



Depuración de programas.

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de Programación

Grupo: 3

Alumna: Aguilar Lara Alexa Patricia y Vázquez Espinosa Ximena Itzel

No. de Equipo de cómputo empleado: Máquina 28 Níger

No. de lista o Brigada: 01 y 52 No. de cuenta: 316315515 y

317058015

Fecha de entrega: 28/10/2019

Observaciones:

CALIFICACIÓN: _____

Depuración de programas

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 los posibles errores.

Se vio un método para linux que te permite identificar el problema en los programas, mejor conocido como depuración. Gracias a esto, se nos muestra una visión controlada y a detalle el proceso de nuestro programa para mejorarlo.

Ejemplo

`gcc -g`

`gdb`

`run`

`list (l)`

`quit (q)`

`CTRL+X+A`

`start`

`next (n)`

`CTRL+L`

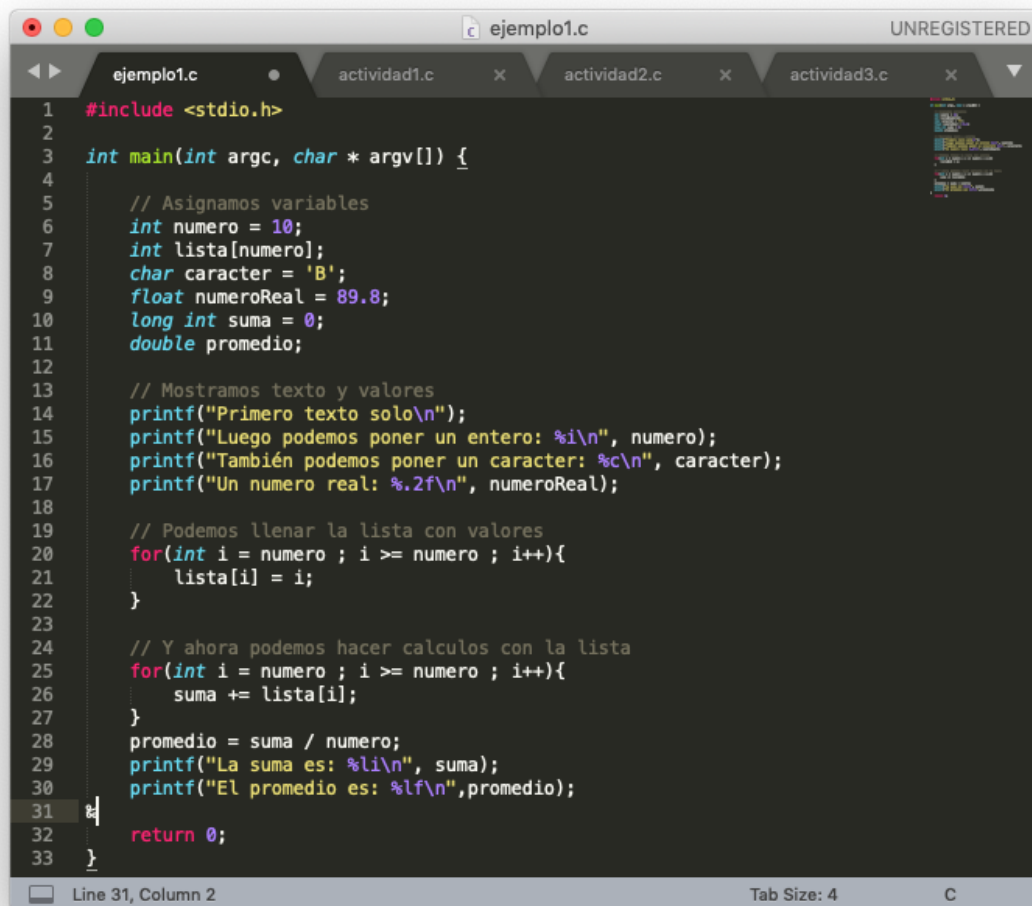
next (n)

break (b)

print (p)

display

Ejemplo1.c



The image shows a screenshot of a code editor window titled 'ejemplo1.c' with a 'UNREGISTERED' label in the top right corner. The editor has several tabs open: 'ejemplo1.c', 'actividad1.c', 'actividad2.c', and 'actividad3.c'. The 'ejemplo1.c' tab is active, displaying the following C code:

```
1  #include <stdio.h>
2
3  int main(int argc, char * argv[]) {
4
5      // Asignamos variables
6      int numero = 10;
7      int lista[numero];
8      char caracter = 'B';
9      float numeroReal = 89.8;
10     long int suma = 0;
11     double promedio;
12
13     // Mostramos texto y valores
14     printf("Primero texto solo\n");
15     printf("Luego podemos poner un entero: %i\n", numero);
16     printf("También podemos poner un caracter: %c\n", caracter);
17     printf("Un numero real: %.2f\n", numeroReal);
18
19     // Podemos llenar la lista con valores
20     for(int i = numero ; i >= numero ; i++){
21         lista[i] = i;
22     }
23
24     // Y ahora podemos hacer calculos con la lista
25     for(int i = numero ; i >= numero ; i++){
26         suma += lista[i];
27     }
28     promedio = suma / numero;
29     printf("La suma es: %li\n", suma);
30     printf("El promedio es: %lf\n", promedio);
31
32     return 0;
33 }
```

The status bar at the bottom indicates 'Line 31, Column 2', 'Tab Size: 4', and 'C'.

```
Documentos — -bash — 81x57
[Nepal26:Documents fp03alu01$ ls
actividad1.c  actividad2.c  actividad3.c  ejemplo1.c
[Nepal26:Documents fp03alu01$ gcc ejemplo1.c -o main
[Nepal26:Documents fp03alu01$ ./main
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80
Segmentation fault: 11
[Nepal26:Documents fp03alu01$ gcc -g ejemplo1.c -o ejemplo1
[Nepal26:Documents fp03alu01$ gdb ./ejemplo1
GNU gdb (GDB) 8.2
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-apple-darwin18.2.0".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./ejemplo1...Reading symbols from /Users/fp03alu01/Documents
/ejemplo1.dSYM/Contents/Resources/DWARF/ejemplo1...done.
done.
[(gdb) run
Starting program: /Users/fp03alu01/Documents/ejemplo1
Unable to find Mach task port for process-id 1467: (os/kern) failure (0x5).
(please check gdb is codesigned - see taskgated(8))
[(gdb) list
1      #include <stdio.h>
2
3      int main(int argc, char * argv[]) {
4
5          // Asignamos variables
6          int numero = 10;
7          int lista[numero];
8          char caracter = 'B';
9          float numeroReal = 89.8;
10         long int suma = 0;
[(gdb)
11         double promedio;
12
13         // Mostramos texto y valores
14         printf("Primero texto solo\n");
15         printf("Luego podemos poner un entero: %i\n", numero);
16         printf("También podemos poner un caracter: %c\n", caracter);
17         printf("Un numero real: %.2f\n", numeroReal);
18
19         // Podemos llenar la lista con valores
20         for(int i = numero ; i >= numero ; i++){
[(gdb) quit
Nepal26:Documents fp03alu01$
```

Fallo todo y nos conectamos a LINUX

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 80x24
Last login: Mon Oct 14 09:18:33 on ttys000
[Nepal26:~ fp03alu01$ servidor
[Nepal26:~ fp03alu01$ ssh fp03alu01@192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is SHA256:jTgFsbvP7IaIpwchV27DaUa9i2pvAVVZwZzbIneOF8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
[fp03alu01@192.168.2.200's password:
Samba
-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[fp03alu01@samba ~]$
```

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 81x39

-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[[fp03alu01@samba ~]$ ls
Escritorio actividad1.c actividad2.c actividad3.c ejemplo1.c
[[fp03alu01@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[[fp03alu01@samba ~]$ gdb ./ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/ejemplo1...done.
(gdb) run
Starting program: /users/fp03/fp03alu01/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x0000000040060c in main (argc=19, argv=0x1100000010) at ejemplo1.c:21
21         lista[i] = i;
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) list
16         printf("También podemos poner un caracter: %c\n", caracter);
17         printf("Un numero real: %.2f\n", numeroReal);
18
19         // Podemos llenar la lista con valores
20         for(int i = numero ; i >= numero ; i++){
21             lista[i] = i;
22         }
23
24         // Y ahora podemos hacer calculos con la lista
25         for(int i = numero ; i >= numero ; i++){
(gdb) █
```



```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 82x44

ejemplo1.c
1      #include <stdio.h>
2
3      int main(int argc, char * argv[]) {
4
5          // Asignamos variables
6      B-> int numero = 10;
7          int lista[numero];
8          char caracter = 'B';
9          float numeroReal = 89.8;
10         long int suma = 0;
11         double promedio;
12
13         // Mostramos texto y valores
14         printf("Primero texto solo\n");
15         printf("Luego podemos poner un entero: %i\n", numero);
16         printf("También podemos poner un caracter: %c\n", caracter);
17         printf("Un numero real: %.2f\n", numeroReal);
18
19         // Podemos llenar la lista con valores
20         for(int i = numero ; i >= numero ; i++){
21             lista[i] = i;
22         }
23
24         // Y ahora podemos hacer calculos con la lista
25         for(int i = numero ; i >= numero ; i++){
26             suma += lista[i];
27         }

```

child process 22069 In: main Line: 6 PC: 0x400542
(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu01/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7fffffff398) at ejemplo1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) next

next


```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 82x44
ejemplo1.c
1  #include <stdio.h>
2
3  int main(int argc, char * argv[]) {
4
5      // Asignamos variables
6      int numero = 10;
7      int lista[numero];
8      char caracter = 'B';
9      float numeroReal = 89.8;
10     long int suma = 0;
11     double promedio;
12
13     // Mostramos texto y valores
14     printf("Primero texto solo\n");
15     printf("Luego podemos poner un entero: %i\n", numero);
16     printf("También podemos poner un caracter: %c\n", caracter);
17     printf("Un numero real: %.2f\n", numeroReal);
18
19     // Podemos llenar la lista con valores
> 20     for(int i = numero ; i >= numero ; i++){
21         lista[i] = i;
22     }
23
24     // Y ahora podemos hacer calculos con la lista
25     for(int i = numero ; i >= numero ; i++){
26         suma += lista[i];
27     }

```

Un numero real: 89.80 un caracter: B

child process 22069 In: main Line: 20 PC: 0x40060f

(gdb) start

Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.

Starting program: /users/fp03/fp03alu01/ejemplo1

Temporary breakpoint 1, main (argc=1, argv=0x7fffffff398) at ejemplo1.c:6

Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64

(gdb) █

break

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 84x49
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/ejemplo1...done.
[[fp03alu01@samba ~]$ gdb ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/ejemplo1...done.
[[fp03alu01@samba ~]$ gdb ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/ejemplo1...done.
[[fp03alu01@samba ~]$ gdb ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/ejemplo1...done.
((gdb) break 20
Breakpoint 1 at 0x4005f7: file ejemplo1.c, line 20.
((gdb) run
Starting program: /users/fp03/fp03alu01/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80

Breakpoint 1, main (argc=1, argv=0x7fffffff398) at ejemplo1.c:20
20         for(int i = numero ; i >= numero ; i++){
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) █
```

print, display

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 95x65
ejemplo1.c
1  #include <stdio.h>
2
3  int main(int argc, char * argv[]) {
4
5      // Asignamos variables
6      int numero = 10;
7      int lista[numero];
8      char caracter = 'B';
9      float numeroReal = 89.8;
10     long int suma = 0;
11     double promedio;
12
13     // Mostramos texto y valores
14     printf("Primero texto solo\n");
15     printf("Luego podemos poner un entero: %i\n", numero);
16     printf("También podemos poner un caracter: %c\n", caracter);
17     printf("Un numero real: %.2f\n", numeroReal);
18
19     // Podemos llenar la lista con valores
B+> 20     for(int i = numero ; i >= numero ; i++){
21         lista[i] = i;
22     }
23
24     // Y ahora podemos hacer calculos con la lista
25     for(int i = numero ; i >= numero ; i++){
26         suma += lista[i];
27     }
28     promedio = suma / numero;
29     printf("La suma es: %li\n", suma);
30     printf("El promedio es: %lf\n", promedio);
31
32     return 0;
33 }^?
34
35
36
37
38
39
40
41

child process 22960 In: main Line: 20 PC: 0x4005f7
(gdb) p i
$1 = 0
(gdb) print lista
$2 = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
(gdb) display i
1: i = 0
(gdb) █
```

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 95x65

ejemplo1.c
1  #include <stdio.h>
2
3  int main(int argc, char * argv[]) {
4
5      // Asignamos variables
6      int numero = 10;
7      int lista[numero];
8      char caracter = 'B';
9      float numeroReal = 89.8;
10     long int suma = 0;
11     double promedio;
12
13     // Mostramos texto y valores
14     printf("Primero texto solo\n");
15     printf("Luego podemos poner un entero: %i\n", numero);
16     printf("También podemos poner un caracter: %c\n", caracter);
17     printf("Un numero real: %.2f\n", numeroReal);
18
19     // Podemos llenar la lista con valores
B+> 20     for(int i = numero ; i >= numero ; i++){
21         lista[i] = i;
22     }
23
24     // Y ahora podemos hacer calculos con la lista
25     for(int i = numero ; i >= numero ; i++){
26         suma += lista[i];
27     }
28     promedio = suma / numero;
29     printf("La suma es: %li\n", suma);
30     printf("El promedio es: %lf\n", promedio);
31
32     return 0;
33 }^?
34
35
36
37
38
39
40
41

child process 22960 In: main                               Line: 20   PC: 0x40060f
$1 = 0
(gdb) print lista
$2 = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
(gdb) display i
1: i = 0
(gdb) display lista
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
(gdb) n
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 10
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 10
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 11
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 11
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 12
2: lista = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
1: i = 12
(gdb) █
```

Actividad 1

- Utilizar GDB para encontrar la utilidad del programa y describir su funcionalidad.

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 94x24
[[fp03alu01@samba ~]$ gcc -std=c99 -g actividad1.c -o actividad1
[[fp03alu01@samba ~]$ ./actividad1
Ingresa un número: 5

El resultado es: 9
[[fp03alu01@samba ~]$
```

```
fp03alu01 — fp03alu01@samba:~ — ssh fp03alu01@192.168.2.200 — 92x31
[[fp03alu01@samba ~]$ gcc -std=c99 -g actividad1.c -o actividad1
[[fp03alu01@samba ~]$ gdb ./actividad1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu01/actividad1...done.
(gdb) run
Starting program: /users/fp03/fp03alu01/actividad1
Ingresa un número: 6

El resultado es: 9
[Inferior 1 (process 23427) exited with code 024]
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) list
1      #include <stdio.h>
2
3      void main()
4      {
5          int N, CONT, AS;
6          AS=0;
7          CONT=1;
8          printf("Ingresa un número: ");
9          scanf("%i",&N);
10         while(CONT<=N)
(gdb) █
```

Actividad 2

- Utilizar GDB para corregir el programa.

- NOTA: para compilar el código de la actividad, ejecutar:
\$ gcc -w actividad2.c -o activiad2 -lm

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

```
#include <math.h>
```

```
void main()
```

```
{
```

```
    int K, AP, N;
```

```
    double X, AS;
```

```
    printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
```

```
    printf("\nN=");
```

```
    scanf("%i",N);
```

```
    printf("X=");
```

```
    scanf("%lf",X);
```

```
    K=0;
```

```
    AP=1;
```

```
    AS=0;
```

```
    while(K<=N)
```

```
    {
```

```
        AS=AS+pow(X,K)/AP;
```

```
        K=K+1;
```

```
        AP=AP*K;
```

```
    }
```

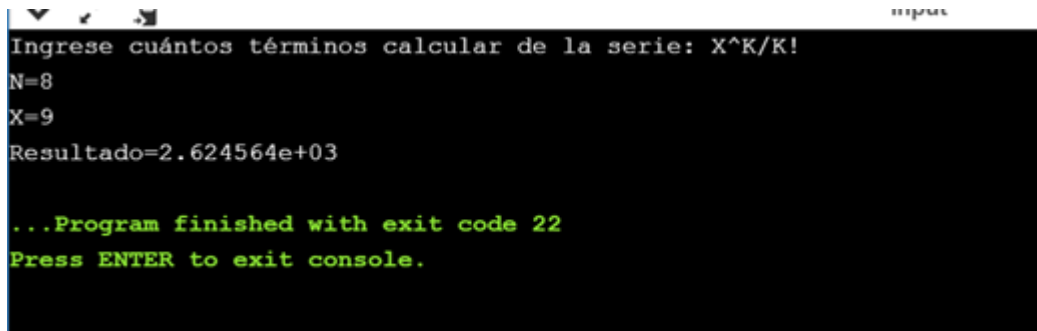
```
    printf("Resultado=%le",AS);
```

```
}
```

Faltaron los & para el scan

Cambio <= a <

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8      printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
9      printf("\nN=");
10     scanf("%i",&N);
11     printf("X=");
12     scanf("%lf",&X);
13     K=0;
14     AP=1;
15     AS=0;
16     while(K<N)
17     {
18         AS=AS+pow(X,K)/AP;
19         K=K+1;
20         AP=AP*K;
21     }
22     printf("Resultado=%le",AS);
23 }
24
```



```
Ingresa cuantos términos calcular de la serie: X^K/K!
N=8
X=9
Resultado=2.624564e+03

...Program finished with exit code 22
Press ENTER to exit console.
```

Actividad 3

- Utilizar GDB para corregir el programa

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

```
int main()
{
    int numero;

    printf("Ingresa un número:\n");
    scanf("%i",&numero);

    long int resultado = 1;
    while(numero>=0){
        numero--;
        resultado *= numero;
    }

    printf("El factorial de %i es %li.\n", numero, resultado);

    return 0;
}
```

Necesita una nueva variable, pues no cambia el valor si dejamos sólo la anterior y siempre es cero

Igualemos ambos números

Cambio `>=` por `>`

Intercambiamos `resultado *= numero` y `numero--` de lugar

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int numero;
6      int num2;
7
8      printf("Ingrese un número:\n");
9      scanf("%i",&numero);
10     num2=numero;
11
12     long int resultado = 1;
13     while(num2>0){
14         resultado *= num2;
15         num2--;
16     }
17
18     printf("El factorial de %i es %li.\n", numero, resultado);
19
20     return 0;
21 }
22
```

input

input

Ingrese un número:
6
El factorial de 6 es 720.

La práctica contuvo varios tropiezos, no comprendimos bien durante clase ya que el tiempo perdido fue una clave importante. De igual manera no hubo tiempo para ver las dudas finales en la clase.