



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Laboratorio de Fundamentos de Programación

Grupo: 3

No de Práctica(s): 9

Integrante(s): Cecilia Torres Bravo

*No. de Equipo de
cómputo empleado:* N/A

No. de Lista o Brigada: 54

Semestre: 1º

Fecha de entrega: 6 de diciembre del 2020

Observaciones:

CALIFICACIÓN: _____

Estructuras de repetición

Objetivo

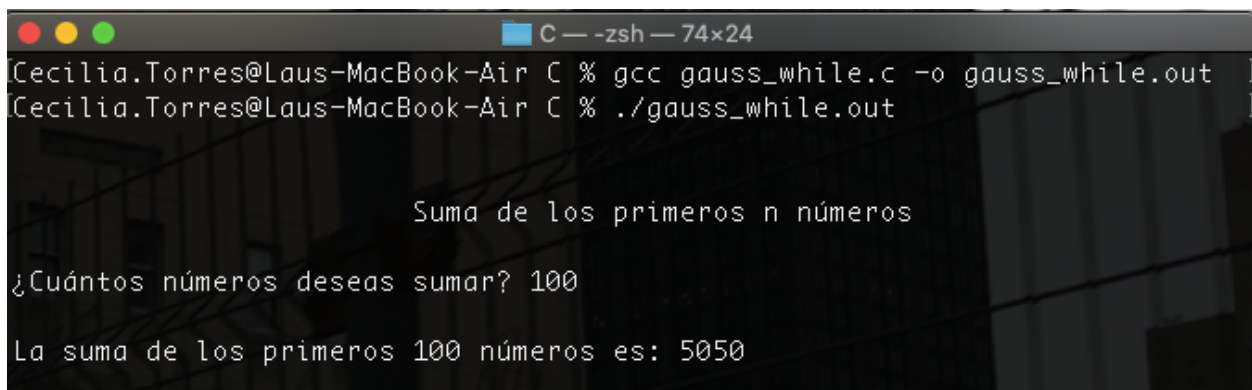
Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva *define*.

Introducción

Las estructuras de repetición nos permiten realizar sentencias en una manera cíclica siempre y cuando se cumpla una condición inicial. En el lenguaje de programación las tres estructuras de repetición son *for*, *while* y *do-while*. La estructura iterativa *while* nos un bloque de instrucciones siempre y cuando se valide y se cumpla la expresión lógica. Luego, la estructura de repetición *do-while*, al igual que *while*, nos permite ejecutar un bloque de instrucciones cuando se cumple la expresión lógica, la diferencia radica en que en *do-while* el código se sigue repitiendo mientras que la expresión lógica sea verdadera. Por último, la estructura de control de repetición *for* consiste en realizar las repeticiones cuando se cuenta con el número de elementos que se quiere recorrer. Asimismo, *define* nos permite definir constantes al principio del código.

Ejemplos

A-1 Elaborar un programa que utilice la estructura *while* en la solución de un problema.



```
Cecilia.Torres@Laus-MacBook-Air C % gcc gauss_while.c -o gauss_while.out
Cecilia.Torres@Laus-MacBook-Air C % ./gauss_while.out

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

```

C gauss_while.c > main()
1  #include<stdio.h>
2  int main()
3  {
4      //Declarar variables
5      int i,n,res;
6
7      //Mensaje de bienvenida
8      printf("\n\n\t\t\tSuma de los primeros n números\n\n");
9
10     //Solicitar el número de elementos a sumar
11     printf("¿Cuántos números deseas sumar? ");
12     scanf("%d",&n);
13
14     //Sumar los n números
15     res=0;
16     i=1;
17     while(i<=n)
18     {
19         res=res+i;
20         i++;
21     }
22
23     //Mostrar el resultado
24     printf("\nLa suma de los primeros %d números es: %d\n\n",n,res);
25
26     return 0;
27 }

```

A-2 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*.

Cecilia.Torres@Laus-MacBook-Air C % ./menuSalir.out

Menú de figuras :)

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

Selecciona una opción: 1

Elegiste triángulo

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

Selecciona una opción: 2

Elegiste rectángulo

C — -zsh — 74x24

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

Selecciona una opción: 3

Elegiste círculo

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

Selecciona una opción: 6

No es una opción válida

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

Selecciona una opción: 4

Elegiste salir

Gracias por usar nuestro programa

Screen Shot
2020-12-13 13:02:36

```

1  #include<stdio.h>
2  int main()
3  {
4      //Variables
5      int op;
6
7      //Mensaje de bienvenida
8      printf("\n\n\t\t\tMenú de figuras :)\n\n");
9
10     do
11     {
12         //Mostrar el menú
13         printf("1) Triángulo\n2) Rectángulo\n3) Círculo\n4) Salir\n");
14
15         //Solicitar la opción
16         printf("Selecciona una opción: ");
17         scanf("%d",&op);
18
19         switch(op)
20         {
21             case 1:
22                 printf("\nElegiste triángulo\n\n");
23                 break;
24             case 2:
25                 printf("Elegiste rectángulo\n\n");
26                 break;
27             case 3:
28                 printf("Elegiste círculo\n\n");
29                 break;
30             case 4:
31                 printf("Elegiste salir\n\n");
32                 break;
33             default:
34                 printf("No es una opción válida\n\n");
35         }
36     }
37     while(op!=4);
38
39     printf("Gracias por usar nuestro programa\n");
40
41     return 0;
42 }

```

La mayor diferencia entre el *while* y el *do-while* es que en *do-while* el código se sigue repitiendo mientras que la expresión lógica sea verdadera mientras que en *while* solo se repite hasta el valor indicado.

A-3 Resolver un problema dado por el profesor que utilice la estructura *for* en lugar de la estructura *while*.

```
C — -zsh — 70x24
[Cecilia.Torres@Laus-MacBook-Air C % gcc gauss.c -o gauss.out
[Cecilia.Torres@Laus-MacBook-Air C % ./gauss.out

Suma de los primeros n números

¿Cuántos números deseas sumar?
100
La suma de los primeros 100 números es: 5050
[Cecilia.Torres@Laus-MacBook-Air C % ./gauss.out

Suma de los primeros n números

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

```
C gauss.c > main()
1  #include<stdio.h>
2  int main()
3  {
4      //Declarar variables
5      int n,res;
6
7      //Mensaje de bienvenida
8      printf("\n\n\t\t\tSuma de los primeros n números\n\n");
9
10     //Solicitar el número de elementos a sumar
11     printf("¿Cuántos números deseas sumar?\n");
12     scanf("%d",&n);
13
14     //Sumar los n números
15     res=0;
16     for(int i=1;i<=n;i++)
17     {
18         res=res+i;
19     }
20
21     //Mostrar el resultado
22     printf("La suma de los primeros %d números es: %d\n",n,res);
23
24     return 0;
25 }
```

A-4 Usar la directiva define para elaboración de código versátil.

```
Cecilia.Torres@Laus-MacBook-Air C % gcc p9.c -o p9.out
Cecilia.Torres@Laus-MacBook-Air C % ./p9.out
Ingrese un número par 1: 34
Ingrese un número par 2: 5
El número insertado no es par.
Ingrese un número par 2: 78
Ingrese un número par 3: 22
Ingrese un número par 4: 96
Ingrese un número par 5: 2
La suma de los números es: 232
```

```
C p9.c > main()
1  #include <stdio.h>
2  /*
3  Este programa obtiene la suma de un
4  límites de números pares ingresados
5  */
6  #define LIMITE 5
7  int main ()
8  {
9      int enteroContador = 1;
10     int enteroNumero = 0;
11     int enteroSuma = 0;
12     while (enteroContador<=LIMITE)
13     {
14         printf("Ingrese un número par %d: ", enteroContador);
15         scanf("%d",&enteroNumero);
16         if (enteroNumero%2 !=0)
17         {
18             printf("El número insertado no es par.\n");
19             continue;
20         }
21         enteroSuma+=enteroNumero;
22         enteroContador++;
23     }
24     printf("La suma de los números es: %d\n",enteroSuma);
25     return 0;
26 }
```

Conclusión

Para terminar, el realizar esta práctica me ayudó a aprender y entender mejor las condiciones iterativas del lenguaje de programación C. Las dudas que me quedaban con respecto a cómo hacer ciertas partes de mi proyecto también se aclararon. Asimismo, logré comprender cómo y en qué casos se puede usar la variable *define*, con ayuda de los ejercicios mostrados en la práctica. Por último, me di cuenta de que debo de revisar con un poco más de cuidado mi código porque luego ponía dos puntos en lugar de punto y coma o se me pasaba definir alguna variable.

Fuentes de Consulta

Facultad de Ingeniería, (2018). *Guía práctica de estudio 09: Estructuras de selección*. Recuperado el 6 de diciembre del 2020.