



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Claudia Rodríguez Espino

Asignatura: Fundamentos de Programación

Grupo: 1102

No de Práctica(s): Práctica número 9

Integrante(s): Martínez Ramírez Pablo César

Semestre: 2018-1

Fecha de entrega: 27/10/2017

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.

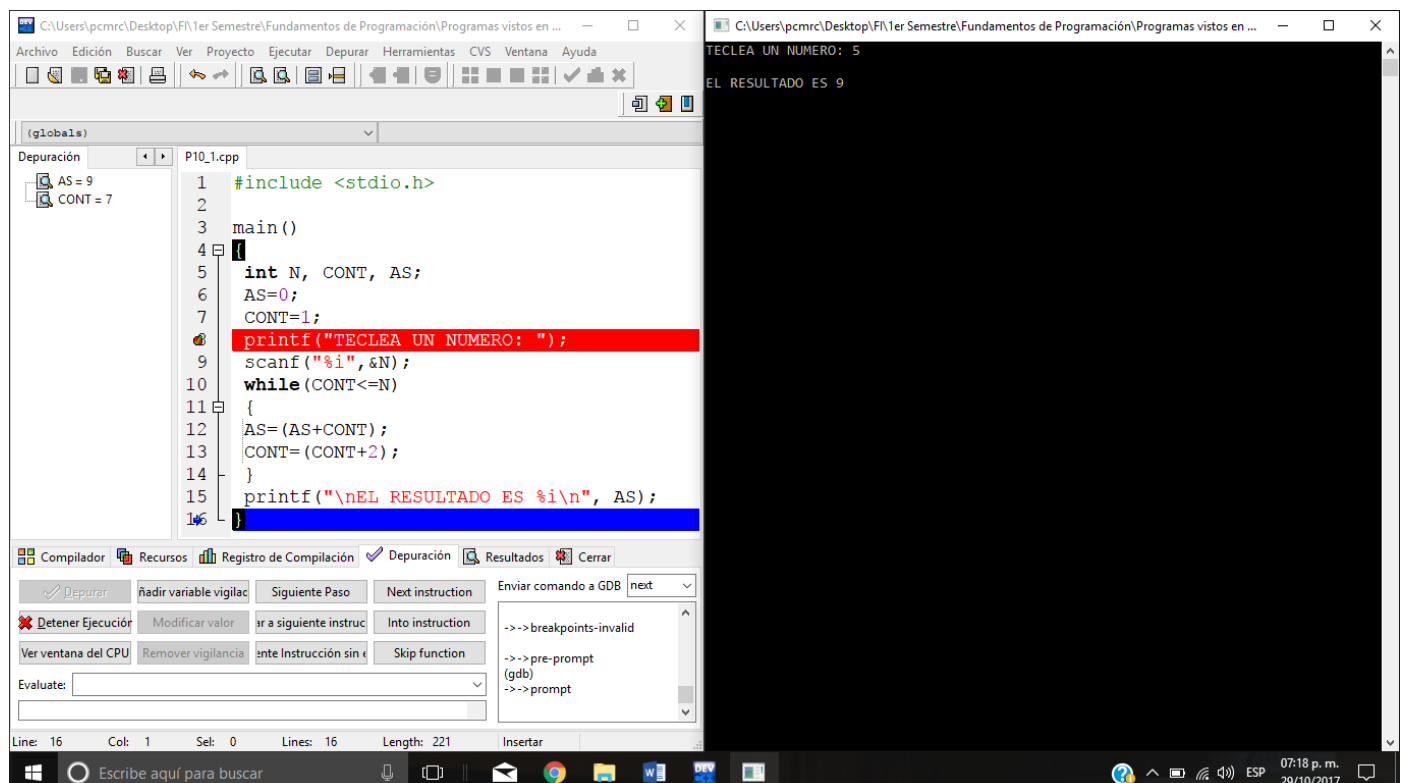
ACTIVIDADES EN LABORATORIO:

CÓDIGO FUENTE DEL PROGRAMA PROPUESTO 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);
}
```

DEPURACIÓN:



EL único error era el "void", ya que está puesto en el main.

El programa ya compilado y corregido dado un valor incrementa una variable mientras es igual o menor al contador que se incrementa de dos en dos.

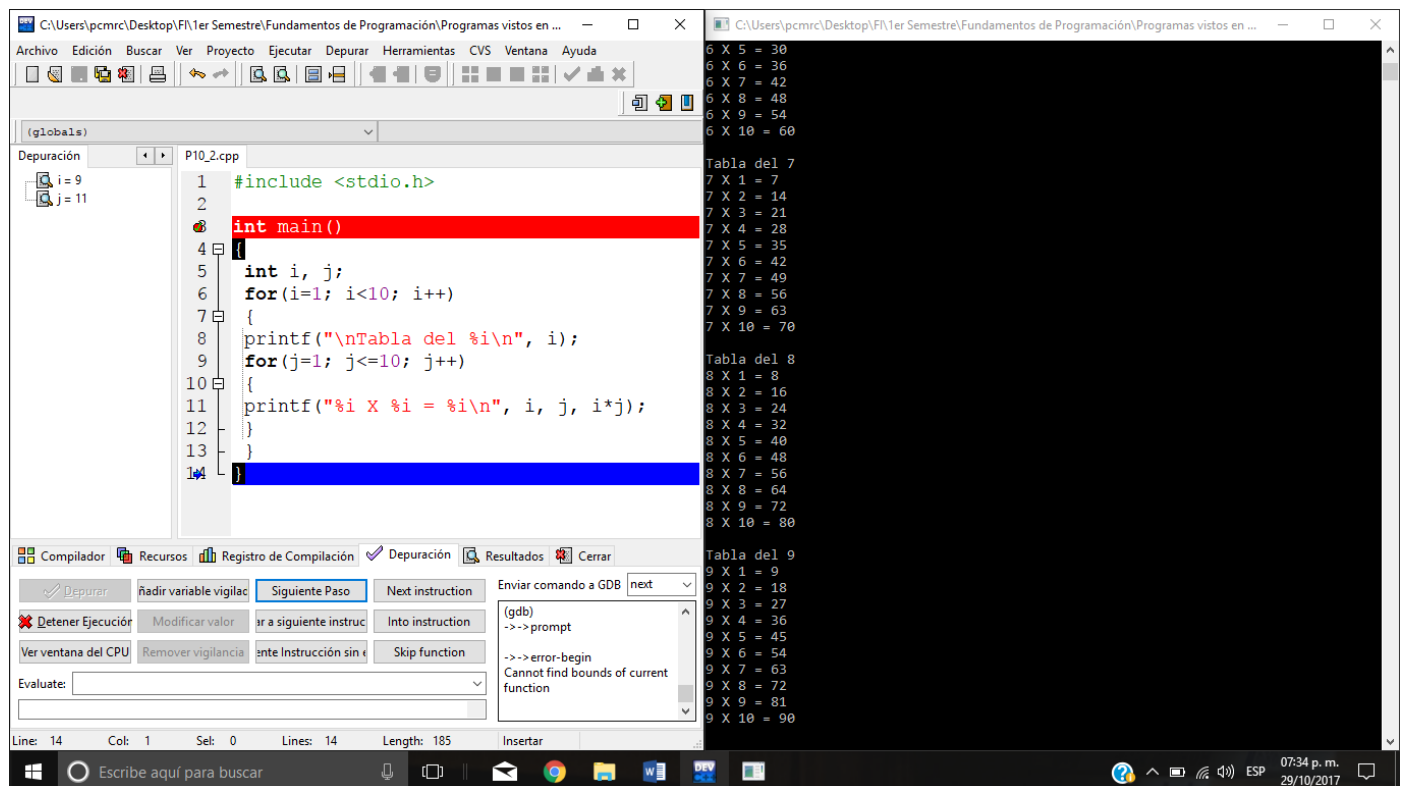
CÓDIGO FUENTE DEL PROGRAMA PROPUESTO 2:

```
#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);
        }
    }
}
```

DEPURACIÓN:



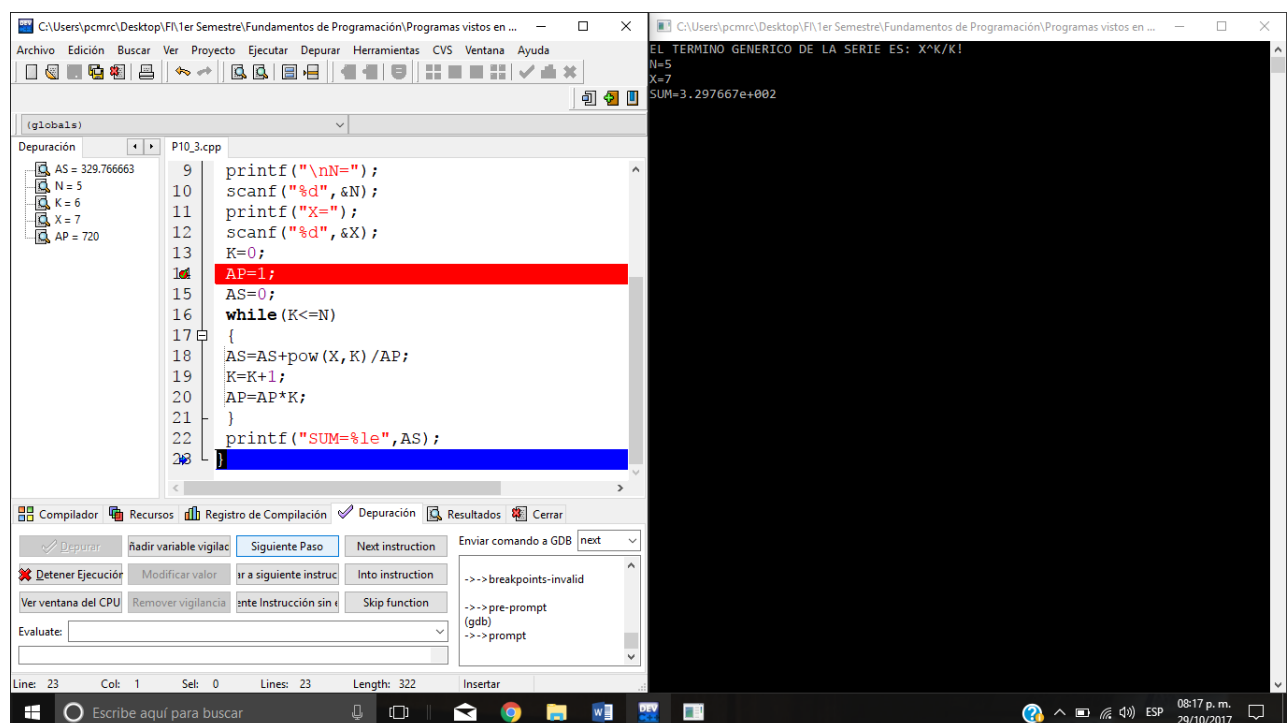
Como en el primer programa había un void previo al main, pero esta vez había un error humano, ya que el ciclo no estaba expresado de manera correcta, se indicaba que se iba a realizar esa operación mientras fuera igual a 10, cuando lo correcto era igual o menor a 10 (los diez productos de las tablas de multiplicar).

Me percaté del error viendo como en la depuración nunca entraba al ciclo, finalmente verifiqué el valor correcto de las variables.

CÓDIGO FUENTE DEL PROGRAMA PROPUESTO 3:

```
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);
}
```

Depuración:

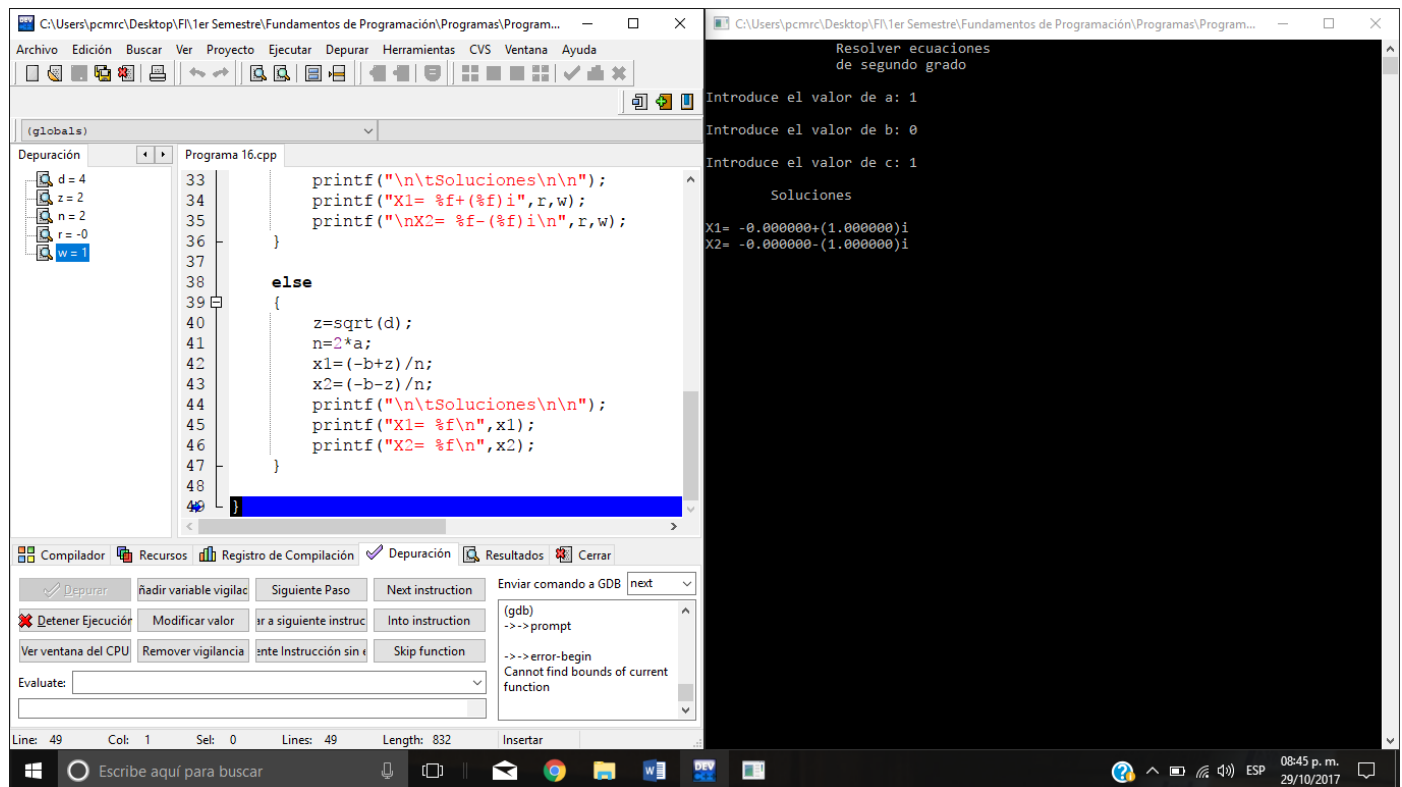


Al igual que los dos programas anteriores el error fue haber puesto el "void" antes del main, y en este caso había un error en la asignación de las variables, faltaba el "&" por lo cual no servía. Finalmente verificamos el correcto flujo del programa y la impresión del valor deseado.

CÓDIGO FUENTE DEL PROGRAMA DE LA FÓRMULA GENERAL DE SEGUNDO GRADO:

```
1  #include<stdio.h>
2  #include<math.h>
3
4  float a,b,c,d,z,n,r,w,x1,x2;
5
6  main()
7  {
8      printf("\t\tResolver ecuaciones \n\t\tde segundo grado\n");
9      printf("\nIntroduce el valor de a: ");
10     scanf("%f",&a);
11     printf("\nIntroduce el valor de b: ");
12     scanf("%f",&b);
13     printf("\nIntroduce el valor de c: ");
14     scanf("%f",&c);
15
16     while (a==0)
17     {
18         printf("\nNo valido para a=0\nIntroduce otro valor para a distitnto de 0");
19         printf("\na: ");
20         scanf("%f",&a);
21     }
22
23     d=(b*b)-(4*a*c);
24
25     if(d<0)
26     {
27         d=-1*d;
28         z=sqrt(d);
29         n=2*a;
30         r=-b/n;
31         w=z/n;
32         printf("\n\tSoluciones\n\n");
33         printf("X1= %f+(%f)i",r,w);
34         printf("\nX2= %f-(%f)i\n",r,w);
35     }
36
37     else
38     {
39         z=sqrt(d);
40         n=2*a;
41         x1=(-b+z)/n;
42         x2=(-b-z)/n;
43         printf("\n\tSoluciones\n\n");
44         printf("X1= %f\n",x1);
45         printf("X2= %f\n",x2);
46     }
47 }
48
49 }
```

DEPURACIÓN:



Al hacer la depuración en este programa, pude seguir el flujo y me percaté de la asignación de valores para cada variable, así como la entrada al ciclo correspondiente (gracias a la estructura que realicé por medio del algoritmo).

CONCLUSIONES:

Podría afirmar que la depuración debe de ser utilizada por cualquier tipo de programador, sea principiante o avanzado, siendo principiante pude ayudarte a reforzar las estructuras básicas y a encontrar muy fácilmente el error, y para un programador avanzado creo yo que sería útil ya que al tener un programa muy grande es un poco complicado encontrar los problemas, con la depuración sería más rápido y eficiente.

FUENTE:

- http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS_FP/fp_p10.pdf