

Programación Digital 10

- Estructuras de Repetición

Prof. Lisbeth Pérez Rivas
lisbethpe@ula.ve

Programación Estructurada

- Enfoque disciplinado que permite escribir programas estructurados, utilizando las siguientes tres estructuras de control.
- Secuencial (asignación, lectura, escritura)
- Decisión (simple, doble, múltiple)
- Repetición (Repita mientras, repita hasta, repita para)

Estructuras de Repetición

- Son aquellas que permiten que un conjunto de sentencias (una o varias) se ejecuten repetidamente.
- Bucle: Estructura de control que permite la repetición de una serie determinada de sentencias. El cuerpo del bucle lo constituyen la serie de sentencias que pueden ser de cualquier tipo (secuenciales, de decisión o repetición) las cuales serán repetidas n veces, pudiendo ser n conocido o desconocido. Si n es desconocido, el número de veces que debe repetirse el cuerpo del bucle estará condicionado por una expresión lógica.

Bucle

- Sentencia 1
- Sentencia 2
- .
- .
- .
- Sentencia r



Cuerpo del Bucle

Conceptos Básicos

- Contador: Variable cuyo valor se incrementa o decrementa en una cantidad constante cada vez que se produce un determinado suceso o acción.
- Operaciones sobre un contador.
 - Inicialización: Nombre del contador= valor inicial;
 - Incremento/decremento:

Nombre del contador=Nombre del contador + constante;

Nombre del contador=Nombre del contador - constante;

Ejemplos:

conta=0;	→	Inicialización
conta= conta + 1;	}	Incremento
conta= conta + 3;		
conta= conta - 2;	}	Decremento
conta= conta - 10;		

Conceptos Básicos

- Acumulador: Variable cuyo valor se incrementa o decrementa en una cantidad variable cada vez que se produce un determinado suceso o acción.
- Operaciones sobre un acumulador.
 - Inicialización: Nombre del acumulador= valor inicial;
 - Acumulación:

Nombre del acumulador=Nombre del acumulador + valor;

Nombre del acumulador=Nombre del acumulador - valor;

Nombre del acumulador=Nombre del acumulador * valor;

Nombre del acumulador=Nombre del acumulador / valor;

Tipos de Estructuras de Repetición

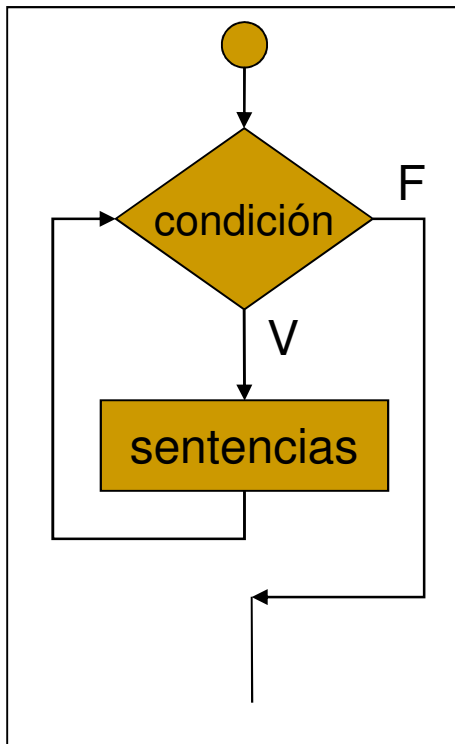
- Repita mientras
- Repita Hasta
- Repita Para

Repita mientras

- Utilizada cuando no se conoce exactamente el número de iteraciones. El cuerpo del bucle se repite mientras se cumple una condición, es decir, se repite mientras el resultado de la evaluación de la expresión lógica es verdadero.
- La condición es evaluada al principio de cada iteración, por tanto, si la evaluación es falsa desde el comienzo de la estructura, el bucle no se ejecuta nunca. Si la condición nunca se hace falsa, el programa entra en un lazo infinito, es decir, las sentencias del lazo se ejecutan indefinidamente.

Repita mientras

Diagrama de Flujo



Algoritmo

Mientras (condición lógica)
 Sentencias
Fin_RM

También se puede escribir como

Repita mientras (condición Lógica)
 Sentencias
Fin_RM

Código

```
while (condición lógica)  
  Sentencia;  
  
while (condición lógica) {  
  sentencia 1;  
  sentencia 2;  
  .  
  .  
  .  
  sentencia N;  
}
```

Aspectos importantes

- Las variables que intervienen en la condición lógica deben inicializarse antes de ejecutar el lazo por primera vez, ya que lo primero que hace la estructura es evaluar la condición.
- Dentro del lazo, se debe modificar los valores de las variables que intervienen en la condición lógica, para garantizar que en algún momento se haga falsa y el lazo pueda terminar su ejecución y así garantizar que el lazo no es infinito.

Ejemplo 1

- Escribir por pantalla los números del 1 al 10.
- Análisis E-P-S.

Entrada	Proceso	Salida
Ninguna	Mientras un contador sea menor o igual que 10 Se escribe el valor del contador	Los números del 1 al 10 Contador (entero)

Ejemplo 1

Algoritmo

```
1.Inicio
2.contador = 1
3.Mientras (contador<=10)
    3.1 Escribir (contador)
    3.2 contador = contador+1
    Fin_RM
4.Fin
```

Código

```
#include<stdio.h>
int main(){
    int contador=1;
    while(contador<=10){
        printf("%d\n",contador);
        contador++;
    }
    return 0;
}
```

Se pueden utilizar diferentes condiciones lógicas, teniendo en cuenta el valor de inicio del contador y lugar de modificación del mismo dentro del lazo

Algoritmo

```
1.Inicio
2.contador = 0
3.Mientras( contador<10)
    3.1 contador = contador+1
    3.2 Escribir (contador)
    Fin_RM
4.Fin
```

```
#include<stdio.h>
int main(){
    int contador=0;
    while(contador<10){
        contador++;
        printf("%d\n",contador);
    }
    return 0;
}
```

Ejemplo 2

- Escribir por pantalla los números del 1 a N.
- Análisis E-P-S.

Entrada	Proceso	Salida
Extensión de la Secuencial (N) - entero	Mientras un contador sea menor o igual que N Se escribe el valor del contador	Los números del 1 al N Contador (entero)

Algoritmo

- 1.Inicio
- 2.Escribir('numeros a escribir')
- 3.Leer (N)
- 4.contador = 0
- 5.Mientras (contador<N)
 - 5.1 contador = contador+1
 - 5.2 Escribir (contador)
- Fin_RM
- 6.Fin

Código

```
#include<stdio.h>
int main(){
    int N,contador=0;
    printf("numeros a escribir");
    scanf("%d",&N);
    while(contador<N){
        contador++;
        printf("%d\n",contador);
    }
    return 0;
}
```

Ejemplo 3

- Sumar 5 números.
- Análisis E-P-S

- Entrada Proceso Salida

Los números a
sumar
num1,num2,num3,
num4,num5
enteros

$suma=num1+num2+num3+num4+num5$

El resultado de la
suma
suma(entero)

Ejemplo 3

Algoritmo

```
1.Inicio
2.contador = 0
3.suma=0
4.Mientras (contador<5)
    4.1  Escribir ('numero a sumar')
    4.2  leer(num)
    4.3  suma=suma+num
    4.4  contador=contador+1
    Fin_RM
5.Escribir('resultado',suma)
6.Fin
```

Código

```
#include<stdio.h>
int main(){
    int suma=0,num,contador=0;

    while(contador<5){
        printf("numero a sumar");
        scanf("%d",&num);
        suma=suma+num;
        contador++;
    }
    printf("\nResultado: %d\n",suma);
    return 0;
}
```

Extendiendo a n números

Algoritmo

- 1.Inicio
- 2.Escribir('Cantidad de numeros a sumar')
- 3.Leer(n)
- 4.contador = 0
- 5.suma=0
- 6.Mientras (contador<n)
 - 6.1 Escribir ('numero a sumar')
 - 6.2 leer(num)
 - 6.3 suma=suma+num
 - 6.4 contador=contador+1
- Fin_mientras
- 7.Escribir('resultado',suma)
- 8.Fin

Código

```
#include<stdio.h>
int main(){
    int n,suma=0,num,contador=0;
    printf("Cantidad de numeros a sumar ");
    scanf("%d",&n);
    while(contador<n){
        printf("numero a sumar");
        scanf("%d",&num);
        suma=suma+num;
        contador++;
    }
    printf("\nResultado: %d\n",suma);
    return 0;
}
```

Ejemplo 3

- Leer una serie de números reales y calcular su media. La marca de fin de lectura será el -999.
- Análisis E-P-S

Entrada	Proceso	Salida
num ₁ , num ₂ , num ₃ ...num _n	Acumular la suma de los números e incrementar un contador mientras el número leído sea diferente de -999.	La media de los números ingresados media → real
Números Reales	suma=num ₁ , num ₂ , num ₃ ...num _n contador=contador+1; media=suma/contador	

Ejemplo 3

Algoritmo

- 1.Inicio
- 2.contador = 0
- 3.suma=0
- 4.Escribir('inserte numero')
- 5.Leer(num)
- 6.Mientras (num≠-999)
 - 6.1 suma=suma+num;
 - 6.2 contador=contador+1;
 - 6.3 Escribir ('inserte numero')
 - 6.4 leer(num)
- Fin_RM
- 7.Escribir('resultado',suma/contador)
- 8.Fin

Código

```
#include<stdio.h>
int main(){
    int contador=0;
    float suma=0,num,media;
    printf("Inserte numero ");
    scanf("%f",&num);
    while(num!=-999){
        suma=suma+num;
        contador++;
        printf("Inserte numero");
        scanf("%f",&num);
    }
    media=suma/contador;
    printf("\nResultado: %f\n",media);
    return 0;
}
```

Ejercicios

- Calcular la sumatoria de los 100 primeros números naturales
- Calcular la suma de los números pares y la suma de los impares comprendidos entre 1 y n.
- Escribir los enteros positivos menores que 100 omitiendo los divisibles por 7.
- Imprimir la tabla de multiplicar de un número dado. El funcionamiento del programa se muestra como sigue.

Introduzca un número: 2

La tabla de multiplicar del 2 es:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

....

$$2 \times 10 = 20$$

Ejercicios

- Leer valores que representan años e indique si son o no años bisiestos. El programa seguirá leyendo años hasta un máximo de 5 o hasta que haya leído 3 años bisiestos.

“Un año es bisiesto si es divisible por 400, o si es divisible por 4 pero no por 100”

El año 2000 es bisiesto (divisible por 400)

El año 1992 es bisiesto (divisible por 4 y no por 100)

El año 2100 no es bisiesto (divisible por 4 y también por 100)

Ejercicios

- Calcular el factorial de un número entero.
- Calcular la suma de la serie $1/1+1/2+\dots+1/N$. Donde N es un número entero que se determina con la condición que $1/N$ sea menor que un epsilon prefijado (por ejemplo $\epsilon=10e^{-6}$).
- Considerar el siguiente código

```
#include<stdio.h>
int main(){

    int i,n=?;
    i=1;
    while(i<=n)
        if(i%n==0)
            i++;
    printf("%d",i);
    return 0;
}
```

Cual es la salida si:

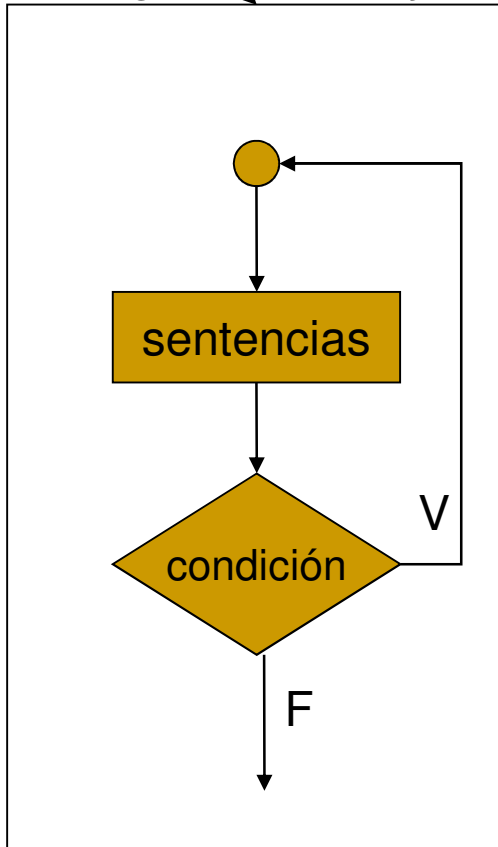
1. n es 0
2. n es 1
3. n es 3

Hacer mientras

- Se utiliza cuando no se conoce el número de iteraciones, mediante esta estructura el cuerpo del bucle se repite hasta que la condición se hace verdadera. La condición se evalúa después de que el cuerpo del bucle se ha ejecutado al menos una vez.
- Si la condición nunca se hace verdadera, el programa entra en un lazo infinito, es decir, las sentencias del cuerpo del lazo se ejecutarán indefinidamente.

Hacer mientras

Diagrama de Flujo



Algoritmo

Hacer
Sentencias
Mientras (condición lógica)

Código

```
do{  
    Sentencia;  
}while(condicion);  
  
do{  
    sentencia 1;  
    sentencia 2;  
    .  
    .  
    .  
    sentencia N;  
}while(condicion);
```

Ejemplo 1

- Sumar 5 números.
- Análisis E-P-S

- Entrada

Proceso

Salida

Los números a
sumar (enteros)
n1,n2,n3,n4,n5

$suma=n1+n2+n3+n4+n5$

El resultado de la
suma
suma(entero)

Ejemplo 1

Algoritmo

1. Inicio
2. contador = 0
3. suma=0
4. Hacer
 - 4.1 Escribir ('numero a sumar')
 - 4.2 leer(num)
 - 4.3 suma=suma+num
 - 4.4 contador=contador+1Mientras(contador<5)
5. Escribir('resultado', suma)
6. Fin

Código

```
#include<stdio.h>
int main(){
    int suma=0,num,contador=0;

    do{
        printf("numero a sumar");
        scanf("%d",&num);
        suma=suma+num;
        contador++;
    }while(contador<5);
    printf("\nResultado: %d\n",suma);
    return 0;
}
```

Ejemplo 2

- Leer una serie de números reales y calcular su media. Se le preguntará al usuario si desea seguir ingresando números.
- Análisis E-P-S
- Entrada Proceso Salida

$num_1, num_2,$
 $num_3 \dots num_n$

Acumular la suma de los números e incrementar un contador mientras el usuario lo desee.

media (real)

Reales

Opción de continuar

S o N, opcion (character)

$suma = num_1 + num_2 + num_3 +$
 $\dots + num_n$
 $contador = contador + 1;$

$media = suma / contador$

Ejemplo 2

Algoritmo

1. Inicio
2. contador = 0
3. suma=0
4. Hacer
 - 4.1 Escribir ('inserte numero')
 - 4.2 leer(num)
 - 4.3 suma=suma+num;
 - 4.4 contador=contador+1;
 - 4.5 Escribir ('desea continuar? S/N')
 - 4.6 Leer(opcion)
- Mientras(opcion!='N')
5. media=suma/contador
6. Escribir('resultado',media)
7. Fin

Código

```
#include<stdio.h>
int main(){
int contador=0;
float suma=0,num,media;
char opcion;
do{
    printf("Inserte numero");
    scanf("%f",&num);
    getchar();
    suma=suma+num;
    contador++;
    printf("Desea continuar? S/N");
    scanf("%c",&opcion);

}while(opcion!='N');
media=suma/contador;
printf("resultado %f",media);
return 0;
}
```

Ejercicios

- Dados 10 números enteros que se introducirán por teclado, calcular la suma de los números pares, cuántos números pares existen y la media aritmética de los números impares.
- Calcular la suma $1+2+3+\dots+n$, donde n es un valor dado. Validar que $n \geq 1$.
- Calcular la suma de los cuadrados de los cien primeros números naturales.

Repita para

- En la ejecución de esta estructura participan tres elementos.
 - Variable de control (V)
 - Valor inicial (V_i)
 - Valor final (V_f)

Existen dos formas de utilizar esta estructura.

- a) Con incremento del contador
- b) Con decremento del contador

Ejemplo 1

- Sumar todos los enteros pares desde 2 hasta 100
- Análisis E-P-S
- Entrada Proceso Salida

No hay

Para cada número par
entre el 2 y el 100
sumar su valor

$$\text{Acu}=2+4+6+\dots+100$$

Resultado de
la suma
Acu (entero)

Ejemplo 1

Algoritmo

- 1. Inicio
- 2. acu=0
- 3. Para (i=1;i<=50;i=i+1)
 - 3.1 acu=acu+2*i;
 - Fin_RP
- 4. Escribir('Suma de numeros pares',acu)
- 5. Fin

Código

```
#include<stdio.h>
int main(){
int i,acu=0;
for(i=1;i<=50;i++)
    acum=acu+2*i;
printf("Suma de numeros pares %d",acu);
return 0;
}
```

Ejemplo 2

- Calcular el seno para los valores $0.0 \leq x \leq 1.6$ con incrementos de 0.1.
- Análisis E-P-S

Entrada	Proceso	Salida
No hay	Con incrementos de 0.1 Calcular $\text{sen}(x)$	Valor del $\text{sen}(x)$ Real

Ejemplo 2

Algoritmo

- 1, Inicio
- 2. Para(x=0;x<=1.6;x=x+0.1)
 - 2.1 Escribir('seno de',x,
'es:', sin(x))
 - Fin_RP
- 3.Fin

Código

```
#include<stdio.h>
#include<math.h>
int main(){
float x;

    for(x=0.0;x<=1.6;x=x+0.1)
        printf("\nSeno de %f es: %f",x,sin(x));

    return 0;
}
```

Ejercicios

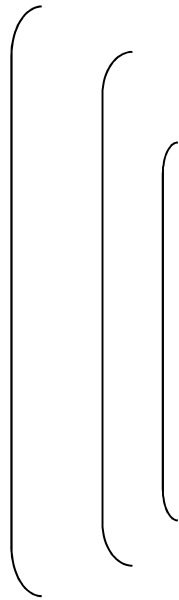
- Dadas las notas de n estudiantes de las materias Estadística, Cálculo 20 y Física 11. Calcular la media de cada estudiante.
- En el programa anterior calcular además la nota media de cada una de las materias.
- Calcular el factorial de un número.
- Calcular la conversión de grados Celsius a grados Fahrenheit comenzando desde $5\text{ }^{\circ}\text{C}$ decrementando hasta $-5\text{ }^{\circ}\text{C}$.
- Imprimir la suma de los cuadrados de los 100 primeros números naturales.
- Dado un valor $n > 1$, calcular la suma de la serie. $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$.
- Dado un valor $n > 1$, calcular la suma de la serie. $1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{9} + \dots + \frac{1}{n}$.

Estructuras de Repetición Anidadas

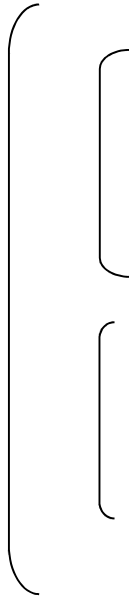
- Las estructuras de repetición se encuentran unas dentro de otras. Los bucles anidados constan de un bucle externo con uno o más bucles internos, cada vez que se repite el bucle externo, los bucles internos se repiten, se vuelven a evaluar los componentes de control y se ejecutan todas las iteraciones requeridas.
- Reglas de Anidamiento:
 - Las estructuras internas deben estar completamente contenidas en la estructura externa.
 - No debe existir solapamiento.

Estructuras de Repetición Anidadas

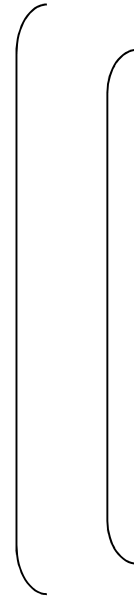
•



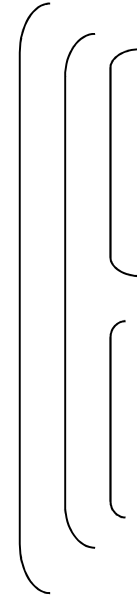
a



b



c



d

Estructuras anidadas

```
for (x = 1; x <= Xultimo; x++)
```

```
    for (y = 1; y <= Yultimo; y++)
```

```
    {
```

```
        int producto;
```

```
        producto = x * y;
```

```
        printf("  %d * %d = %d\n", x,y,producto);
```

bucle externo

bucle interno

Indique cual es la salida del siguiente código

```
#include<stdio.h>
#include<math.h>
int main(){
int x,y;

for(x=0;x<=3;x++){
    for(y=0;y<=2;y++)
        printf("*");
    printf("\n");
}

    return 0;
}
```


Indique cual es la salida del siguiente código

```
#include<stdio.h>
#include<math.h>
int main(){
int x,y;

for(x=0;x<=3;x++)
    for(y=0;y<=2;y++)
        printf("*");
    printf("\n");

    return 0;
}
```

Indique cual es la salida del siguiente código

```
#include<stdio.h>
#include<math.h>
int main(){
int x,y;

for(x=0;x<=3;x++){
    for(y=0;y<=x;y++)
        printf("*");
    printf("\n");
}

    return 0;
}
```

Ejercicios

- Se conoce la población de cada una de las cinco ciudades mas importantes de veintiocho países y se desea identificar e imprimir la población de la ciudad mas grande (en número de habitantes) de cada país.
- Calcular el factorial de n números dados.
- Dibujar un triángulo por pantalla.
- Encuentre el número de puntos con coordenadas enteras que están dentro de la elipse $2x^2+3y^2=100$

Realizar:

- Análisis E-P-S
- Algoritmo
- Código