iterativa)

```
1
232
34543
4567654
567898765
```

a)

Análisis:

Entrada = Arreglo de 10 caracteres

Restricciones = Todos los elementos deben ser caracteres y la forma de impresión debe ser al revés.

Datos de Salida = Carácter a la inversa

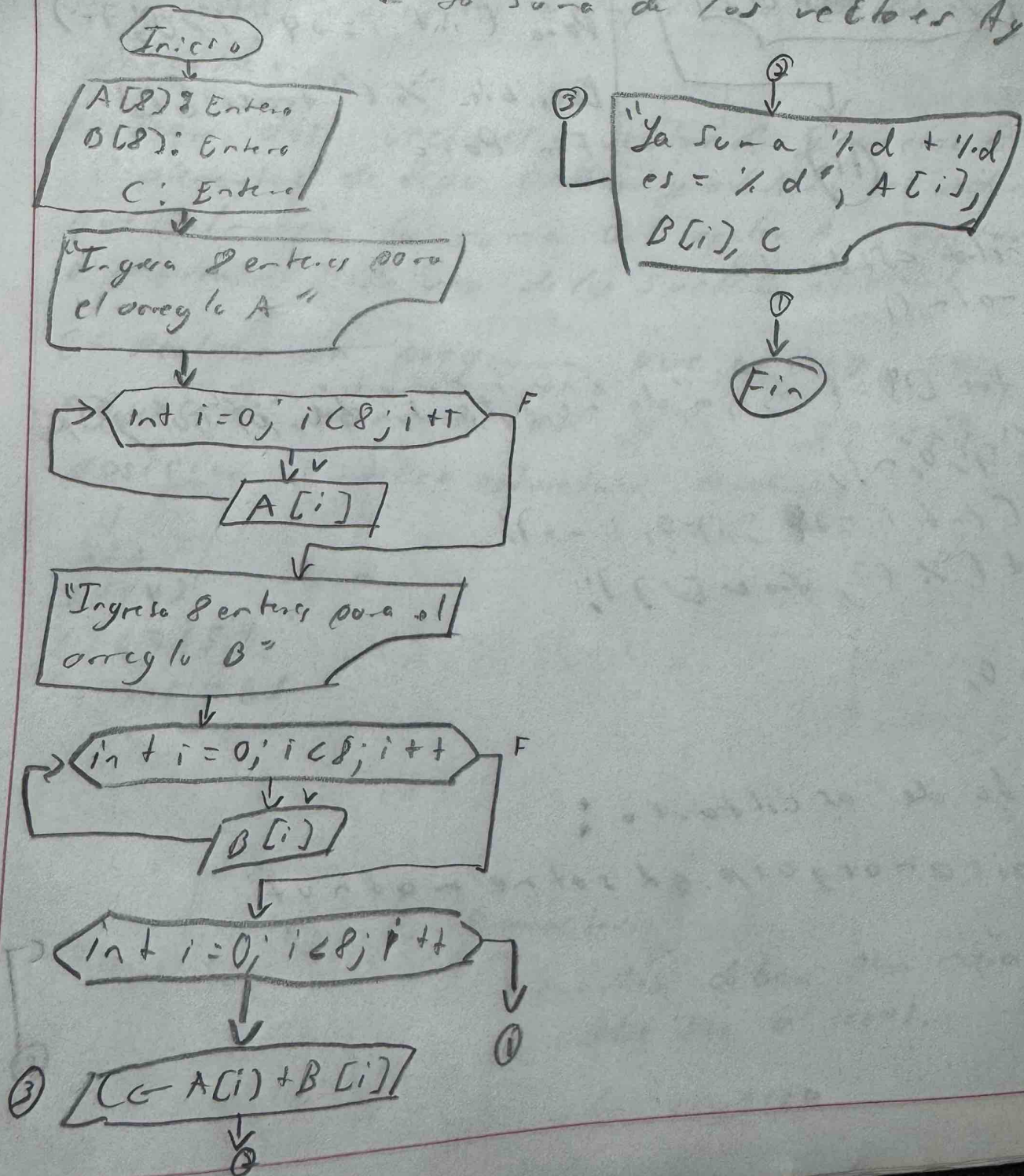
B)

Análisis:

Entrada = 8 enteros para el vector A y otros 8 enteros para el vector B

Restricción: Para cada vector deben ser 8 elementos enteros

Dato de Salida: La suma de los vectores A y B



Inicio

A[8]: Entero

B[8]: Entero

C: Entero

Imprimir ("Ingresar 8 enteros para el arreglo A")

Para (int i=0; i<8; i++)

Leer A[i]

Fin Para

Imprimir ("Ingresar 8 elementos para el arreglo B")

Para (int i=0; i<8; i++)

Leer B[i]

Fin Para

Para (int i=0; i<8; i++)

C := A[i] + B[i]

Imprimir ("La suma %d + %d es: %d", A[i], B[i], C)

Fin Para

Fin

#include <stdio.h>

int main()

{

int A[8];

int B[8];

int C;

printf ("Ingresar 8 enteros para el arreglo A");

for (int i=0; i<8; i++)

{

scanf ("%d", &A[i]);

```

}
printf ("Ingrese Partes para el arreglo D");
for (int i=0; i<D; i++)
{
    scanf ("%d", &D[i]);
}

for (int i=0; i<D; i++)
{
    C = A[i] + B[i];
    printf ("%s suma de %d y %d es: %d", A[i],
    B[i], C);
}
return 0;
}

```

Procesos de escritura.

i	A	B	C (Suma)
1	2	16	18
2	4	14	18
3	6	12	18
4	8	10	18
5	10	8	18
6	12	6	18
7	14	4	18
8	16	2	18

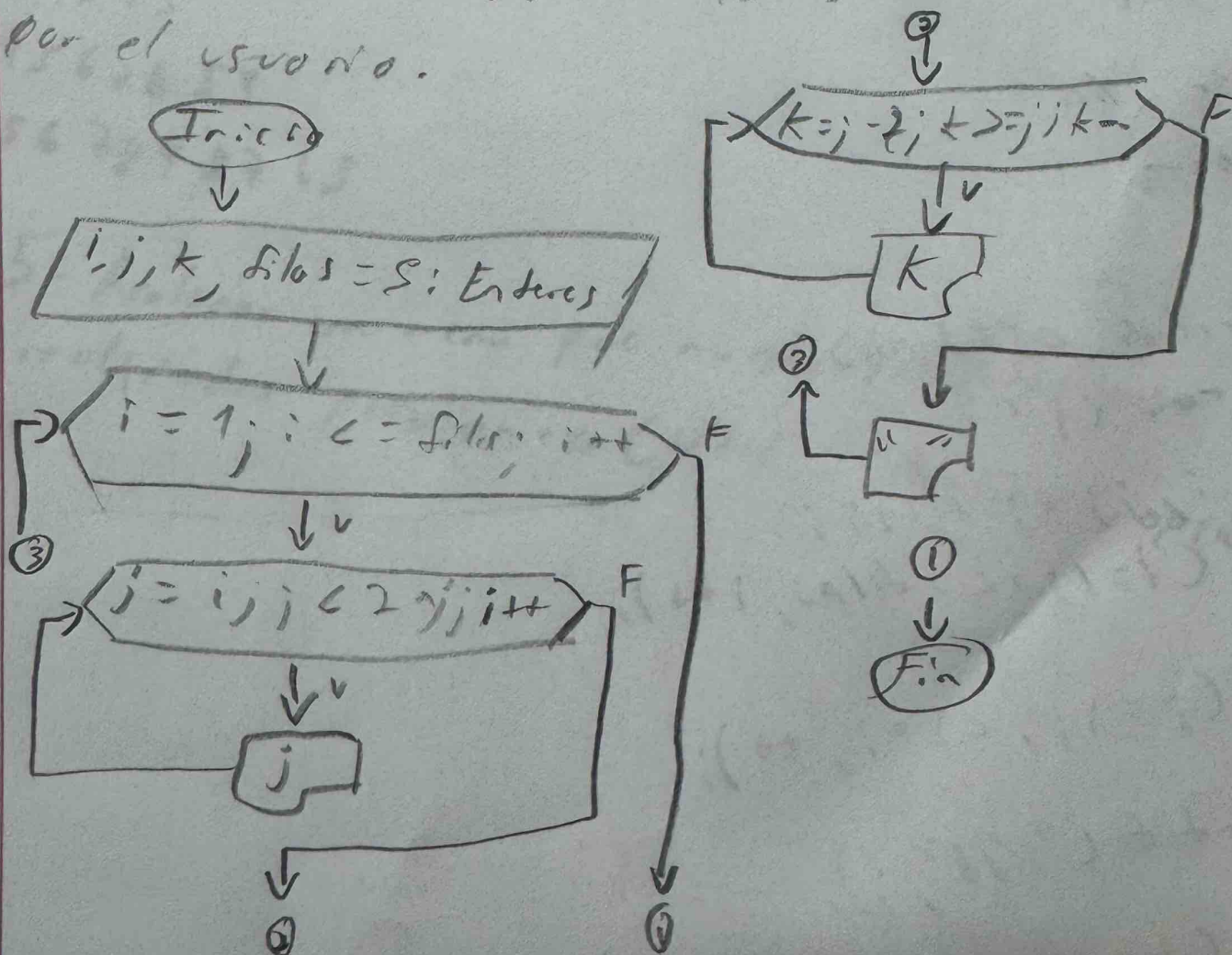
c)

Análisis

Entada = Tres variables, la primera sirve a control a los fillos, otra el ascenso de los números y la otra el descenso de estos.

Restricciones = Dentro de las interacciones hay restricciones que hacen por ser uno hasta un punto y se reste hasta otro.

Datos de Salida = Obtener la estructura deseada por el usuario.



Tarefa

$i, j, k, \text{filas} = 5$; Eater

Para ($i = 1; i \leq \text{filas}; i++$)

Para ($j = i; j \leq 2 * i; j++$)

Imprimir;

Fim Para

Para ($k = j - 2; k \geq j; k--$)

Imprimir k

Fim Para

Imprimir '\n'

Fim para

Fim

#Include <stdio.h>

int main ()

{
int i, j, k, filas;

for ($i = 1; i \leq \text{filas}; i++$);

{
for ($j = i; j \leq 2 * i; j++$);

{
printf ("%d", j);
}

for ($k = j - 2; k \geq j; k--$);


```

{
    Print &("u/d, k");
}
Print &("n");
}
Return 0;
}
    
```

Pueblos de escritores

1
 232
 34543
 4567654
 567898765

5: Realizar un menu que muestre los programas realizados previamente (Utiliza Funciones, opcional)


```

#include <stdio.h>

void invertirCadena();
void sumarVectores();
void piramideNumeros();

int main() {
    int opcion;

    do {
        printf("\n--- MENÚ ---\n");
        printf("1. Invertir cadena\n");
        printf("2. Sumar vectores\n");
        printf("3. Pirámide de números\n");
        printf("4. Salir\n");
        printf("Seleccione una opción: ");
        scanf("%d", &opcion);

        switch(opcion) {
            case 1: invertirCadena(); break;
            case 2: sumarVectores(); break;
            case 3: piramideNumeros(); break;
            case 4: printf("Saliendo...\n"); break;
            default: printf("Opción no válida\n"); break;
        }
    } while(opcion != 4);

    return 0;
}

void invertirCadena() {
    char cadena[] = "Fundamentos de programación";
    int longitud = strlen(cadena);

    printf("Cadena invertida: ");
    for (int i = longitud - 1; i >= 0; i--) {
        printf("%c", cadena[i]);
    }
    printf("\n");
}

void sumarVectores() {
    int vectorA[8] = {1, 2, 3, 4, 5, 6, 7, 8};
    int vectorB[8] = {8, 7, 6, 5, 4, 3, 2, 1};
    int vectorC[8];

    for (int i = 0; i < 8; i++) {
        vectorC[i] = vectorA[i] + vectorB[i];
    }

    printf("Vector A: ");
    for (int i = 0; i < 8; i++) {
        printf("%d ", vectorA[i]);
    }
    printf("\n");

    printf("Vector B: ");
    for (int i = 0; i < 8; i++) {
        printf("%d ", vectorB[i]);
    }
    printf("\n");
}

```

```

    }
    printf("\n");

    printf("Vector B: ");
    for (int i = 0; i < 8; i++) {
        printf("%d ", vectorB[i]);
    }
    printf("\n");

    printf("Vector C: ");
    for (int i = 0; i < 8; i++) {
        printf("%d ", vectorC[i]);
    }
    printf("\n");
}

void piramideNumeros() {
    int filas = 5;
    for (int i = 1; i <= filas; i++) {
        for (int j = 0; j < i; j++) {
            printf("%d", i + j);
        }
        for (int j = i - 2; j >= 0; j--) {
            printf("%d", i + j);
        }
        printf("\n");
    }
}

```



```

main.c: In function 'invertirCadena':
main.c:33:20: warning: implicit declaration of function 'strlen' [-Wimplicit
-function-declaration]
    33 |         int longitud = strlen(cadena);
        |                        ^~~~~~
main.c:2:1: note: include '<string.h>' or provide a declaration of 'strlen'
    1 | #include <stdio.h>
    ++ |+#include <string.h>
    2 |
main.c:33:20: warning: incompatible implicit declaration of built-in functio
n 'strlen' [-Wbuiltin-declaration-mismatch]
    33 |         int longitud = strlen(cadena);
        |                        ^~~~~~
main.c:33:20: note: include '<string.h>' or provide a declaration of 'strlen'
,

--- MENÚ ---
1. Invertir cadena
2. Sumar vectores
3. Pirámide de números
4. Salir
Seleccione una opción: 3
1
232
34543
4567654
567898765

--- MENÚ ---
1. Invertir cadena
2. Sumar vectores
3. Pirámide de números
4. Salir
Seleccione una opción: 2
Vector A: 1 2 3 4 5 6 7 8
Vector B: 8 7 6 5 4 3 2 1
Vector C: 9 9 9 9 9 9 9 9

--- MENÚ ---
1. Invertir cadena
2. Sumar vectores
3. Pirámide de números
4. Salir
Seleccione una opción: 4
Saliendo...

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

```