

PRÁCTICA 5. ESTRUCTURAS DE REPETICIÓN

☑ OBJETIVOS:

- Familiarizarse con el empleo de la sentencia `for`
- Hacer uso de los contadores y acumuladores dentro de los ciclos.
- Determinar la necesidad del empleo del operador condicional.
- Aprender a generar condiciones adecuadas para limitar la acción de ciclos.
- Familiarizarse con el empleo de las sentencias `while` y `do_while`
- Emplear la sentencia `while` para validar errores

✂ EQUIPO:

- 💻 Computadora
- 💿 Disco Flexible o algún otro medio para almacenar sus prácticas
- 🖨 Turbo C (TC)

📖 GENERALIDADES

↳ ESTRUCTURAS DE CONTROL REPETITIVAS

Las estructuras de control repetitivas realizan 'n' veces una operación ó bloque de sentencias, conociendo con anticipación el número de iteraciones a realizar.

La estructura de control repetitiva `for` realiza 'n' veces una instrucción de manera fija como un contador.

```
for(variable_control=valor_inicial;condición;expresión_incremento) {  
    bloque de sentencias;  
}
```

La estructura de control repetitiva `while` el número de iteraciones no se conoce por anticipado y las sentencias se repiten MIENTRAS se cumple determinada condición.

```
while (condición) {  
    bloque de sentencias;  
}
```

La estructura de control repetitiva `do_while` es una variación de la estructura `while`, el operador `while` evalúa la condición al principio de cada iteración. El operador `do` efectúa un bloque de sentencias hasta que la condición se haga verdadera, y por lo menos lo realizará una vez. Su función se caracteriza por un REPITE-HASTA.

```
do {  
    bloque de sentencias;  
} while (condición);
```

ACTIVIDAD #1 . CICLOS `for`

1. Calcular la suma de los cuadrados de los primeros 100 enteros.
2. Calcular independientemente la suma de los números pares e impares comprendidos entre 1 y 200.
3. Escribir un programa en el cual imprima la tabla de Código ASCII. Debe de ser desplegada en número hexadecimal asociado al caracter y el caracter.
4. Escriba un programa que calcule y tabule los valores de x y de la función de muestreo $Sa(x) = \frac{\sin(x)}{x}$, para $x = \pm 3\pi$. Hacer que x se incremente en $\frac{\pi}{4}$ unidades.
5. Escriba un programa que calcule y tabule los valores de la función $f(x,y) = \frac{x^2 - y^2}{x^2 + y^2}$ para $x = 2, 4, 6, 8$ e $y = 6, 9, 12, 15, 18, 21$.

6. Escribir un programa que genere la siguiente serie:

2	19
4	17
6	15
⋮	⋮
20	1

7. Escribir un programa que genere la siguiente serie 1, 1, 1, 3, 5, 9, 17, 31...
8. Sumar 10 números introducidos por el usuario.
9. Dados dos números enteros calcular el cociente y el residuo de dichos números (realizarlo por medio de restas sucesivas).
10. Calcular el factorial de un número dado por el usuario.
11. Imprimir los números impares menores que el indicado por el usuario.
12. Calcular la suma de n números introducidos por el usuario.
13. Hacer un programa que genere la tabla de caracteres ASCII
14. Hacer un programa que determine si un número es "perfecto", "deficiente" o "excedente". De acuerdo a los siguientes criterios.

El número es PERFECTO si la suma de sus divisores es igual a el mismo.

El número es DEFICIENTE si la suma de sus divisores es menor que el.

El número es EXCEDENTE si la suma de sus divisores es mayor que el.

15. Generar las tablas de multiplicar del 1 al 10 (mostrar en forma tabular)
16. Elevar un número a cualquier potencia por medio de ciclos sin utilizar la función matemática `pow()`.

ACTIVIDAD #2 . CICLOS `while`

1. Lea una serie de números distintos de cero y obtener su media aritmética.
2. Crear un programa donde pida el monto total a pagar y la cantidad de dinero dado; desplegar la cantidad mínima de monedas (25¢, 10¢, 5¢ y 1¢) a dar de cambio.
3. Escribir un programa que estime el valor de la constante matemática e utilizando la fórmula

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

4. Escribir un programa que lea la base y la altura de un triángulo e imprima el área del triángulo. Validar para evitar errores de introducción de números negativos. Si se encuentra algunos, deberá de imprimirse junto con un mensaje que lo identifique con la base o la altura.
5. Un corredor de maratón (distancia 42.195 Km) Para calcular el tiempo medio en minutos por kilómetro, el juez debe de introducir el tiempo de carrera en horas con minutos. Validar para tiempos incorrectos tales como 0 o negativos, en caso de ser así pedir tiempo correcto hasta que este se introduzca bien.
6. Transformar el programa para calcular el factorial de un número para validar que sólo se calcule el factorial de números positivos.
7. Validar para que al introducir la fecha de manera numérica (dd/mm/aaaa), los días de ésta corresponda a mes del calendario.
8. Mediante la instrucción `rand()` genere un numero aleatorio [0-500] y pida al usuario que adivine cual es. Cada vez que el usuario de un número el programa deberá indicar si el número a adivinar es mayor o menor que número proporcionado por el usuario. Cuando el usuario adivine el programa mostrará en cuantos intentos lo hizo. Utilice ciclo `do-while`
9. Utilizando el ciclo `while` hacer un programa que dibuje un marco con el caracter que el usuario elija de un menú que su programa mostrará. Las coordenadas de inicio y fin deberán ser indicadas por el usuario.

Menú

- 1) @
- 2) #
- 3) *
- 4) Salir

El programa deberá permanecer en el menú hasta que el usuario seleccione la opción de salida.

10. Hacer un programa que pida la estatura de n personas y calcule el promedio. Se pedirán estaturas mientras no se introduzca una estatura de 0.