



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de programación.

Grupo: 03

No de Práctica(s): 09

Integrante(s): Ingrid Ailin Girón Reyes

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

No. de Lista o Brigada: 18

Semestre: 2021-1

Fecha de entrega: enero2021

Observaciones:

CALIFICACIÓN: _____

Objetivos:

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 son las llamadas estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que

la expresión lógica a evaluar se cumpla (sea verdadera).

En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

Actividades:

Estructura de control repetitiva while

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

Código (estructura de repetición while)

```
#include <stdio.h>  
/*  
Este programa genera la tabla de multiplicar de un número dado.  
El número se lee desde la entrada estándar (teclado).  
*/  
int main(){  
    int num, cont = 0;  
    printf("\n----- 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;  
}
```

Código (estructura de repetición while)

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

Este programa genera un ciclo infinito.

```
*/  
int main(){  
// Al igual que en la estructura if-else  
// 0 -> falso  
// 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 de control repetitiva 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);
```

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;  
}
```

Código (estructura de repetición do-while)

```
#include <stdio.h>
/* Este programa genera una calculadora básica. */
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 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;
case 5:
printf("\tSalir\n");
break;
default:
printf("\tOpción inválida.\n");
}
} while (op != 5);
return 0;
}
```

Estructura de control de repetición for

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

```
*/
```

```
}
```

Código (estructura de repetición for)

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

```
/* Este programa genera un arreglo unidimensional de 5 elementos y  
* accede a cada elemento del arreglo 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 alumn %d\n", indice+1);
```

```
scanf("%f",&realCalif);
```

```
realPromedio += realCalif;
```

```
}
```

```
printf("\nEl promedio de las calificaciones ingresadas es: %f\n",  
realPromedio/enteroNumAlumnos);
```

```
return 0;
```

```
}
```

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]);
```

```
}
```

```
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;
```

```
}
```

Código (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 > 50){
printf("Se rebasó la cantidad límite.\n");
break;
printf("El valor de la suma es: %d\n", enteroSuma);
return 0;
}

```

Código (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;
}

```

Conclusión:

En conclusion, gracias a la práctica, aprendí elaborar programas en C para la resolución de problemas básicos que incluyan lasestructuras de repetición y la directiva define. La verdad es que no sabia nada sobre esto y me pareció sumamente interesante y util saberlo porque son cosas básicas que se necesitan para la programación, no solo en este lenguaje.