



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ingeniería Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 20

No de Práctica: 8^o

Integrante: Yushua Haza Jibaja

*No. de Equipo de
cómputo empleado:*

No. de Lista o 22

Semestre: 2022 - 1

Fecha de entrega: 28 de septiembre de 2021

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.

Desarrollo de la práctica:

1-Estructura repetitiva while

El nuevo comando es while que es el equivalente a comando Mientras, no lleva “.”
Para comprobar y hacer uso de este nuevo comando, se añadió el programa 1.c de la práctica y se probó su ejecución

Prog 01

```
#include <stdio.h>^M
int main()^M
{^M
    int num, cont = 0;^M
    printf("\a----- Tabla de multiplicar -----\\n");^M
    printf("Ingrese un número: \\n");^M
    scanf("%d", &num);^M
    printf("La tabla de multiplicar del %d es:\\n", num);^M
    while (++cont <= 10)^M
        printf("%d x %d = %d\\n", num, cont, num*cont);^M
    return 0;^M
}
~
~
~
```

```
~
[root@localhost ~]# gcc Prog01Yushua.c -o Prog01Yushua.out
[root@localhost ~]# ./Prog01Yushua.out
----- Tabla de multiplicar -----
Ingrese un número:
```



```
[root@localhost ~]# gcc Prog01Yushua.c -o Prog01Yushua.out
[root@localhost ~]# ./Prog01Yushua.out
----- Tabla de multiplicar -----
Ingrese un número:
5
La tabla de multiplicar del 5 es:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
[root@localhost ~]#
```

Posteriormente se explico con el programa de la práctica 2.c, que el While puede generar un ciclo infinito si no se le agrega un comando Break y si su valor no es 0 o neutro, esto porque no hay nada que impida que el while siempre se cumpla, a lo cual se repetirá constantemente.

Programa2.c

```
#include <stdio.h>
int main()
{
    /* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero.
       El siguiente es un ciclo infinito porque la condición siempre es verdadera.
       Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son
       opcionales.*/

    while (100)
    {
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
    }
    return 0;
}
```

Estructura Do While

Esta estructura funciona para hacer y luego preguntar, de ello su nombre Hacer Mientras.

Posteriormente probamos la ejecución del programa 3.c para comprobar el funcionamiento de esta misma estructura

Prog2

```
[root@localhost ~]# gcc Prog02Yushua.c -o Prog02Yushua.out
[root@localhost ~]# ./Prog02Yushua.out
```

```
[root@localhost ~]# gcc Prog02Yushua.c -o Prog02Yushua.out
[root@localhost ~]# ./Prog02Yushua.out
      Suma de calificaciones
Ingrese la calificación:
7
¿Desea sumar otra? S/N
6
El promedio de las calificaciones ingresadas es: 7.000000
[root@localhost ~]#
```



Ejercicio 1

```
[root@localhost ~]# gcc Prog02Yushua.c -o Prog02Yushua.out
[root@localhost ~]# ./Prog02Yushua.out
      Suma de calificaciones
Ingrese la calificación:
7
¿Desea sumar otra? S/N
s
      Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
S
      Suma de calificaciones
Ingrese la calificación:
5
¿Desea sumar otra? S/N
S
      Suma de calificaciones
Ingrese la calificación:
10
¿Desea sumar otra? S/N
N
El promedio de las calificaciones ingresadas es: 7.500000
[root@localhost ~]#
```

Ejercicio 2

```
[root@localhost ~]# gcc Prog02AYushua.c -o Prog02AYushua.out
[root@localhost ~]# ./Prog02AYuhua.out
sh: ./Prog02AYuhua.out: not found
[root@localhost ~]# ./Prog02AYushua.out
    Suma de calificaciones
Ingrese la calificación:
5
¿Desea sumar otra? S/N
s
El promedio de las calificaciones ingresadas es: 5.000000
[root@localhost ~]#
```

Al remover la línea de código de limpieza de buffer, el programa es incapaz de agregar más calificaciones

Programa do while días de la semana

Se solicito que usásemos el programa usado para la práctica 7 y que a este se agregase un ciclo do while, para ello tuvimos que agregar dichos comandos al inicio y final del programa.

```
#include <stdio.h> //Yushua
int main()
{
    // Los valores de una enumeración son enteros y constantes
    enum diasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES,
    SABADO, DOMINGO};
    int op;
    do
    {

        printf("Ingrese el día de la semana.\n");
        printf("1) Lunes\n");
        printf("2) Martes\n");
        printf("3) Miércoles\n");
        printf("4) Jueves\n");
        printf("5) Viernes\n");
        printf("6) Sábado\n");
        printf("7) Domingo\n");
        scanf("%d", &op);
        switch(op-1)
        {
```

```

case LUNES:
case MARTES:
printf("Inicio de semana.\n");
break;
case MIERCOLES:
printf("Mitad de semana.\n");
break;
case JUEVES:
printf("¡Casi inicia el fin de semana!\n");
break;
case VIERNES:
case SABADO:
printf("¡Fin de semana!\n");
break;
case DOMINGO:
printf("Día de descanso.\n");
break;
// No se necesita default
}
}
while (op !=8);
}

```

```

[root@localhost ~]# gcc Prog03Yushua.c -o Prog03Yushua.out
[root@localhost ~]# ./Prog03Yushua.out
Ingrese el da de la semana.
1) Lunes
2) Martes
3) Mircoles
4) Jueves
5) Viernes
6) Sbado
7) Domingo
1
Inicio de semana.
Ingrese el da de la semana.
1) Lunes
2) Martes
3) Mircoles
4) Jueves
5) Viernes
6) Sbado
7) Domingo
8
[root@localhost ~]#

```



Programa 5.c

Se explico que en el copiado y pegado de este programa desde C++ existía la posibilidad de que no compilase debido al "loop" que contiene la codificación, sin embargo es completamente operable en la consola de Linux

```
[root@localhost ~]# gcc Prog05Yushua.c -o Prog05Yushua.out
[root@localhost ~]# ./Prog05Yushua.out
    Promedio de calificaciones

Ingrese la calificacin del alumn 1
9

Ingrese la calificacin del alumn 2
8

Ingrese la calificacin del alumn 3
7

Ingrese la calificacin del alumn 4
9

Ingrese la calificacin del alumn 5
9

El promedio de las calificaciones ingresadas es: 8.400000
[root@localhost ~]#
```

Ejercicio 1

Se solicito que cambiásemos la cantidad de calificaciones que se van a ingresar a solo 3 (alumnos), a lo cual solo fue necesario cambiar la variable de cantidad de alumnos que antes era 5 a 3.

```
[root@localhost ~]# gcc Prog05Yushua.c -o Prog05Yushua.out
[root@localhost ~]#
[root@localhost ~]# ./Prog05Yushua.out
    Promedio de calificaciones

Ingrese la calificacin del alumn 1
4

Ingrese la calificacin del alumn 2
9

Ingrese la calificacin del alumn 3
9

El promedio de las calificaciones ingresadas es: 7.333333
[root@localhost ~]#
```

Directiva Define

Define funciona como una declaración para 2 propósitos explicados:

- 1-Definir una variable constante con un valor pero ahora con un nombre, esta nunca cambiará y siempre será el nombre el que conecta con el valor
- 2-Definir a un texto para referirse a un comando, por ejemplo, #Define p printf, que ahora permite que podamos escribir p y que esta denote que hay un comando printf

El ejemplo ejercicio que se nos solicito elaborar fue el siguiente
Cambiar el define MAX 5 a otro valor y otro nombre

```
#include <stdio.h>
#define RAD 4
int main ()
{
    int arreglo[RAD], cont;
    for (cont=0; cont<RAD; 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<RAD; cont++)
    {
        printf("%d\t", arreglo[cont]);
    }
    printf("]\n");
    return 0;
}
```

```
[root@localhost ~]# gcc Prog06Yushua.c -o Prog06Yushua.out
[root@localhost ~]# ./Prog06Yushua.c
sh: ./Prog06Yushua.c: Permission denied
[root@localhost ~]# ./Prog06Yushua.out
Ingrese el valor 1 del arreglo: 3
Ingrese el valor 2 del arreglo: 6
Ingrese el valor 3 del arreglo: 7
Ingrese el valor 4 del arreglo: 8
El valor ingresado para cada elemento del arreglo es:
[3      6      7      8      ]
[root@localhost ~]#
```



Ejercicio para cambio de estructura

Para este ejercicio se nos solícito agregar la funcionalidad de volver a ejecutar el programa a través de una estructura do-while

```
#include <stdio.h>
#include <locale.h>
int main()
{
    {setlocale(LC_ALL, "");

    int y, num, cont = 0;
    do{
        printf("\a----- Tabla de multiplicar -----\\n");
        printf("Ingrese un número: \\n");
        scanf("%d", &num);
        printf("La tabla de multiplicar del %d es:\\n", num);
        while (++cont <= 10)
            printf("%d x %d = %d\\n", num, cont, num*cont);
        printf("¿Desea volver a ejecutar?\\n1) Si\\n2) No\\n");
        scanf("%i", &y);
    }
    while(y==1);
    return 0;
}
```

```
[root@localhost ~]# gcc Prog07Yushua.c -o Prog07Yushua.out
[root@localhost ~]# ./Prog07Yushua.out
----- Tabla de multiplicar -----
Ingrese un número:
6
La tabla de multiplicar del 6 es:
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
¿Desea volver a ejecutar?
1)Si
2)No
Si
[root@localhost ~]#
```

Conclusión

Las prácticas van aumentando la complejidad de los comandos a usar y aumentando la intensidad de los ejercicios, sin embargo, sigue siendo muy clara y concisa respecto a los objetivos y cumple finalmente con ayudar y administrar herramientas al alumno para que entienda y sepa como operar las nuevas estructuras en C++ While, Do-While, For y Define, equivalentes a estructuras previamente vistas en PSE int salvo la herramienta Define.

Tarea

1.- Del ejercicio de página 154 agregar condición a la división, si hay división entre cero pedir otro valor al usuario.

Para este programa usé la codificación del programa 4.c de la práctica, en el cual es su case 4 del ciclo switch modifiqué algunas líneas para que fuera posible que el programa solicitase valores distintos de 0 para los números

Sin embargo, en consola no se ejecuta completamente bien, pues el resultado es erróneo, pero en C++ se ejecuta correctamente

Codificación

```
#include <stdio.h>
#include <wchar.h>
#include <locale.h>
#include <stdlib.h>
//Yushua
int main()
{
    int op, uno, dos, h, j, k;

    printf(" --- Calculadora ---\n");
    printf("\n¿Qué desea hacer\n");
    printf("1) Sumar\n");
    printf("2) Restar\n");
    printf("3) Multiplicar\n");
    printf("4) Dividir\n");
    printf("5) Salir\n");
    scanf("%d", &op);
    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("Division\n");
printf("Ingresa el núm 1\n");
scanf("%d",&h);
if (h==0) {
printf("Ingresa un valor mayor o menor a 0\n");
scanf("%d",&h);
}
printf("Ingresa el numero 2\n");
scanf("%d",&j);
if (j==0) {
printf("Ingresa un valor mayor o menor a 0\n");
scanf("%d",&j);
(k=(h/j));
printf("%d/%d=%d\n",h,j,((double)h/j));
}
else
printf("%d/%d=%.21f\n",h,j,((double)h/j));
break;
case 5:
printf("\tSalir\n");
break;
default:
printf("\tOpción inválida.\n");

}
return 0;
}
```


2.- Modificar el programa para que permita solicitar al usuario el Número de alumnos a evaluar empleando un ciclo while en lugar del ciclo for.

```
#include <stdio.h>

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("\nIngrese la calificación del alumn %d\n", indice+1);

        scanf("%f",&realCalif);

        realPromedio += realCalif;

    }

    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",

    realPromedio/enteroNumAlumnos);

    return 0;

}
```

La codificación me ha quedado de la siguiente manera

```
#include <stdio.h>

int main (){
    int enteroNumAlumnos = 5, indice = 0;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    indice = 0;
    while (indice < enteroNumAlumnos)
    {
        printf("\nIngrese la calificación del alumn %d\n", indice+1);
```

```

scanf("%f",&realCalif);
indice++;
realPromedio += realCalif;
}
printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
realPromedio/enteroNumAlumnos);
return 0;
}

```

```

Prog07Yushua.c  [*] Untitled4  Prog05Yushua.c  Prog06Yushua.c  Prog08Yushua.c  ProgT2Yushua.c
1  #include <stdio.h>
2  int main (){
3      int enteroNumAlumnos = 5, indice = 0;
4      float realCalif = 0.0, realPromedio = 0.0;
5      printf("\tPromedio de calificaciones\n");
6      indice = 0;
7      while (indice < enteroNumAlumnos)
8      {
9          printf("\nIngrese la calificación del alumn %d\n", indice+1);
10         scanf("%f",&realCalif);
11         indice++;
12         realPromedio += realCalif;
13     }
14     printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
15         realPromedio/enteroNumAlumnos);
16     return 0;
17 }

```

Y la prueba de ejecución de prueba fue exitosa

```

[root@localhost ~]# gcc ProgT2Yushua.c -o ProgT2Yushua.out
[root@localhost ~]# ./ProgT2Yushua.out
    Promedio de calificaciones

Ingrese la calificacion del alumn 1
4

Ingrese la calificacion del alumn 2
5

Ingrese la calificacion del alumn 3
6

Ingrese la calificacion del alumn 4
7

Ingrese la calificacion del alumn 5
8

El promedio de las calificaciones ingresadas es: 6.000000
[root@localhost ~]#

```

3.- Indica que realiza el siguiente programa:

```
#include <stdio.h>

int main(){

while (100) {

printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");

continue;

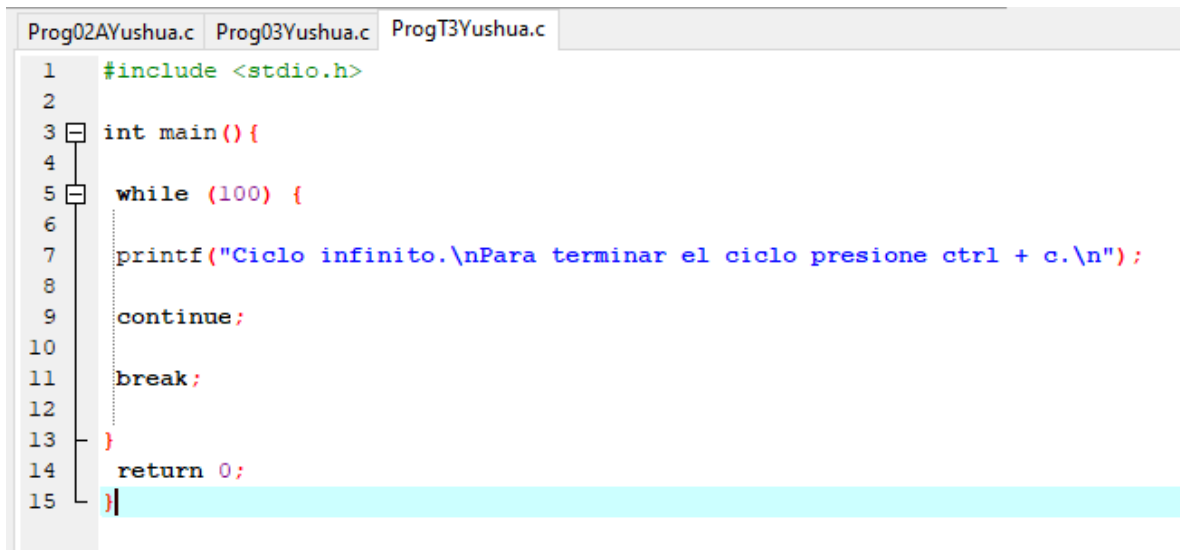
break;

}

return 0;

}
```

El programa en ejecución de C++, a pesar de tener un comando break, este se anula y supongo que sucede por el comando continue que esta previo al comando break, el programa imprime infinitamente la línea de texto: " Para terminar el ciclo presione ctrl + c", finalizando de imprimir solo si se teclea control+c



```
1  #include <stdio.h>
2
3  int main(){
4
5  while (100) {
6
7  printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
8
9  continue;
10
11 break;
12
13 }
14 return 0;
15 }
```


C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe




[illegible]

Sucede lo mismo para la ejecución en la consola de Linux

[illegible]

```
[root@localhost ~]# gcc ProgT3Yushua.c -o ProgT3Yushua.out
[root@localhost ~]# ./ProgT3Yushua.out
Ciclo infinito.
Para terminar el ciclo presione ctrl + c.
Ciclo infinito.
Para terminar el ciclo presione ctrl + c.
Ciclo infinito.
Para terminar el ciclo presione ctrl + c.
Ciclo infinito.
```

4.- Completar la columna de codificación.

Estructura	Diagrama	Pseudocódigo	Codificación
Iterativa Mientras		MIENTRAS que <expresión lógica> Instrucciones FIN DEL MIENTRAS	
Iterativa Hacer Mientras		HACER Instrucciones MIENTRAS que <expresión lógica>	
Iterativa Para		PARA (valor inicial; <expresión lógica>; incremento/decremento) Instrucciones FIN PARA	

El ciclo MIENTRAS es equivalente en C++ al ciclo **While**

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

un ejemplo de programa que hace uso de este comando while es:

```
#include <stdio.h>  
int main()  
{  
    int num, cont = 0;  
    printf("\a----- Tabla de multiplicar -----\\n");  
    printf("Ingrese un número: \\n");  
    scanf("%d", &num);  
    printf("La tabla de multiplicar del %d es:\\n", num);  
    while (++cont <= 10)  
        printf("%d x %d = %d\\n", num, cont, num*cont);  
    return 0;  
}
```

El ciclo HACER MIENTRAS es equivalente en C++ al ciclo **Do-While**

```
do {  
    /*  
    Bloque de código que se ejecuta  
    por lo menos una vez y se repite  
    mientras la expresión lógica sea  
    verdadera.  
    */  
} while (expresión_lógica);
```

un ejemplo de programa que hace uso de este comando do-while es:

```
#include <stdio.h>  
int main ()  
{  
    char op = 'n';  
    double sum = 0, calif = 0;  
    int veces = 0;  
    do  
    {  
        printf("\tSuma de calificaciones\n");  
        printf("Ingrese la calificación: ");  
        scanf("%lf", &calif);  
        veces++;  
        sum = sum + calif;  
        printf("¿Desea sumar otra? S/N\n");  
        scanf("%c", &op);  
        getchar();  
    }  
    while (op == 'S' || op == 's');  
    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);  
    return 0;  
}
```

El ciclo Para es equivalente en C++ al ciclo **for**

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {  
    /*  
        Bloque de código  
        a ejecutar  
    */  
}
```

un ejemplo de programa que hace uso de este comando for es:

```
#include <stdio.h>  
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("\nIngrese la calificación del alumn %d\n", indice+1);  
        scanf("%f",&realCalif);  
        realPromedio += realCalif;  
    }  
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",  
        realPromedio/enteroNumAlumnos);  
    return 0;  
}
```

Referencia

-Facultad de Ingenieria. (2021). Laboratorio de computacion Sala A y B.
Recuperado el 10 de noviembre de 2021 de <http://lcp02.fi-b.unam.mx/>