



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3 Bloque: 136

No de Práctica(s): 09: Estructuras de repetición

Integrante(s): Carranza Ochoa José David

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: 6

Semestre: 2021-1

Fecha de entrega: 23/11/2020

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

Introducción:

Las estructuras de repetición son las llamadas 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.

Desarrollo:

Durante el desarrollo de la práctica, se realizaron diferentes actividades en formato de ciclos, utilizando el condicional For y While. Este ciclo es repetitivo siempre y cuando no se cumpla la condición final; así mismo, la estructura indica que si se cumple la condición se ejecuta la acción, de lo contrario sigue el código su programa

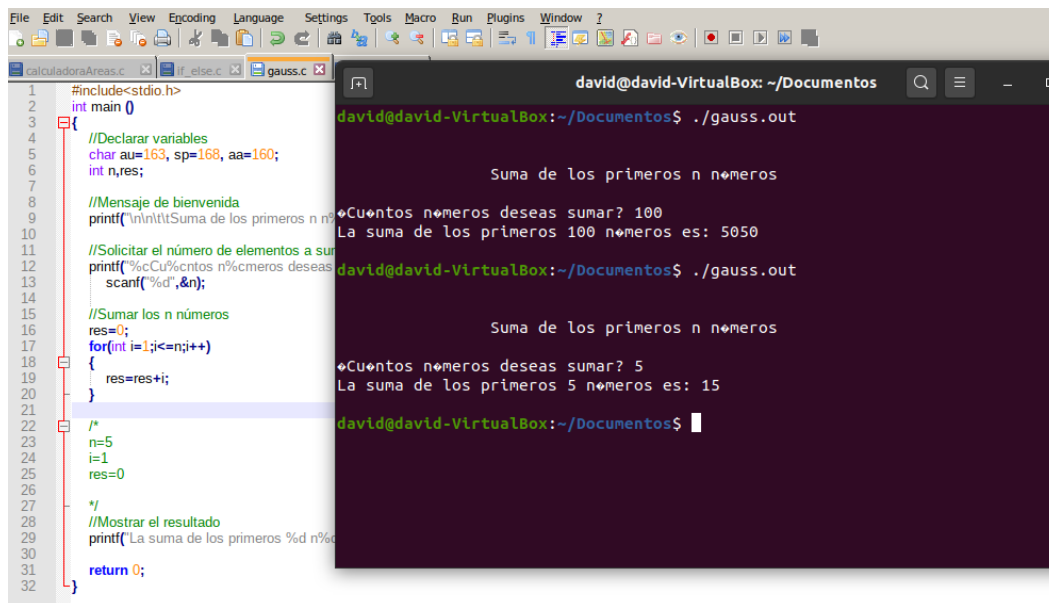
Para ejemplificar este proceso, se puede realizar por medio de la repetición “for” el ejemplo hecho hace años por Gauss donde se suma un número junto con todos sus antecesores, recordando la sintaxis

for (inicialización; expresión lógica; operaciones por iteración)

{

/* Bloque de código a ejecutar */

}



The screenshot shows a C program in a text editor and its execution in a terminal. The C program, named `gauss.c`, implements Gauss's summation algorithm. It includes a header `<stdio.h>` and a `main` function. It declares variables `au`, `sp`, `aa`, `n`, and `res`. It prints a welcome message and prompts the user for the number of elements to sum. It then uses a `for` loop to calculate the sum of the first `n` numbers. Finally, it prints the result and returns 0.

```
1 #include<stdio.h>
2 int main ()
3 {
4     //Declarar variables
5     char au=163, sp=168, aa=160;
6     int n,res;
7
8     //Mensaje de bienvenida
9     printf("\n\n\t\t\tSuma de los primeros n números\n\n");
10
11     //Solicitar el número de elementos a sumar
12     printf("%cCu%cntos n%cmneros deseas sumar? ");
13     scanf("%d",&n);
14
15     //Sumar los n números
16     res=0;
17     for(int i=1;i<=n;i++)
18     {
19         res=res+i;
20     }
21
22     /*
23     n=5
24     i=1
25     res=0
26
27     */
28     //Mostrar el resultado
29     printf("La suma de los primeros %d números es: %d\n",n,res);
30
31     return 0;
32 }
```

The terminal output shows the program being executed with `n=100` and `n=5`, resulting in the sums 5050 and 15 respectively.

```
david@david-VirtualBox: ~/Documentos
david@david-VirtualBox:~/Documentos$ ./gauss.out

Suma de los primeros n números

Cuántos números deseas sumar? 100
La suma de los primeros 100 números es: 5050

david@david-VirtualBox:~/Documentos$ ./gauss.out

Suma de los primeros n números

Cuántos números deseas sumar? 5
La suma de los primeros 5 números es: 15

david@david-VirtualBox:~/Documentos$
```

Así mismo, se puede obtener la factorial de un número dado siguiendo la regla en la que se multiplican todos los números antecesores al expresado:

```

1  #include<stdio.h>
2  int main ()
3  {
4      //Declarar variables
5      char au=163, sp=168, aa=160;
6      int n,res;
7
8      //Mensaje de bienvenida
9      printf("\n\n\t\t\tFactorial del n°cmro n \n\n",au);
10
11     //Solicitar el número de elementos a sumar
12     printf("%cCu%cl n°cmro deseas obtener su factorial? ",sp,aa,au);
13     scanf("%d",&n);
14
15     //Multiplicar los n numeros
16     res=1;
17     for(int i=1;i<=n;i++)
18     {
19         res=res*i;
20     }
21
22     /*
23     n=5
24     i=1
25     res=1
26     res = 1 * 1 =1
27     */
28
29     //Mostrar el resultado
30     printf("El factorial del n°cmro %d es: %d \n\n",au,n,res);
31
32     return 0;
33 }

```

```

david@david-VirtualBox: ~/Documentos
david@david-VirtualBox:~/Documentos$ ./factorial.out

Factorial del número n

¿Cu?l número deseas obtener su factorial? 5
El factorial del número 5 es: 120

david@david-VirtualBox:~/Documentos$ ./factorial.out

Factorial del número n

¿Cu?l número deseas obtener su factorial? 6
El factorial del número 6 es: 720

david@david-VirtualBox:~/Documentos$

```

La estructura While, indica al código ejecutar en el orden indicado siempre y cuando se cumpla la condición expresada, su sintaxis presenta pequeñas diferencias a For, por lo que en ejemplos sencillos no es mucha la diferencia

```

while (expresión_lógica)
{
    // Bloque de código a repetir
    // mientras que la expresión
    // lógica sea verdadera.
}

```

Para ejemplificar, se hará el mismo ejercicio de Gauss cambiando la estructura:

```

1  #include<stdio.h>
2  int main ()
3  {
4      //Declarar variables
5      char au=163, sp=168, aa=160;
6      int n,res,i;
7
8      //Mensaje de bienvenida
9      printf("\n\n\t\t\tSuma de los primeros n n°cmros \n\n",au);
10
11     //Solicitar el número de elementos a sumar
12     printf("%cCu%cl n°cmros deseas sumar? ",sp,aa,au);
13     scanf("%d",&n);
14
15     //Sumar los n números
16     res=0;
17     i=1;
18     while(i<=n)
19     {
20         res=res+i;
21         i++;
22     }
23
24     /*
25     n=5
26     i=1
27     res=0
28     */
29
30     //Mostrar el resultado
31     printf("La suma de los primeros %d n°cmros es: %d \n\n",au,n,res);
32
33     return 0;
34 }

```

```

david@david-VirtualBox: ~/Documentos
david@david-VirtualBox:~/Documentos$ ./gaussWhile.out

Suma de los primeros n números

¿Cu?ntos números deseas sumar? 100
La suma de los primeros 100 números es: 5050

david@david-VirtualBox:~/Documentos$ ./gaussWhile.out

Suma de los primeros n números

¿Cu?ntos números deseas sumar? 5
La suma de los primeros 5 números es: 15

david@david-VirtualBox:~/Documentos$

```

Mientras que la factorial de un número con la estructura While queda como:

```
#include<stdio.h>
int main ()
{
    //Declarar variables
    char au=163, sp=168, aa=160;
    int n,res,i;

    //Mensaje de bienvenida
    printf("\n\n\t\tFactorial del n%cmero n \n\n",au);

    //Solicitar el número de elementos a sumar
    printf("\n\n\t\t¿Cuánto n%cmero deseas obtener su factorial?\n\n",aa);
    scanf("%d",&n);

    //Multiplicar los n numeros
    res=1;
    i=1;
    while (i<=n)
    {
        res=res*i;
        i++;
    }

    /*
    n=5
    i=1
    res=1
    res = 1 * 1 =1
    */

    //Mostrar el resultado
    printf("El factorial del n%cmero %d es: %d \n\n",au,n,res);

    return 0;
}
```

La estructura Do While, indica que al menos una vez se realizará la condición sin validar su correcto funcionamiento, notando el “;” que incluye al final:

do

{

```
/* Bloque de código que se ejecuta por lo menos una vez y se repite mientras la
expresión lógica sea verdadera. */
```

}

```
while (expresión_lógica);
```

Comúnmente es empleado para realizar menús con opciones diversas

```
File Edit Search View Encoding Language Settings Tools Macro Bin Plugins Window
#include<stdio.h>
int main()
{
    //Variables a utilizar
    int opc;
    char aa=160, ae=130, ai=161, ao=162, au=163;

    //Mensaje de bienvenida
    printf("\nInit Menu de figuras \n\n",au);

    do
    {
        //Mostrar el menú
        printf("1)Triángulo\n2) Rectángulo\n3) Círculo\n4) Salir\n\n",aa,aa,ai);

        //Solicitar la opción que desea
        printf("Elige una opción: ",ao);
        scanf("%d",&opc);

        switch(opc)
        {
            case 1:
                printf("\nElige triángulo \n\n",aa);
                break;
            case 2:
                printf("\nElige rectángulo \n\n",aa);
                break;
            case 3:
                printf("\nElige círculo \n\n",ai);
                break;
            case 4:
                printf("\nElige salir \n\n");
            default:
                printf("Opción no válida \n\n",ao,aa);
        }

    }while(opc!=4);

    printf("Gracias por usar el programa\n\n");
    return 0;
}
```

```

Menu de figuras

1)Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción
1
Elige triángulo

1)Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción
2
Elige rectángulo

1)Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción
3
Elige círculo

Elige una opción
4
Elige salir

Elige una opción
5
Opción no válida

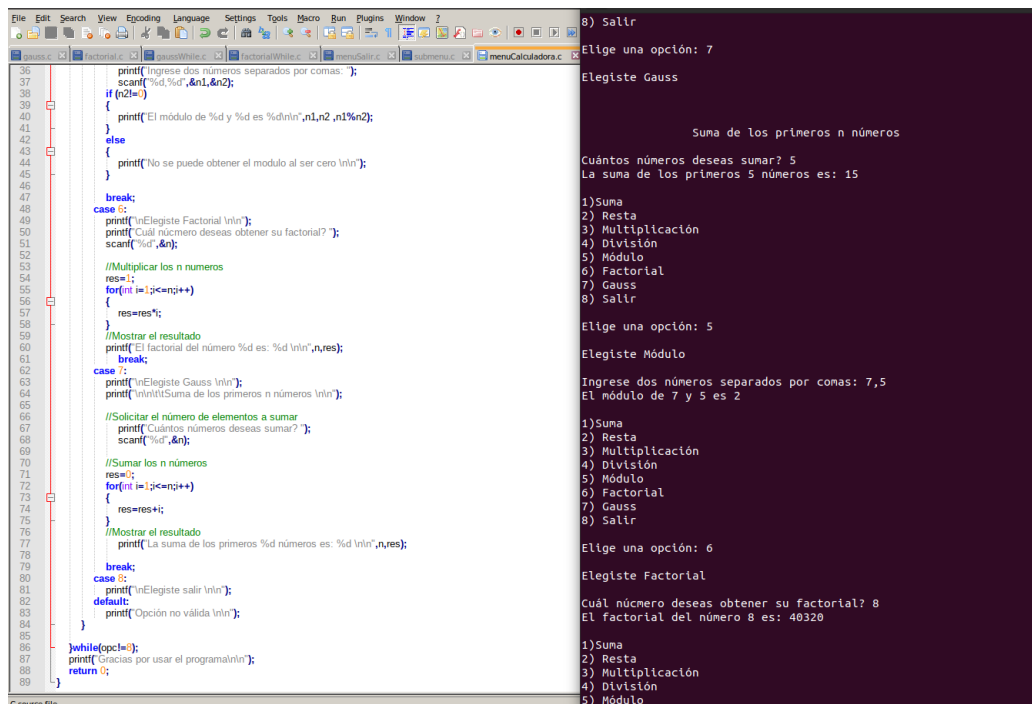
1)Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción
3
Elige círculo

Elige una opción
4
Elige salir

Elige una opción
5
Opción no válida
```

Podemos realizar diferentes acciones con el menú tal es el caso de una calculadora



```
36 printf("Ingrese dos números separados por comas: ");
37 scanf("%d,%d",&n1,&n2);
38 if (n2!=0)
39 {
40     printf("El módulo de %d y %d es %d\n",n1,n2,n1%n2);
41 }
42 else
43 {
44     printf("No se puede obtener el modulo al ser cero \n");
45 }
46
47 break;
48 case 5:
49     printf("\nElegiste Factorial \n");
50     printf("Cuál número deseas obtener su factorial? ");
51     scanf("%d",&n);
52
53     //Multiplicar los n números
54     res=1;
55     for(int i=1;i<=n;i++)
56     {
57         res=res*i;
58     }
59     //Mostrar el resultado
60     printf("El factorial del número %d es: %d \n",n,res);
61     break;
62 case 6:
63     printf("\nElegiste Gauss \n");
64     printf("\n\nLa suma de los primeros n números \n");
65
66     //Solicitar el número de elementos a sumar
67     printf("¿Cuántos números deseas sumar? ");
68     scanf("%d",&n);
69
70     //Sumar los n números
71     res=0;
72     for(int i=1;i<=n;i++)
73     {
74         res=res+i;
75     }
76     //Mostrar el resultado
77     printf("La suma de los primeros %d números es: %d \n",n,res);
78
79     break;
80 case 7:
81     printf("\nElegiste salir \n");
82     default:
83         printf("Opción no válida \n");
84 }
85
86 }while(opc!=0);
87 printf("Gracias por usar el programa\n");
88 return 0;
89 }
```

8) Salir
Elige una opción: 7
Elegiste Gauss

Suma de los primeros n números
Cuántos números deseas sumar? 5
La suma de los primeros 5 números es: 15

1) Suma
2) Resta
3) Multiplicación
4) División
5) Módulo
6) Factorial
7) Gauss
8) Salir
Elige una opción: 5
Elegiste Módulo
Ingrese dos números separados por comas: 7,5
El módulo de 7 y 5 es 2

1) Suma
2) Resta
3) Multiplicación
4) División
5) Módulo
6) Factorial
7) Gauss
8) Salir
Elige una opción: 6
Elegiste Factorial
Cuál número deseas obtener su factorial? 8
El factorial del número 8 es: 40320

1) Suma
2) Resta
3) Multiplicación
4) División
5) Módulo

Conclusiones:

Las estructuras repetitivas presentan una gran importancia al momento de codificar ya que dictan nuevas funciones que sentencias anteriores no hubieran podido realizar, por lo mismo, es de suma importancia ejecutarlas de manera correcta conociendo su función.

Con base a la práctica las identifiqué y supe administrar en los diferentes ejercicios realizados, dando los resultados que esperaba después de estudiar un poco más por mi cuenta. Respecto al manual, tuve ligeras dudas en la implementación del código que tuve que investigar por mi cuenta.

Bibliografía:

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

