
Estructuras de Control Iterativas

```
while (expresión lógica){
```

```
    Sentencia1;
```

```
    Sentencia2;
```

```
    Sentencia3;
```

```
    .....
```

```
}
```

```
do{
```

```
    Sentencia1;
```

```
    Sentencia2;
```

```
    Sentencia3;
```

```
    .....
```

```
}while (expresión lógica);
```

```
for ( inicialización ; expresión lógica ; actualización){
```

```
    Sentencia1;
```

```
    Sentencia2;
```

```
    Sentencia3;
```

```
    .....
```

```
}
```

Ejercicios aplicativos:

1) “La conjetura de Ulam” nos dice que si aplicamos a cualquier numero natural, las siguientes operaciones:

- Si el número es par, se divide entre 2.
- Si el número es impar, se multiplica por 3 y se suma 1.

de forma iterativa, obtendremos una secuencia de numeros que siempre terminará en 1.

Ejm: Para $n=13$ se obtiene:

13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Realize usted un programa que imprima la secuencia de Ulam para cualquier numero natural leído por teclado.

2) Dado un numero en sistema decimal, convertirlo en binario y lo vuelva a convertir a decimal.

Solución:

Funciones

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

```
tipo NOMBRE (tipo argumento1,tipo argumento2){
```

```
    Sentencia1;
```

```
    Sentencia2;
```

```
    Sentencia3;
```

```
    return resultado;
```

```
}
```

```
void main(){
```

```
    tipo variable1;
```

```
    tipo variable2;
```

```
    tipo almacena;
```

```
    Sentencias;
```

```
// llamada a funcion:
```

```
    almacena = NOMBRE (variable1,variable2);
```

```
    Sentencias;
```

```
}
```

PASO POR VALOR :

```
tipo NOMBRE (tipo argumento1,tipo argumento2){
    Sentencia1;
    Sentencia2;
    Sentencia3;
    // Si en un momento dado las variables argumento1 y argumento2 cambian
    // este cambio no afecta directamente a las variables variable1 y variable2 que
    // se encuentran en la funcion main().
    .....
    return resultado;
}
```

PASO POR REFERENCIA :

```
tipo NOMBRE (tipo *argumento1,tipo *argumento2){
    Sentencia1;
    Sentencia2;
    Sentencia3;
    // En este caso si podemos modificar directamente los valores de las variables
    // variable1 y variable2 que se encuentran en la funcion main().
    // Solo es necesario modificar *argumento1 o *argumento2.
    return resultado;
}
```

```
void main(){

    tipo variable1;
    tipo variable2;
    tipo almacena;
    Sentencias;
    // llamada a funcion por valor:
    almacena = NOMBRE (variable1,variable2);
    // llamada a funcion por referencia:
    almacena = NOMBRE (&variable1,&variable2);
    Sentencias;
}
```

Ejercicios aplicativos:

- 1) El valor de la función e^x puede aproximarse con el desarrollo de Taylor:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Diseña una función que aproxime el valor de e^x usando n términos del desarrollo de Taylor, siendo n un número entero positivo. (Puedes usar, si te conviene, la función de exponenciación del último ejercicio para calcular los distintos valores de x^i , aunque hay formas más eficientes de calcular $x/1!$, $x^2/2!$, $x^3/3!$, ..., ¿sabes cómo? Plantéate cómo generar un término de la forma $x^i/i!$ a partir de un término de la forma $x^{i-1}/(i-1)!.$)

- 2) Imprime los n primeros términos de la sucesión de Fibonacci. Resuélvelo con una función recursiva.

Solución :

LIBRO DE REFERENCIA

Introducción a la programación con C.

by [Andrés Marzal Varó](#); [Isabel Gracia Luengo](#)

Revisar Estructuras de Control Iterativas (págs 60-68) y Funciones (págs 152-176).

link de acceso a libro:

https://ia800808.us.archive.org/0/items/2010IntroduccionALaProgramacionConC/2010_introduccion-a-la-programacion-con-c.pdf

