



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:

3

No de Práctica(s):

10

Integrante(s):

Ángel Joel Flores Torres

*No. de Equipo de
cómputo empleado:*

N/A

No. de Lista o Brigada:

14

Semestre:

1er Semestre

Fecha de entrega:

26 de Enero de 2021

Observaciones:

CALIFICACIÓN: _____

OBJETIVO

Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

ACTIVIDADES

- Revisar, a través de un depurador, los valores que va tomando una variable en un programa escrito en C, al momento de ejecutarse.
- Utilizando un depurador, revisar el flujo de instrucciones que se están ejecutando en un programa en C, cuando el flujo depende de los datos de entrada.

INTRODUCCIÓN

Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

Dependiendo de la herramienta usada para compilar el programa, si es de consola o de terminal, su uso y las funciones disponibles variarán.

RESULTADOS

Ejercicio 1

```
#include<stdio.h>
void main()
{
    int N, CONT, AS;
    AS=0;
    CONT=1;
    printf("TECLEA UN NUMERO: ");
    scanf("%i",&N);
    while(CONT<=N)
    {
        AS=(AS+CONT);
        CONT=(CONT+2);
    }
    printf("\nEL RESULTADO ES: %i\n",AS);
}
```

```
TECLEA UN NUMERO: 5
EL RESULTADO ES: 9
```

Funcionalidad del programa: El programa guarda un número, y le suma un contador siempre que el número sea mayor que el contador y después le suma 2 al contador.

Ejercicio 2

```
#include <stdio.h>
void main()
{
    int i, j;

    for(i=1; i<10; i++)
    {
        printf("\nTabla del %i\n", i);
        for(j=1; j==10; j++)
        {
            printf("%i X %i = %i\n", i, j, i*j);
        }
    }
}
```

Funcionalidad del programa: Se supone que el programa debe mostrar las tablas de multiplicar del 1 al 10

Solución:

```
#include<stdio.h>
void main()
{
    int i, j;

    for(i=1; i<=10; i++)
    {
        printf("\nTabla de multiplicar del %i\n",i);
        j=0;
        while(++j<=10)
            printf("%i x %i = %i\n",i, j, i*j);
    }
}
```

Tabla de multiplicar del 1

```
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
```

Tabla de multiplicar del 2

```
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
```

Tabla de multiplicar del 3

```
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
```

Tabla de multiplicar del 4

```
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
```

Tabla de multiplicar del 5

```
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
```

Tabla de multiplicar del 6

```
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
```

```
Tabla de multiplicar del 7
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
```

```
Tabla de multiplicar del 8
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
```

```
Tabla de multiplicar del 9
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
```

```
Tabla de multiplicar del 10
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
```

Ejercicio 3

```
#include <stdio.h>
#include <math.h>
void main()
{
    int K, X, AP, N;
    float AS;
    printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");
    printf("\nN=");
    scanf("%d",N);
    printf("X=");
    scanf("%d",X);
    K=0;
    AP=1;
    AS=0;
    while(K<=N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("SUM=%le",AS);
}
```

El error está en que no se está guardando los valores de las variables y por lo tanto el programa no se puede ejecutar, esto se resuelve colocando "&"

Solución:

```
#include<stdio.h>
#include <math.h>
void main()
{
    int K, X, AP, N;
    float AS;
    printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");
    printf("\nN=");
    scanf("%d",&N);
    printf("X=");
    scanf("%d",&X);
    K=0;
    AP=1;
    AS=0;
    while(K<N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("SUM=%le",AS);
}
```

```
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=5
X=8
SUM=2.970000e+002
```

CONCLUSIÓN

Como sabemos la depuración es uno de los aspectos más importantes en la programación, pues esta contribuye en la revisión línea por línea de todo el código y así verificar si existe algún error.

Considero que esta fue de las practicas que menos entendí, me costó bastante trabajo entender bien el proceso de depuración, pues como lo comentaba en la clase es un resultado muy subjetivo pues no dice a ciencia cierta donde se

encuentra el error, por suerte pude ver a “ojo” algunos de los errores que tenía cada programa y fueron corregidos. Esta práctica debería tener más énfasis o ser más detallados a la hora de su explicación pues considero que la depuración es algo muy importante en la vida de un programador.

REFERENCIAS

Colaboradores de Wikipedia. (s. f.). *Depuración de programas*. Wikipedia, la enciclopedia libre. Recuperado 26 de enero de 2021, de https://es.wikipedia.org/wiki/Depuraci%C3%B3n_de_programas

Caballero, A. (2018, 20 julio). *Depurar un Programa en Lenguaje C con GDB*. Reydes. http://www.reydes.com/d/?q=Depurar_un_Programa_en_Lenguaje_C_con_GDB