	Carátula para entrega de prácticas	Código	
		Versión	02
		Página	1/1
		Sección ISO	
		Fecha de emisión	25 de junio de 2014
Secretaría/División: División de Ingeniería Eléctrica		Área/Departamento: Laboratorios de computación salas A y B	

Laboratorio de computación salas A y B

Profesor: Claudia Rodríguez Espino

Asignatura: Fundamentos de Programación

Grupo: 1102

No de Práctica(s): 9

Integrante(s): Carrasco Mendoza Jennifer

Semestre: 2018-I

Fecha de entrega: 20-October-2017

Observaciones:

CALIFICACIÓN: _____

Práctica 09:

Estructuras de repetición

Objetivo:

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

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*.
- Usar la directiva *define* para elaboración de código versátil.

Introducción

ACTIVIDAD1:

Código estructura de repetición do-while

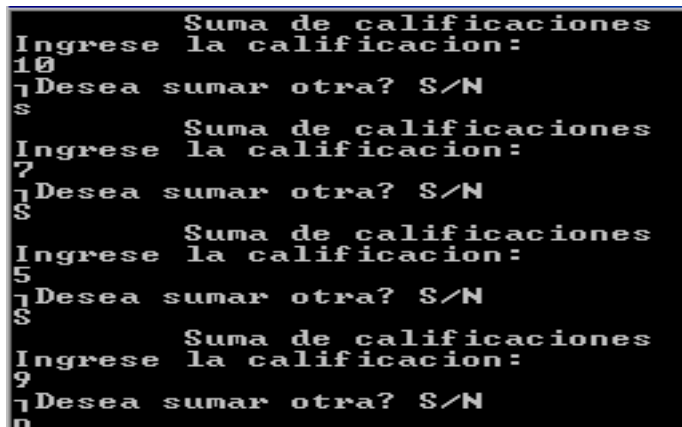
```
#include <stdio.h>
```

```
/*
```

Este programa obtiene el promedio de calificaciones ingresadas por el usuario. Las calificaciones se leen desde la entrada estándar (teclado). La inserción de calificaciones termina cuando el usuario presiona una tecla diferente de 'S' o 's'.

```
*/
```

```
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("¿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 calificación:  
10  
¿Desea sumar otra? S/N  
S  
Suma de calificaciones  
Ingrese la calificación:  
7  
¿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:  
9  
¿Desea sumar otra? S/N  
n
```

Se agregó la función *tolower* (variable) con su respectiva biblioteca (*ctype.h*). Esta función nos permite modificar a minúscula cualquier carácter que ingrese el usuario (mayúscula o minúscula) para poderla asignar a los casos establecidos.

ACTIVIDAD 2:

Código estructura de repetición for

```
#include <stdio.h>
/*
 * Este programa genera un promedio de 5 elementos y
 * accede a cada elemento a través de un ciclo for.
 */
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 alumno %d\n", indice+1);
        scanf("%f",&realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
    realPromedio/enteroNumAlumnos);
    return 0;
}
```

```
Promedio de calificaciones
Ingrese la calificacion del alumno 1
10
Ingrese la calificacion del alumno 2
5
Ingrese la calificacion del alumno 3
7
Ingrese la calificacion del alumno 4
8
Ingrese la calificacion del alumno 5
9
El promedio de las calificaciones ingresadas es: 7.800000
```

ACTIVIDAD 3:

Código 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]);
}
```

```
Ingrese el valor 1 del arreglo: 6
Ingrese el valor 2 del arreglo: 100
Ingrese el valor 3 del arreglo: 89
Ingrese el valor 4 del arreglo: 76
Ingrese el valor 5 del arreglo: 54
El valor ingresado para cada elemento del arreglo es:
[6      100    89      76      54      ]
```

```

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;
}

```

Se ocupan valores numéricos, así como palabras o letras; cabe precisar que ocupamos define p printf para ahorrarnos línea de código en el programa.

ACTIVIDAD 4:

Código operadores.

```

#include <stdio.h>
#define VALOR_MAX 5

```

```

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

```

```

int main (){

int enteroSuma = 0;
int enteroNumero = 0;
int enteroContador = 0;

```

```

Ingrese un numero:20
Ingrese un numero:2
Ingrese un numero:9
Ingrese un numero:5
Ingrese un numero:6
El valor de la suma es: 42

```

```

while (enteroContador < VALOR_MAX){
printf("Ingrese un número:");
scanf("%d", &enteroNumero);
enteroSuma += enteroNumero;
enteroContador++;

if (enteroSuma > 50){
printf("Se rebasó la cantidad límite.\n");
break;
}
}

```

```

Ingrese un numero:50
Ingrese un numero:4
Se rebaso la cantidad limite.
El valor de la suma es: 54

```

```

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

}

```

Tablas de multiplicar del 1 al 10 (estructura de repetición for)

```
#include <stdio.h>
int a, m, x;
main()
{
for(a=1;a<=10;a++)
{
printf("\n\t\t\tTabla de multiplicar del %d\n",a);
for(x=1; x<11; x++)
{
m=a;
m=(m*x);
printf("%d X %d = %d\t",a,x,m);
}
}
}
```

```

1 x 1 = 1      1 x 2 = 2      1 x 3 = 3      1 x 4 = 4      1 x 5 = 5
1 x 6 = 6      1 x 7 = 7      1 x 8 = 8      1 x 9 = 9      1 x 10 = 10

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

3 x 1 = 3      3 x 2 = 6      3 x 3 = 9      3 x 4 = 12     3 x 5 = 15
3 x 6 = 18     3 x 7 = 21     3 x 8 = 24     3 x 9 = 27     3 x 10 = 30

4 x 1 = 4      4 x 2 = 8      4 x 3 = 12     4 x 4 = 16     4 x 5 = 20
4 x 6 = 24     4 x 7 = 28     4 x 8 = 32     4 x 9 = 36     4 x 10 = 40

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

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

7 x 1 = 7      7 x 2 = 14     7 x 3 = 21     7 x 4 = 28     7 x 5 = 35
7 x 6 = 42     7 x 7 = 49     7 x 8 = 56     7 x 9 = 63     7 x 10 = 70

8 x 1 = 8      8 x 2 = 16     8 x 3 = 24     8 x 4 = 32     8 x 5 = 40
8 x 6 = 48     8 x 7 = 56     8 x 8 = 64     8 x 9 = 72     8 x 10 = 80

9 x 1 = 9      9 x 2 = 18     9 x 3 = 27     9 x 4 = 36     9 x 5 = 45
9 x 6 = 54     9 x 7 = 63     9 x 8 = 72     9 x 9 = 81     9 x 10 = 90

10 x 1 = 10    10 x 2 = 20    10 x 3 = 30    10 x 4 = 40    10 x 5 = 50
10 x 6 = 60    10 x 7 = 70    10 x 8 = 80    10 x 9 = 90    10 x 10 = 100

Process exited after 5.816 seconds with return value 0
Presione una tecla para continuar . . .
```

Programa 2:

Tablas de multiplicar del 1 al 10 (estructura repetitiva while)

```
#include <stdio.h>
int a=1, m, x;
main()
{
while(a<=10)
{
printf("\n\t\t\tTabla de multiplicar del %d\n",a);
for(x=1; x<11; x++)
{
m=a;
m=(m*x);
printf("%d X %d = %d\t",a,x,m);
}
a++;
}
}
```

```

1 x 1 = 1      1 x 2 = 2      1 x 3 = 3      1 x 4 = 4      1 x 5 = 5
1 x 6 = 6      1 x 7 = 7      1 x 8 = 8      1 x 9 = 9      1 x 10 = 10

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

3 x 1 = 3      3 x 2 = 6      3 x 3 = 9      3 x 4 = 12     3 x 5 = 15
3 x 6 = 18     3 x 7 = 21     3 x 8 = 24     3 x 9 = 27     3 x 10 = 30

4 x 1 = 4      4 x 2 = 8      4 x 3 = 12     4 x 4 = 16     4 x 5 = 20
4 x 6 = 24     4 x 7 = 28     4 x 8 = 32     4 x 9 = 36     4 x 10 = 40

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

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

7 x 1 = 7      7 x 2 = 14     7 x 3 = 21     7 x 4 = 28     7 x 5 = 35
7 x 6 = 42     7 x 7 = 49     7 x 8 = 56     7 x 9 = 63     7 x 10 = 70

8 x 1 = 8      8 x 2 = 16     8 x 3 = 24     8 x 4 = 32     8 x 5 = 40
8 x 6 = 48     8 x 7 = 56     8 x 8 = 64     8 x 9 = 72     8 x 10 = 80

9 x 1 = 9      9 x 2 = 18     9 x 3 = 27     9 x 4 = 36     9 x 5 = 45
9 x 6 = 54     9 x 7 = 63     9 x 8 = 72     9 x 9 = 81     9 x 10 = 90

10 x 1 = 10    10 x 2 = 20    10 x 3 = 30    10 x 4 = 40    10 x 5 = 50
10 x 6 = 60    10 x 7 = 70    10 x 8 = 80    10 x 9 = 90    10 x 10 = 100

Process exited after 0.7239 seconds with return value 0
Presione una tecla para continuar . . .
```

Programa 3:

Tablas de multiplicar del 1 al 10 (estructura cíclica do-while)

```
#include <stdio.h>
int a=1, m, x;
main()
{
do
{
printf("\n\t\tTabla de multiplicar del %d\n",a);
for(x=1; x<11; x++)
{
m=a;
m=(m*x);
printf("%d X %d = %d\t",a,x,m);
}
a++;
}
while(a<=10);
}
```

Process exited after 0.5769 seconds with return value 0
Presione una tecla para continuar . . .

Programa 4:

Mostrar las potencias de un número ingresado de la 1 a la 10.

```
#include <stdio.h>
#include <math.h>
#define u 163

int p,n,e;

main()
{
printf("Introduzca un n%cmero: ", u);
scanf("%d",&n);
printf("\nPotencia del 1 al 10 del n%cmero %d", u,n);

for(e=1;e<11;e++)
{
p=(pow(n,e));
printf("\n%d^%d = %d", n,e,p);
}
}
```

Conclusiones:

El manejo de las estructuras repetitivas del Lenguaje C (do-while, while o for) es de gran ayuda para la optimización del código. Incluir estas estructuras en la elaboración de programas permite realizar sus iteraciones necesarias y su respectivo control.