	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	Adrian Ulises Mercado Martínez
Asignatura:	Fundamentos de programación
Grupo:	07
No de Práctica(s):	08
Integrante(s):	Arredondo Granados Gerardo Casillas Herrera Leonardo Didier Diaz Gonzalez Rivas Angel Iñaqui Galván Castro Martha Selene
No. de Equipo de cómputo empleado:	20
No. de Lista o Brigada:	04
Semestre:	2022-1
Fecha de entrega:	25 de diciembre del 2021
Observaciones:	

CALIFICACIÓN:_____

Práctica 8
Estructuras de repetición

Índice

Referencias 13

Objetivo

El alumno elaborará programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición.

Introducción

En lenguaje C tenemos ciertas estructuras de repetición, estas también son llamadas cíclicas, iterativas o bucles, el objetivo de todas estás estructuras es ejecutar un conjunto de instrucciones de manera repetida, de aquí se les da su nombre de cíclicas, mientras que la expresión que se esté evaluando sea verdadera.

Como se menciona anteriormente, en el lenguaje C tenemos las estructuras de repetición While, do-while y for.

Estructura de control repetitiva while.
Esta expresión a diferencia de do-while, lo primero que hace es validar la expresión lógica que nosotros le hayamos dado al programa, en caso de ésta ser verdadera, procede a ejecutar el bloque de instrucciones de la estructura, una vez que termina de ejecutar este bloque, regresa a validar la condición nuevamente hasta que la condición sea falsa. La manera en que escribimos la estructura de control repetitiva while es la siguiente:

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

En caso de que el bloque de código solo tenga una sentencia, no es necesario el uso de llaves.

Programa1.c

- Este programa genera la tabla de multiplicar de un número.

practica8 > C prog1.c > main()

```c
#include <stdio.h>
int main()
{
    int num, cont = 0;
    printf("\a----- Tabla de multiplicar -----\n");
    printf("Ingrese un numero: \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);
        cont++;
    }
    return 0;
}
```

```
----- Tabla de multiplicar -----
Ingrese un numero:
5
La tabla de multiplicar del 5 es:
5 x 0 = 0
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

Press any key to continue . . .
```

Programa2.c

- Este programa genera un ciclo infinito.

```
practica8 > C prog2.c > main()
#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 (5)
    {
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
    }
}
```

```
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.
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.
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.
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.
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.
```

Estructura de control repetitiva do-while

Esta estructura cíclica, a diferencia de la estructura while, lo primero que hace es ejecutar el bloque de código y después de haberlo ejecutado, valida la condición, en otras palabras, con la estructura do-while, siempre se va a ejecutar el bloque de código mínimo una vez.La forma en que escribimos una estructura do-while es la siguiente:

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

Programa3.c

- Este programa obtiene el promedio de calificaciones que el usuario ingrese.

practica8 > C prog3.c > main()

```c
#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 calificacion:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;
        printf("¿Desea 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;
}
```

```
        Suma de calificaciones
Ingrese la calificacion:
10
¿Desea sumar otra? S/N
s
        Suma de calificaciones
Ingrese la calificacion:
5
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 7.500000

Press any key to continue . . .
```

Programa4.c

- Este programa genera una calculadora básica

```c
practica8 > C prog4.c > main()
#include <stdio.h>
int main()
{
    int op, uno, dos;
    do
    {
        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 numeros 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 numeros 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 numeros 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 numeros a dividir separados por comas\n");
            scanf("%d, %d", &uno, &dos);

            printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
            break;
        case 5:
            printf("\tSalir\n");
            break;
        default:
            printf("\tOpcion invalida.\n");
        }
    }

    while (op != 5);
}
```

```
 --- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
3
        Multiplicar
Introduzca los numeros a multiplicar separados por comas
10,5
10 * 5 = 50
 --- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
        Salir

Press any key to continue . . .
```

Estructura de control de repetición for

La estructura de repetición for permite realizar varias repeticiones. La forma en que escribimos la estructura for es la siguiente:

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

Como se puede ver, la estructura for es diferente a las demás que hemos visto, al momento de ejecutar el programa, se ejecutan 3 acciones básicas, dos antes y una después de ejecutar el bloque de código. La primera acción de estas 3 es la iniciación, en donde se pueden definir variables e inicializar sus valores. La segunda es una expresión lógica, esta es la expresión que se evalúa y en caso de que sea verdadera, se continúa con la estructura. La tercera es un conjunto de operaciones que se ejecutarán al terminar la ejecución del bloque de código.

Programa5.c

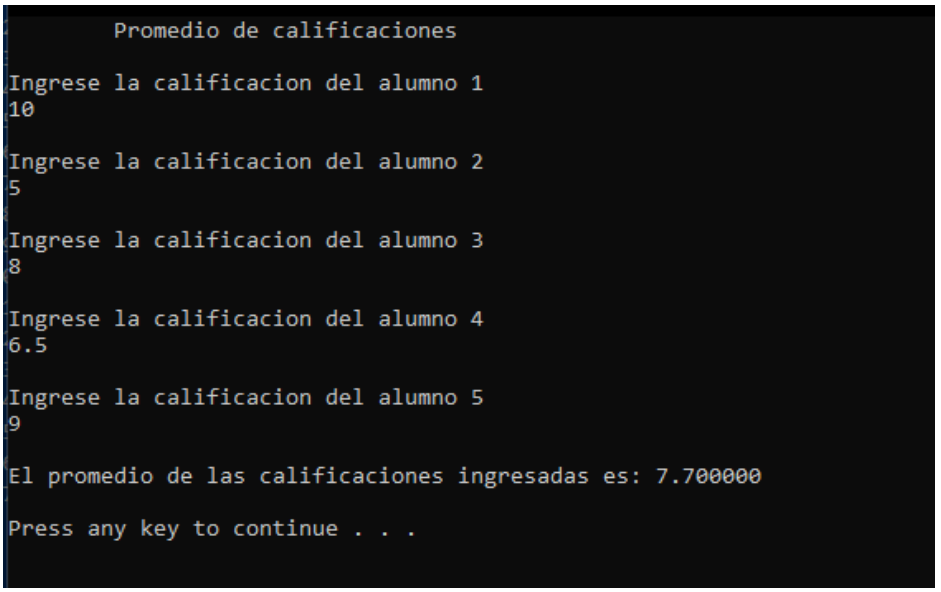
- Este programa genera un arreglo unidimensional de 5 elementos y accede a cada elemento del arreglo a través de un ciclo for.

practica8 > C prog5.c > ...

```c
#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 calificacion 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;
}

```



Desarrollo

Ejercicios 1 :

1. Elaborar un programa para calcular el factorial de un número
Datos de entrada: número entero largo.
Datos de salida: número entero largo.
Dominio: todos los números enteros.

Pseudocódigo:

1. **Inicio**
2. Escribir “Ingrese el número del que desea obtener factorial”
3. Leer num
4. Para i← 1 Hasta i<=num Con Paso 1 Hacer
5. fac← fac*i
6. Fin para
7. Escribir “El factorial de “ num “ es “ fac
8. **Fin**

Código:

```c
#include <stdio.h>

int main(){

    int num, i, fac=1;
    printf("Ingrese el número del que desea obtener factorial\n");
    scanf("%d", &num);

    for(i=1; i<= num; i++){
        fac=fac*i;
    }

    printf("El factorial de %d es: %d", num, fac);
    return 0;
}
```

```
Ingrese el numero del que desea obtener factorial
5
El factorial de 5 es: 120
Press any key to continue . . .
```

Ejercicio 2.

Calcular la suma de los primeros n números con un for
s= 1 + 2+ 3+ ...+n

Datos de entrada: número entero
Datos de salida: número entero
Dominio: todos los números enteros

Pseudocódigo:

1. **Inicio**
2. Escribir “Ingresa un numero”
3. Leer n
4. Para cont← 1 Hasta cont<= n Con Paso 1 Hacer
5. num← num+cont
6. Fin para
7. Escribir “La suma de los números desde 1 hasta “ n “ es “ num
8. **Fin**

Código:

```
int main(){
    int numero=0 ,x ,i;
    printf("Ingresa un número \n");
    scanf("%d", &x);

    for(i = 1; i <= x; i++){
        numero = numero + i;
    }
    printf("La suma de los números desde 1 hasta %d es: %d", x,
numero);
    return 0;
}
```

```
Ingresa un numero
10
La suma de los numeros desde 1 hasta 10 es: 55
Press any key to continue . . .
```

Ejercicio 3.

Elaborar un programa que determine si un número es primo o no
Datos de entrada: número entero
Datos de salida: número entero
Dominio: todos los números enteros

Pseudocódigo:

1. Inicio
2. Define n, cont como Entero
3. c← 0
4. Escribir “Ingresa el número que se desea conocer si es primo o no”
5. Leer n
6. Para cont ← 1 hasta cont <= n con paso 1 Hacer
7. Si n MOD cont = 0 Entonces
8. c += 1
9. Fin Si
10. Fin Para
11. Si c ← 2 Entonces
12. Escribir “El número ingresado es primo”
13. Sino
14. Escribir “El número ingresado no es primo”
15. Fin Si
16. Fin

Código:

```c
#include <stdio.h>


int main(){
    int x, cont, c=0;
    printf("Ingresa el número que se desea conocer si es primo o no");
    scanf("%d", &x);
    for(cont=1; cont <= x; cont++){
        if(x%cont == 0){
            c += 1;
        }
    }
    if(c == 2){
        printf("El número es primo\n");
    }else{
        printf("El número ingresado no es primo\n");
    }
    return 0;
}
```

```
Ingresa el n├╢mero que se desea conocer si es primo o no 13
El n├╢mero es primo
```

Conclusiones

En la realización de la práctica se revisaron los conceptos de las estructuras de repetición, así como su código en lenguaje C. Las estructuras de repetición necesitaban validar la expresión lógica para ejecutar un conjunto de instrucciones de forma repetida. Se realizaron ejercicios que requerían del uso de estructuras while, do-while y for para dar solución a los problemas analizados en esta práctica.

Conclusiones individuales

Arredondo Granados Gerardo:
A lo largo de esta práctica, nosotros como alumnos logramos elaborar programas en el lenguaje C con las estructuras de repetición o iteración vistas en esta práctica, mientras que en la clase con el profesor igualmente se vieron las diferentes estructuras cíclicas más a fondo con los que el profesor nos explicó.

Cabe aclarar que los ejercicios propuestos por el profesor, específicamente nos pedían la resolución con alguna de las estructuras vistas, los ejercicios se concluyeron exitosamente con dichas estructuras; con eso podemos dar por cumplido el objetivo de la práctica.

Casillas Herrera Leonardo Didier: En toda la práctica estuvimos viendo las estructuras de repetición y se nos proporcionaron algunos ejemplos para poder realizar unos ejercicios con las estructuras cíclicas que se nos solicitaban o con la que fuera más óptima para resolverlos, por lo que se cumplió el objetivo de esta práctica.

Diaz Gonzalez Rivas Angel Iñaqui: Al momento de realizar la práctica analizamos las estructuras de repetición while, do-while y for para poder realizar la ejecución de instrucciones de forma iterativa a partir de la expresión lógica. En los ejercicios, para proporcionar una solución al problema, se escribió el pseudocódigo y el código en lenguaje C, utilizando las estructuras, con datos de entrada y salida de tipo entero.

Galvan Castro Martha Selene:como conclusión se puede decir que se cumplió con el objetivo de la práctica ya que con los programas se pudo en práctica el uso de estructuras repetitivas como lo don el for ya que para la resolución de estos programas era necesario es uso de estos.

Referencias

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