



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

No de Práctica(s): #9

Integrante(s): Fausto Ángel Reséndiz Álvarez

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

No. de Lista o Brigada: 38

Semestre: 2021-1

Fecha de entrega: 04/12/2020

Observaciones:

CALIFICACIÓN: _____

Objetivo

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

Introducción

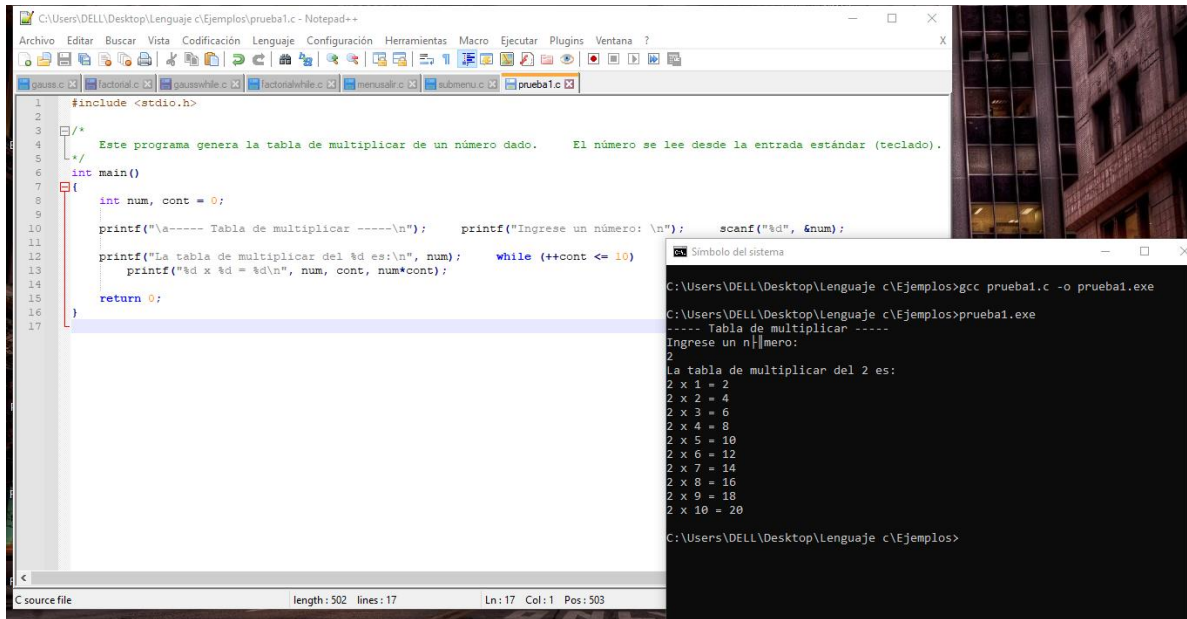
Las estructuras de repetición o estructuras cíclicas, iterativas o de bucles sirven para ejecutar un conjunto de instrucciones de manera repetida mientras que la expresión lógica a evaluar sea verdadera.

Algunos tipos de estas estructuras son:

- While
- do-while
- for

Desarrollo

Estructura de repetición while

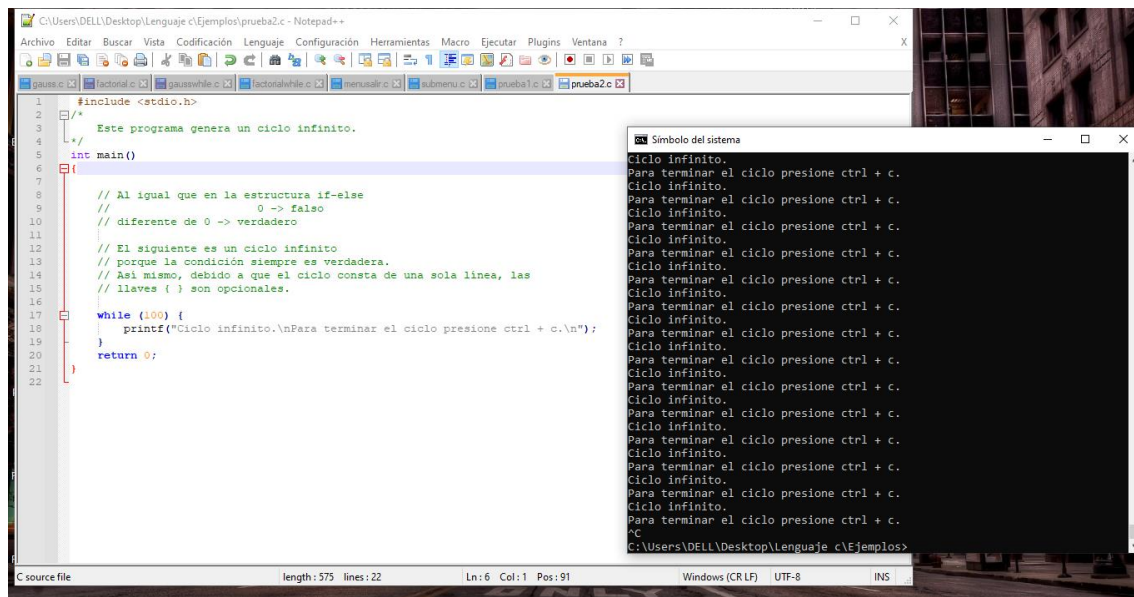


The screenshot shows a Notepad++ window with a C program named `prueba1.c`. The program uses a `while` loop to generate a multiplication table for a user-input number. The code is as follows:

```
1 #include <stdio.h>
2
3 /*
4  * Este programa genera la tabla de multiplicar de un número dado. El número se lee desde la entrada estándar (teclado).
5  */
6 int main()
7 {
8     int num, cont = 0;
9
10    printf("\a----- Tabla de multiplicar -----\\n");    printf("Ingrese un número: \\n");    scanf("%d", &num);
11
12    printf("La tabla de multiplicar del %d es:\\n", num);    while (++cont <= 10)
13        printf("%d x %d = %d\\n", num, cont, num*cont);
14
15    return 0;
16 }
17
```

The terminal window shows the execution of the program. It prompts the user to enter a number, and when '2' is entered, it displays the multiplication table for 2:

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc prueba1.c -o prueba1.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>prueba1.exe
----- Tabla de multiplicar -----
Ingrese un número: 2
La tabla de multiplicar del 2 es:
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

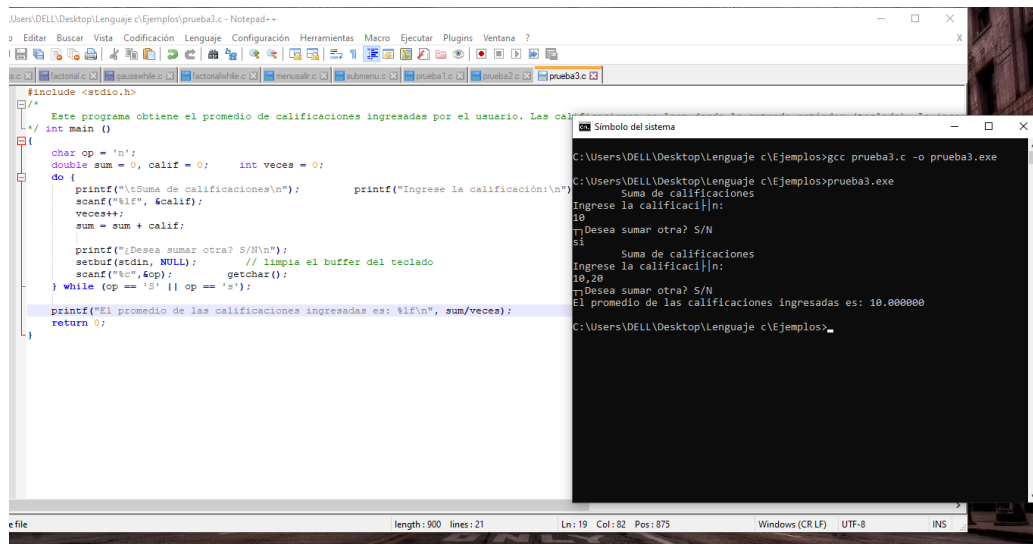


The screenshot shows a Notepad++ window with a C program named `prueba2.c`. The program uses a `while` loop to create an infinite loop. The code is as follows:

```
1 #include <stdio.h>
2
3 /*
4  * Este programa genera un ciclo infinito.
5  */
6 int main()
7 {
8     // Al igual que en la estructura if-else
9     // 0 -> falso
10    // diferente de 0 -> verdadero
11
12    // El siguiente es un ciclo infinito
13    // porque la condición siempre es verdadera.
14    // Así mismo, debido a que el ciclo consta de una sola línea, las
15    // llaves { } son opcionales.
16
17    while (100) {
18        printf("Ciclo infinito.\\nPara terminar el ciclo presione ctrl + c.\\n");
19    }
20    return 0;
21 }
22
```

The terminal window shows the execution of the program, which enters an infinite loop. It repeatedly prints the message "Ciclo infinito.\\nPara terminar el ciclo presione ctrl + c.\\n". The user is instructed to press `ctrl + c` to terminate the cycle.

estructura de repetición do-while



The screenshot shows a C program in Notepad++ and its execution in a command prompt. The program calculates the average of grades entered by the user. It uses a do-while loop to repeatedly prompt the user for grades until they enter '5' to stop.

```
#include <stdio.h>

/*
Este programa obtiene el promedio de calificaciones ingresadas por el usuario. Las calificaciones
deben ser ingresadas entre 0 y 10.
*/

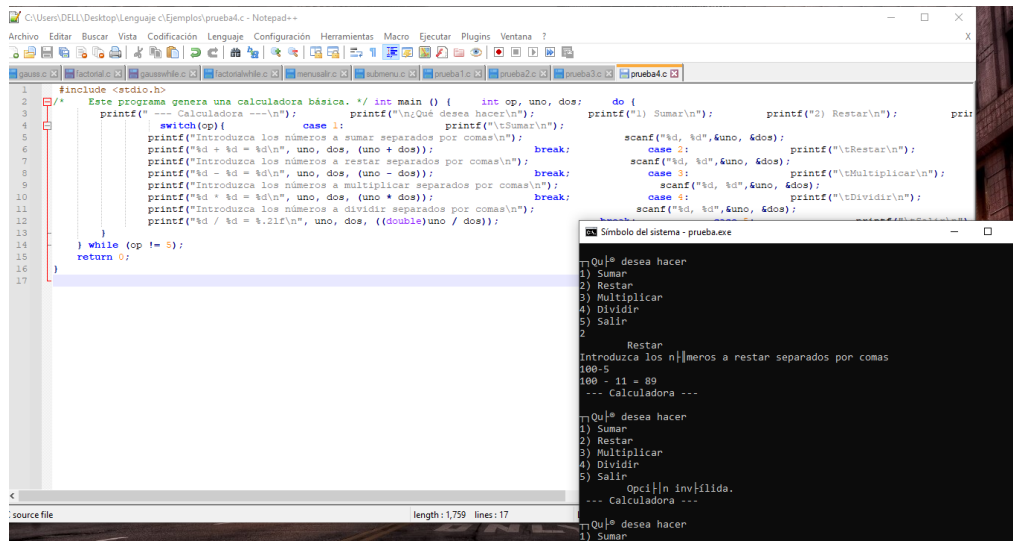
int main ()
{
    char op = '\n';
    double sum = 0, calif = 0;    int veces = 0;
    do {
        printf("\tSuma de calificaciones\n");    printf("Ingrese la calificación:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;

        printf("\tDesea sumar otra? S/N\n");
        setbuf(stdin, NULL);    // Limpia el buffer del teclado
        scanf("%c", &op);    getchar();
    } while (op == 'S' || op == 's');

    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
    return 0;
}
```

Command Prompt Output:

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc prueba3.c -o prueba3.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>prueba3.exe
Suma de calificaciones
Ingrese la calificación:\n
10
Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:\n
10,20
Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 10.000000
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```



The screenshot shows a C program in Notepad++ and its execution in a command prompt. The program is a basic calculator that allows the user to perform addition, subtraction, multiplication, and division. It uses a do-while loop to repeatedly prompt the user for operations and numbers until they enter '5' to exit.

```
#include <stdio.h>

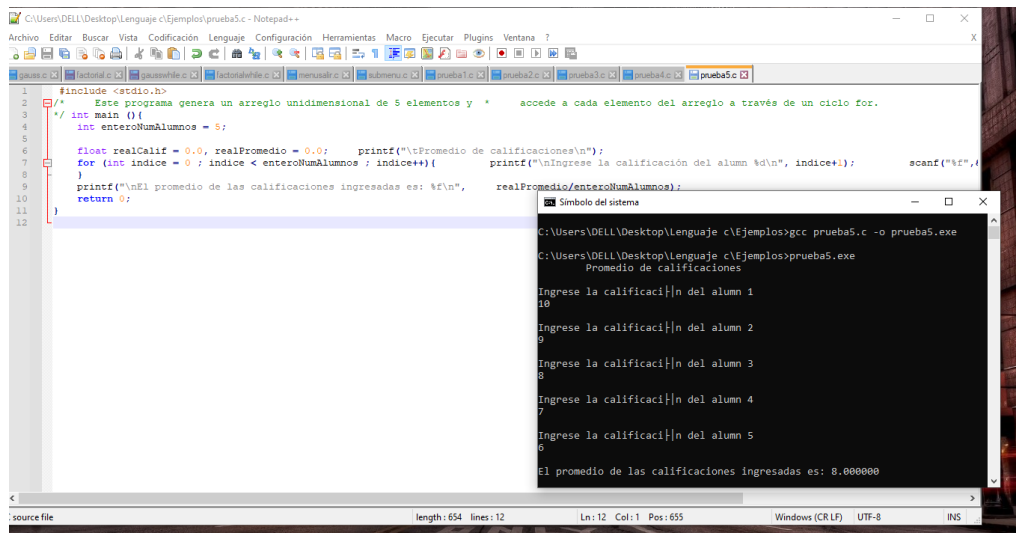
/*
Este programa genera una calculadora básica.
*/

int main () {
    int op, uno, dos;
    printf("\n--- Calculadora ---\n");
    printf("\n¿Qué desea hacer?\n");
    switch(op) {
        case 1:
            printf("\tSumar\n");
            printf("Introduzca los números a sumar separados por comas\n");
            scanf("%d, %d", &uno, &dos);
            printf("%d + %d = %d\n", uno, dos, (uno + dos));
            break;
        case 2:
            printf("\tRestar\n");
            printf("Introduzca los números a restar separados por comas\n");
            scanf("%d, %d", &uno, &dos);
            printf("%d - %d = %d\n", uno, dos, (uno - dos));
            break;
        case 3:
            printf("\tMultiplicar\n");
            printf("Introduzca los números a multiplicar separados por comas\n");
            scanf("%d, %d", &uno, &dos);
            printf("%d * %d = %d\n", uno, dos, (uno * dos));
            break;
        case 4:
            printf("\tDividir\n");
            printf("Introduzca los números a dividir separados por comas\n");
            scanf("%d, %d", &uno, &dos);
            printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
            break;
    }
    printf("\nOpciones:\n");
    printf("1) Sumar\n");
    printf("2) Restar\n");
    printf("3) Multiplicar\n");
    printf("4) Dividir\n");
    printf("5) Salir\n");
    printf("\nOpción inválida.\n");
    printf("\n--- Calculadora ---\n");
    return 0;
}
```

Command Prompt Output:

```
Qu? desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
2
Restar
Introduzca los números a restar separados por comas
100-5
100 - 5 = 95
--- Calculadora ---
Qu? desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
Opción inválida.
--- Calculadora ---
Qu? desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
Opción inválida.
--- Calculadora ---
```

estructura de repetición for



The screenshot shows a C program in Notepad++ and its execution in a command prompt. The program calculates the average of grades for 5 students. It uses a for loop to iterate through the array of grades and calculate the average.

```
#include <stdio.h>

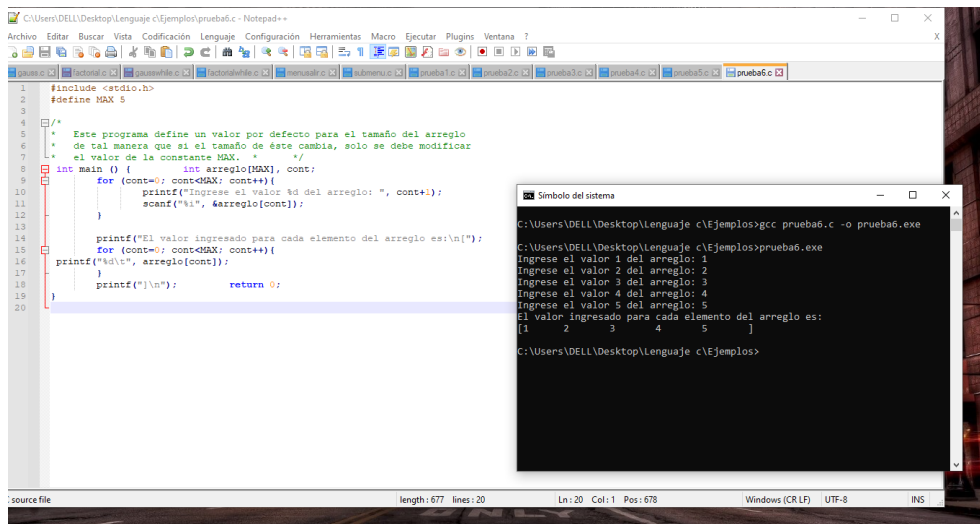
/*
Este programa genera un arreglo unidimensional de 5 elementos y
calcula el promedio de las calificaciones.
*/

int main () {
    int enteroNumAlumnos = 5;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    for (int indice = 0; indice < enteroNumAlumnos; indice++) {
        printf("Ingrese la calificación del alumno %d\n", indice+1);
        scanf("%f", &realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n", realPromedio/enteroNumAlumnos);
    return 0;
}
```

Command Prompt Output:

```
Promedio de calificaciones
Ingrese la calificación del alumno 1
10
Ingrese la calificación del alumno 2
9
Ingrese la calificación del alumno 3
8
Ingrese la calificación del alumno 4
7
Ingrese la calificación del alumno 5
6
El promedio de las calificaciones ingresadas es: 8.000000
```

Define



```
#include <stdio.h>
#define MAX 5

/*
 * Este programa define un valor por defecto para el tamaño del arreglo
 * de tal manera que si el tamaño de éste cambia, solo se debe modificar
 * el valor de la constante MAX.
 */

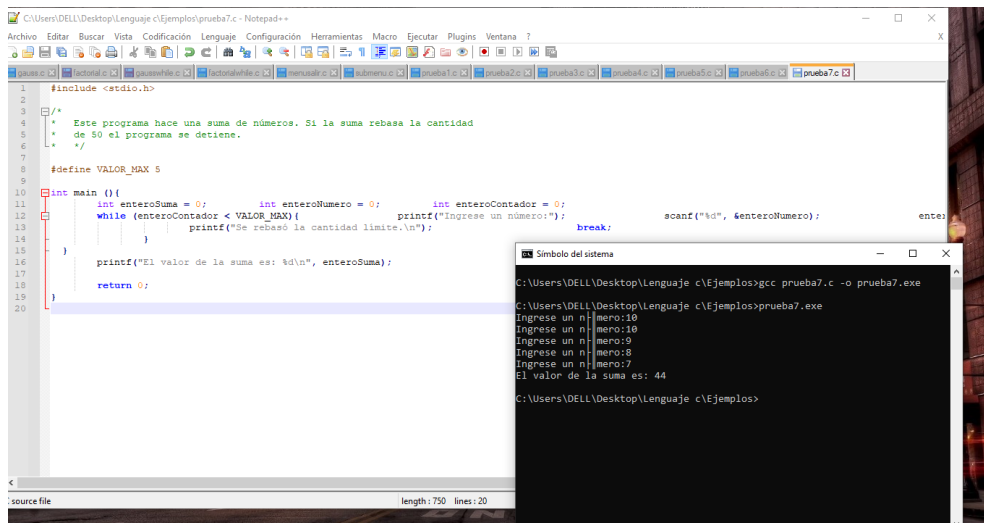
int main ()
{
    int arreglo[MAX], cont;
    for (cont=0; cont<MAX; cont++){
        printf("Ingrese el valor %d del arreglo: ", cont+1);
        scanf("%i", &arreglo[cont]);
    }

    printf("El valor ingresado para cada elemento del arreglo es:\n");
    for (cont=0; cont<MAX; cont++){
        printf("%d\t", arreglo[cont]);
    }

    printf("\n");
    return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc prueba6.c -o prueba6.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>prueba6.exe
Ingrese el valor 1 del arreglo: 1
Ingrese el valor 2 del arreglo: 2
Ingrese el valor 3 del arreglo: 3
Ingrese el valor 4 del arreglo: 4
Ingrese el valor 5 del arreglo: 5
El valor ingresado para cada elemento del arreglo es:
[1 2 3 4 5 ]
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

Break



```
#include <stdio.h>

/*
 * Este programa hace una suma de números. Si la suma rebasa la cantidad
 * de 50 el programa se detiene.
 */

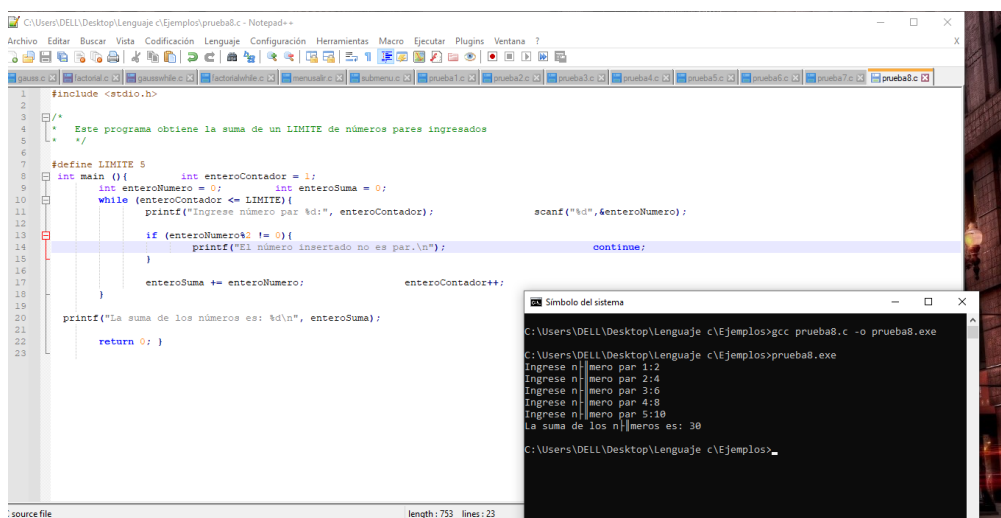
#define VALOR_MAX 5

int main ()
{
    int enteroSuma = 0;
    int enteroNumero = 0;
    int enteroContador = 0;
    while (enteroContador < VALOR_MAX)
    {
        printf("Ingrese un número: ");
        scanf("%d", &enteroNumero);
        enteroSuma += enteroNumero;
        enteroContador++;
        if (enteroSuma > VALOR_MAX)
        {
            printf("Se rebasó la cantidad limite.\n");
            break;
        }
    }

    printf("El valor de la suma es: %d\n", enteroSuma);
    return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc prueba7.c -o prueba7.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>prueba7.exe
Ingrese un n|mero:10
Ingrese un n|mero:10
Ingrese un n|mero:9
Ingrese un n|mero:8
Ingrese un n|mero:7
El valor de la suma es: 44
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

Continue



```
#include <stdio.h>

/*
 * Este programa obtiene la suma de un LIMITE de números pares ingresados
 */

#define LIMITE 5

int main ()
{
    int enteroContador = 1;
    int enteroNumero = 0;
    int enteroSuma = 0;
    while (enteroContador <= LIMITE)
    {
        printf("Ingrese número par %d:", enteroContador);
        scanf("%d", &enteroNumero);
        if (enteroNumero % 2 != 0)
        {
            printf("El número insertado no es par.\n");
            continue;
        }
        enteroSuma += enteroNumero;
        enteroContador++;
    }

    printf("La suma de los números es: %d\n", enteroSuma);
    return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc prueba8.c -o prueba8.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>prueba8.exe
Ingrese n|mero par 1:2
Ingrese n|mero par 2:4
Ingrese n|mero par 3:6
Ingrese n|mero par 4:8
Ingrese n|mero par 5:10
La suma de los n|meros es: 30
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

Conclusión

Las estructuras de repetición nos permiten agilizar los procesos donde necesitamos abarcar múltiples datos y con los cuales se requeriría que el programa se ejecute infinidad de veces y con el fin de evitar prácticas poco ortodoxas como estar ejecutando un ay otra vez el programa pasa solo ingresar un dato a la vez, se utilizan estas estructuras que nos evitan dicha tarea, obteniendo grandes beneficios para el usuario tales como la reducción de tiempos y el aumento en la productividad.

Dándole un valor muy importante a este tipo de estructuras y es crucial que nosotros como programadores dominemos estas estructuras para así lograr un mejor y aun más complejo desarrollo de programas.