



Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ciencias Básicas

Fundamentos de Programación (1122)



Profesor: M.I. Marco Antonio Martínez

Semestre 2021-1

Actividad asíncrona #14: **Estructuras de repetición**

Terán García Rodolfo Mario

Grupo: 03

N° de cuenta: 318017624

Correo electrónico:

mario_teran@comunidad.unam.mx



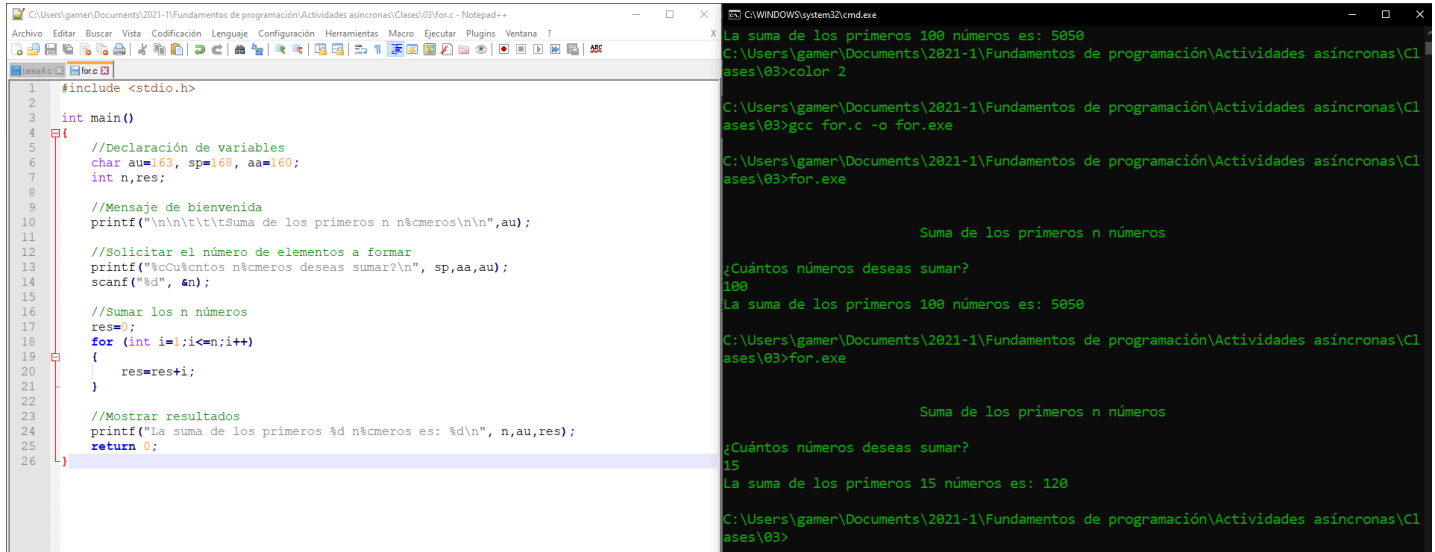
Ciudad de México a 02 de diciembre del 2020

Estructuras de repetición

Las estructuras de repetición nos permiten como su nombre lo indica, repetir una o más instrucciones, ya sea un número determinado de veces o mientras se cumpla una condición.

For: Nos permite repetir una o más instrucciones un determinado número de veces.

Implementación(gauss):



```
#include <stdio.h>

int main()
{
    //Declaración de variables
    char au=163, sp=168, aa=160;
    int n,res;

    //Mensaje de bienvenida
    printf("\n\n\t\tSuma de los primeros n números\n\n",au);

    //Solicitar el número de elementos a formar
    printf("\n\n\t\t¿Cuántos números deseas sumar?\n", sp,aa,au);
    scanf("%d", &n);

    //Sumar los n números
    res=0;
    for (int i=1;i<=n;i++)
    {
        res=res+i;
    }

    //Mostrar resultados
    printf("\n\n\t\tLa suma de los primeros %d números es: %d\n", n,au,res);
    return 0;
}
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gcc for.c -o for.exe

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>for.exe

Suma de los primeros n números

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

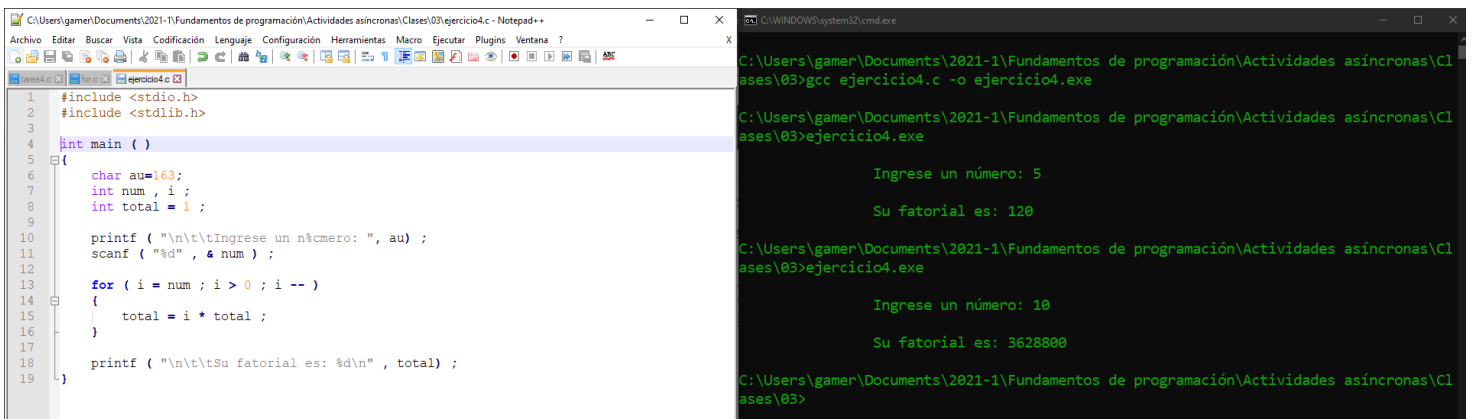
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>for.exe

Suma de los primeros n números

¿Cuántos números deseas sumar?
15
La suma de los primeros 15 números es: 120

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>
```

Ejercicio 4 (factorial):



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

int main ( )
{
    char au=163;
    int num , i ;
    int total = 1 ;

    printf ( "\n\n\t\tIngrese un número: ", au ) ;
    scanf ( "%d" , & num ) ;

    for ( i = num ; i > 0 ; i -- )
    {
        total = i * total ;
    }

    printf ( "\n\n\t\tSu factorial es: %d\n" , total ) ;
}
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gcc ejercicio4.c -o ejercicio4.exe

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>ejercicio4.exe

Ingrese un número: 5

Su factorial es: 120

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>ejercicio4.exe

Ingrese un número: 10

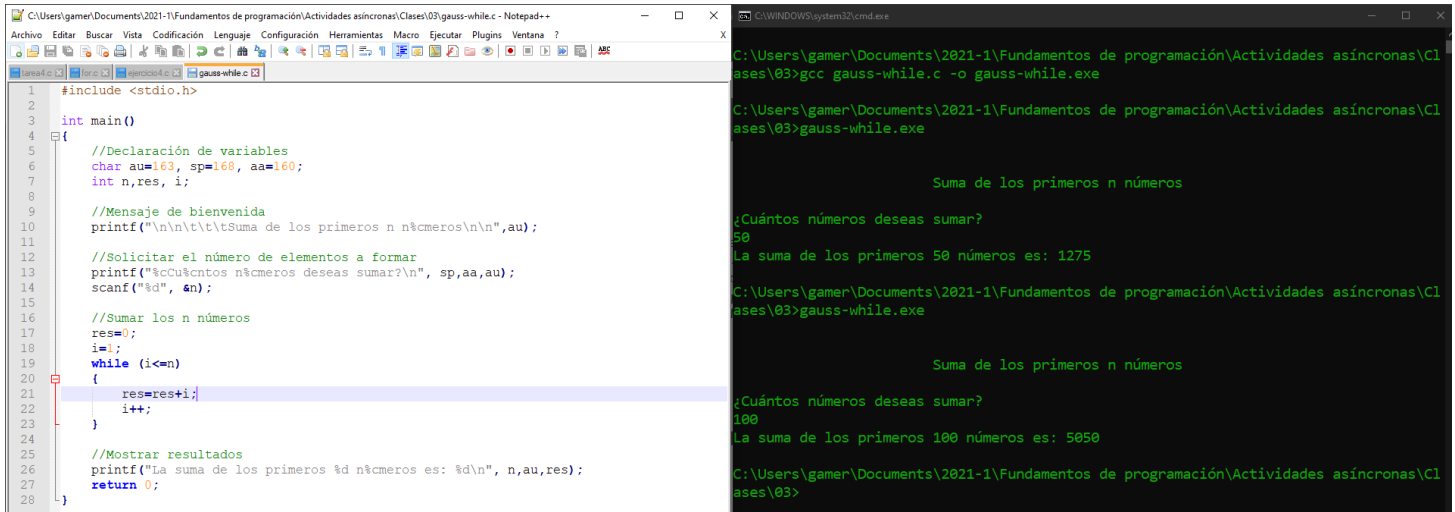
Su factorial es: 3628800

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>
```

Con el uso de la herramienta for se puede realizar un sinfín de actividades relacionados al ámbito industrial de la computación, una de ellas es la impresión de una factura o la generación de número de cuenta o números de lista que ocupen diferentes caracteres de forma masiva, pues realmente el único limite que encuentran este tipo de programas es la imaginación y los recursos de hardware disponibles, los cuales muy difícilmente serán insuficientes para el correcto funcionamiento de un programa en C por más complejo que pueda parecer.

While: Esta estructura de repetición nos permite validar una condición antes de realizar el ciclo.

Implementación (Gauss-while):



```
#include <stdio.h>

int main()
{
    //Declaración de variables
    char au=163, sp=168, aa=160;
    int n,res, i;

    //Mensaje de bienvenida
    printf("\n\n\t\t\tSuma de los primeros n números\n\n",au);

    //Solicitar el número de elementos a formar
    printf("\n\n\t\t\t¿Cuántos números desea sumar?\n", sp,aa,au);
    scanf("%d", &n);

    //Sumar los n números
    res=0;
    i=1;
    while (i<=n)
    {
        res=res+i;
        i++;
    }

    //Mostrar resultados
    printf("\n\n\t\t\tLa suma de los primeros %d números es: %d\n", n,au,res);
    return 0;
}
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gcc gauss-while.c -o gauss-while.exe

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gauss-while.exe

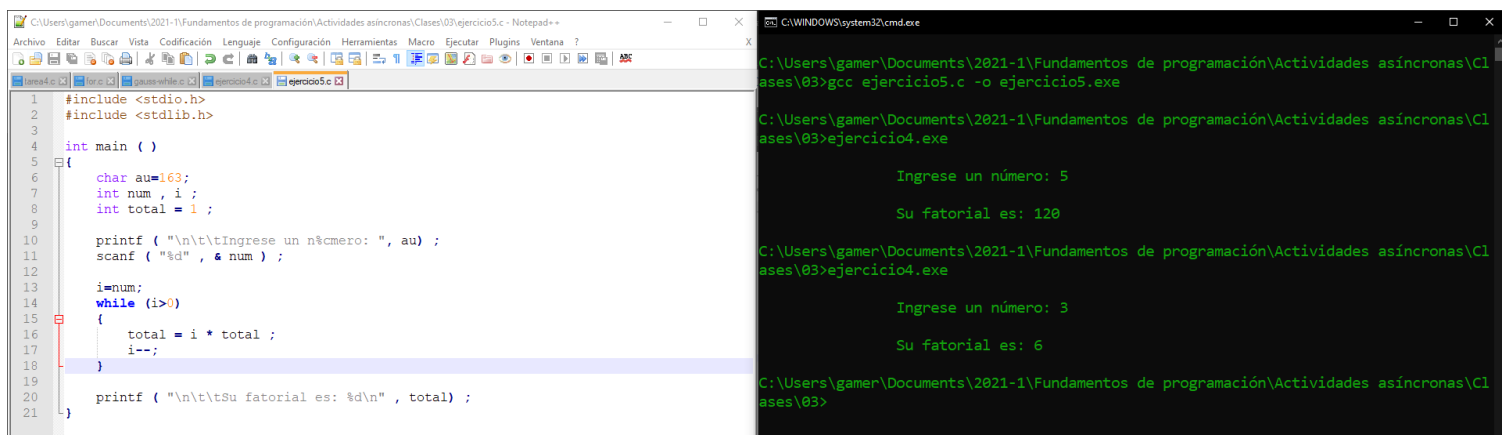
Suma de los primeros n números
¿Cuántos números desea sumar?
50
La suma de los primeros 50 números es: 1275

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gauss-while.exe

Suma de los primeros n números
¿Cuántos números desea sumar?
100
La suma de los primeros 100 números es: 5050

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>
```

Ejercicio 5 (factorial-while):



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

int main ( )
{
    char au=163;
    int num , i ;
    int total = 1 ;

    printf ( "\n\n\t\t\tIngrese un número: ", au ) ;
    scanf ( "%d" , & num ) ;

    i=num;
    while (i>0)
    {
        total = i * total ;
        i--;
    }

    printf ( "\n\n\t\t\tSu factorial es: %d\n" , total ) ;
}
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>gcc ejercicio5.c -o ejercicio5.exe

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>ejercicio5.exe

Ingrese un número: 5

Su factorial es: 120

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>ejercicio5.exe

Ingrese un número: 3

Su factorial es: 6

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Actividades asincronas\Clases\03>
```

El caso de la estructura while es bastante peculiar, pues pese a existir la estructura for, sus usos pueden ser muy variados a esta e incluso brindarnos soluciones creativas para acciones que el ciclo for no puede cumplir, pues los ciclos while son también una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos, los ciclos while, no dependen directamente de valores numéricos, sino de valores booleanos, es decir su ejecución depende del valor de verdad de una condición dada, verdadera o falso, nada más.

Implementación (menú de figuras):

