



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Maricela Castañeda Perdomo

Asignatura: Fundamentos de programación

Grupo: 12

No. de práctica(s): Práctica 8: Estructuras de repetición.

Integrante(s): Kanafany Badillo Fernando

No. de lista o brigada: 22

Semestre: 2023-2

Fecha de entrega: 12/05/2023

Observaciones:

CALIFICACIÓN: _____

Objetivo:

El alumno elaborará programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición.

Actividades:

- Elaborar un programa que utilice la estructura while en la solución de un problema
- Elaborar un programa que requiera el uso de la estructura do-while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while.
- Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while.

Introducción

Las estructuras de repetición son las llamadas también estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera). En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

Una estructura repetitiva o bucle se utiliza cuando se quiere repetir un conjunto de sentencias un número determinado de veces o mientras se mantenga el cumplimiento de una condición. El bucle for utiliza la primera opción usando una variable numérica capaz de controlar el número de iteraciones.

¿Cómo funciona un bucle while?

El bucle while es otra estructura de control de flujo, concretamente lo que hace es repetir un código mientras dure una determinada condición. Se puede decir que el bucle while se utiliza para hacer algo repetidamente, bajo unas condiciones específicas, sin saber cuántas veces se repetirá.

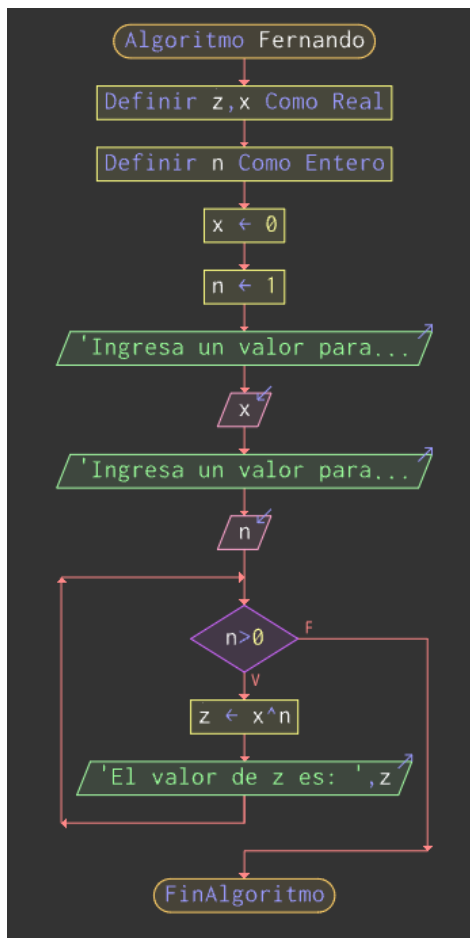
¿Qué significa el bucle for?

Un bucle for es un bucle que repite el bloque de instrucciones un número predeterminado de veces. El bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición se suele llamar iteración.

Ejercicio 1

While

```
int main() {
    int n;
    float x;
    float z;
    x = 0;
    n = 1;
    cout << "Ingresa un valor para x" << x << endl;
    cin >> x;
    cout << "Ingresa un valor para el exponente" << n << endl;
    cin >> n;
    while (n>0) {
        z = pow(x,n);
        cout << "El valor de z es: " << z << endl;
    }
    return 0;
}
```



For

Lamentablemente no pude hacer con el bucle for, me marcaba solo errores cuando lo intentaba.):

Ejercicio 2

Serie Fibonacci

```

Algoritmo fibonacci
  Escribir 'Ingresar un numero'
  Leer n

  a<-0
  b<-1

  Para i<-1 Hasta n Hacer
    Escribir a
    c<-a+b
    a<-b
    b<-c
  Fin Para
FinAlgoritmo

```

PSelnt - Ejecutando proceso FIBONACCI

```

*** Ejecución Iniciada. ***
Ingresar un numero
> 4
0
1
1
2
*** Ejecución Finalizada. ***

```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```

#include<stdio.h>

int main() {
    float a;
    float b;
    float c;
    float i;
    float n;
    printf("Ingresa un numero\n");
    scanf("%f",&n);
    a = 0;
    b = 1;
    for (i=1;i<=n;i+=1) {
        printf("%f\n",a);
        c = a+b;
        a = b;
        b = c;
    }
    return 0;
}

```

Ejercicio 3

Mostrar los N primeros números y sus respectivos cuadrados.

```

Algoritmo Naturales
    Definir i,num, cuad Como Real
    Escribir 'Dame los primeros 4 numeros naturales: '
    num<-0
    Leer num
    Para num<-1 Hasta num Con Paso 1 Hacer
        cuad<-num 2

        Escribir 'Tus numeros naturales: ', num
        Escribir 'Sus cuadrados: ',cuad
    Fin Para
FinAlgoritmo

```

```

*** Ejecución Iniciada. ***
Dame los primeros 4 numeros naturales:
> 4
Tus numeros naturales: 1
Sus cuadrados: 1
Tus numeros naturales: 2
Sus cuadrados: 4
Tus numeros naturales: 3
Sus cuadrados: 9
Tus numeros naturales: 4
Sus cuadrados: 16
*** Ejecución Finalizada. ***

```

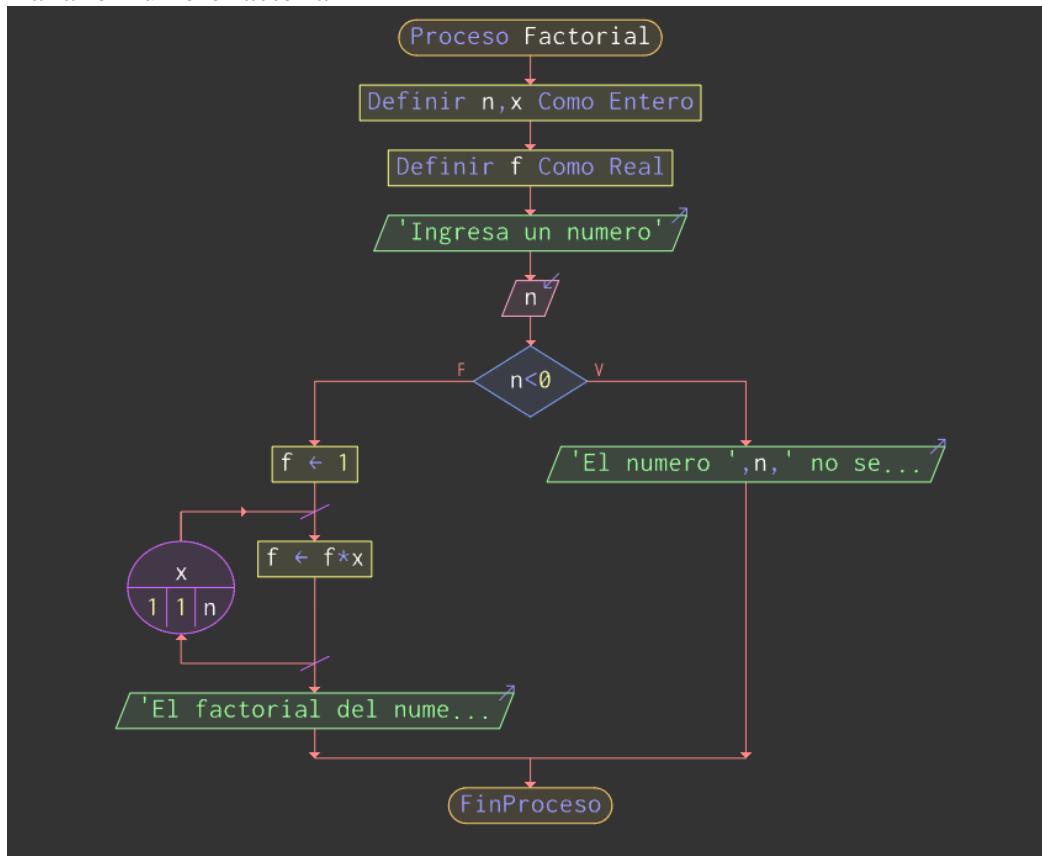
```

int main() {
    float cuad;
    float i;
    float num;
    printf("Dame los primeros 4 numeros naturales: \n");
    num = 0;
    scanf("%f",&num);
    for (num=1;num<=num;num+=1) {
        cuad = pow(num,2);
        printf("Tus numeros naturales: %f\n",num);
        printf("Sus cuadrados: %f\n",cuad);
    }
    return 0;
}

```

Ejercicio 4

Hallar el numero factorial



```

#include<stdio.h>

int main() {
    float f;
    int n;
    int x;
    printf("Ingresa un numero\n");
    scanf("%i",&n);
    if (n<0) {
        printf("El numero %i no se puede calcular\n",n);
    } else {
        f = 1;
        for (x=1;x<=n;x+=1) {
            f = f*x;
        }
        printf("El factorial del numero%i es: %f\n",n,f);
    }
    return 0;
}

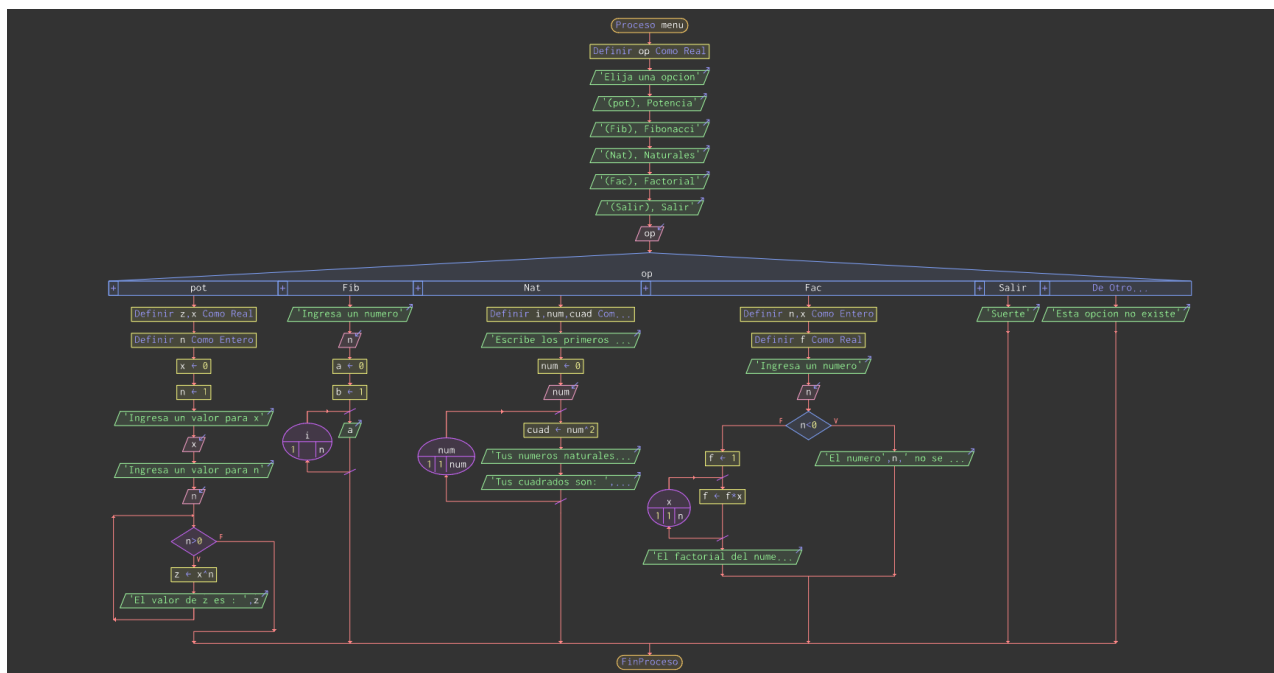
```

Ejercicio 5

Integrar los ejercicios anteriores en un menú con ciclo. El usuario debe seleccionar que desea realizar. Se debe hacer con selección múltiple.

MENÚ

1. Elevar un número a una potencia
2. Serie de Fibonacci
3. N números y los respectivos cuadrados
4. Factorial de un número
5. Salir



```

int main() {
    float a;
    float b;
    float cuad;
    float f;
    int fac;
    int fib;
    float i;
    int n;
    int nat;
    float num;
    int op;
    int pot;
    int salir;
    int x;
    float z;
    printf("Elija una opcion\n");
    printf("(pot), Potencia\n");
    printf("(fib), Fibonacci\n");
    printf("(nat), Naturales\n");
    printf("(fac), Factorial\n");
    printf("(salir), Salir\n");
    scanf("%i",&op);
    switch (op) {
        case pot:
            x = 0;
            n = 1;
            printf("Ingresa un valor para x\n");
            scanf("%i",&x);
            printf("Ingresa un valor para n\n");
            scanf("%i",&n);
            while (n>0) {
                z = pow(x,n);
                printf("El valor de z es : %f\n",z);
            }
            break;
        case fib:
            printf("Ingresa un numero\n");
            scanf("%i",&n);
            a = 0;
            b = 1;
            for (i=1;i<=n;i+=1) {
                printf("%i\n",a);
            }
            break;
        case nat:
            printf("Escribe los primeros 4 numeros reales: \n");
            num = 0;
            scanf("%f",&num);
            for (num=1;num<=num;num+=1) {
                cuad = pow(num,2);
                printf("Tus numeros naturales son: %f\n",num);
                printf("Tus cuadrados son: %f\n",cuad);
            }
            break;
        case fac:
            printf("Ingresa un numero\n");
            scanf("%i",&n);
            if (n<0) {
                printf("El numero no se puede calcular\n",n);
            } else {
                f = 1;
                for (x=1;x<=n;x+=1) {
                    f = f*x;
                }
                printf("El factorial del numero es: %f\n",n,f);
            }
            break;
        case salir:
    }
}

```

Conclusiones: En la práctica aprendí a usar los bucles while y for, pude elaborar pseudocodigos y diagramas de flujo en pseint y me fue mucho de ayuda para poder guiarme en la elaboración de los algoritmos en c.

Referencias:

- Laboratorio Salas A y B. (s. f.). <http://lcp02.fi-b.unam.mx/>
- Bucle for (1). Python. Bartolomé Sintés Marco. www.mclibre.org. (s. f.). <https://www.mclibre.org/consultar/python/lecciones/python-for.html#:~:text=Un%20bucle%20for%20es%20un,repetici%C3%B3n%20se%20suele%20llamar%20iteraci%C3%B3n>.
- Formación, A. (2023). Programación en Python: el bucle While. www.adrformacion.com. <https://www.adrformacion.com/knowledge/programacion/programacion-en-python-el-bucle-while.html#:~:text=El%20bucle%20while%20es%20otra,saber%20cuantas%20veces%20se%20repetir%C3%A1>.