



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Esteban Pimentel Alarcón

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* #10

*Integrante(s):* Miranda Gutiérrez Celine, Torres Mendoza Alexa Erandy

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 30, 49

*Semestre:* 2020-1

*Fecha de entrega:* 28/octubre/2019

*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.

- **Introducción**

La depuración de un programa es la forma de saber si un programa contiene errores o no, así mismo también nos ayuda a corregir dichos errores.

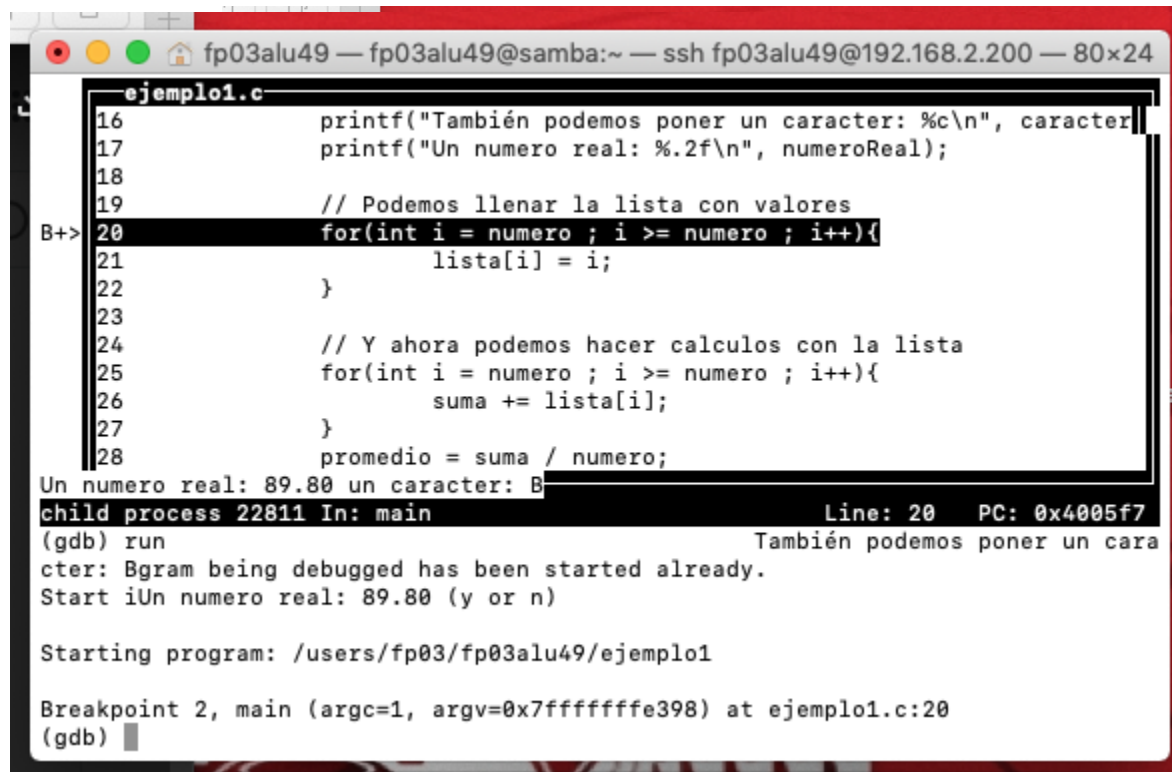
Se dice que un programa esta depurado cuando está libre de errores.

Cuando se depura un programa se hace un seguimiento del funcionamiento de dicho programa y se van estudiando los valores de las distintas variables, así como los resultados obtenidos en las operaciones.

Una vez depurado el programa se solucionan los posibles errores encontrados y se procede a depurar otra vez.

Estas acciones se repiten hasta que el programa no contiene ningún tipo de errores, tanto en tiempo de programación como en tiempo de ejecución.

Ejemplo:



```
fp03alu49 — fp03alu49@samba:~ — ssh fp03alu49@192.168.2.200 — 80x24
ejemplo1.c
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;
Un numero real: 89.80 un caracter: B
child process 22811 In: main                               Line: 20   PC: 0x4005f7
(gdb) run                                                  También podemos poner un cara
cter: Bgram being debugged has been started already.
Start iUn numero real: 89.80 (y or n)

Starting program: /users/fp03/fp03alu49/ejemplo1

Breakpoint 2, main (argc=1, argv=0x7fffffffe398) at ejemplo1.c:20
(gdb) █
```

```
fp03alu49 — fp03alu49@samba:~ — ssh fp03alu49@192.168.2.200 — 80x24
ejemplo1.c
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 22811 In: main                                Line: 21    PC: 0x4005ff
1: i = 11
1: i = 11
1: i = 12
1: i = 12
1: i = 13
1: i = 13
1: i = 14
(gdb)
```

## ACTIVIDAD 1

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

onlinegdb.com

Aplicaciones Sitios sugeridos Apple Bing Google Yahoo Importado de Inter... La corrosión atmosf... csxc Otros favoritos

OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE  
My Projects  
Learn Programming  
Programming Questions  
Sign Up  
Login

49.1K

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

```
main.c
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)
11     {
12         AS=(AS+CONT);
13         CONT=(CONT+2);
14     }
15     printf("\nEl resultado es: %i\n", AS);
16 }
17
```

Run Debug Stop Share Save Beautify Language C

#	Function	File:Line
0	main	main.c:12

Variable	Value
N	9
CONT	3
AS	1

Expression	Value
Enter expression to watch	

#	Description
---	-------------

input

Debug Console

(gdb) n  
10 while (CONT<=N)  
(gdb) n  
12 AS=(AS+CONT);  
(gdb)

start pause continue step over step into step out help

11:42 p.m. 28/10/2019

OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Sign Up

Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

```
main.c
1 #include <stdio.h>
2
3 void main()
4 {
5     int N;
6     AS=0;
7     printf("Ingresar un número: ");
8     scanf("%i", &N);
9     while (CONT<=N)
10    {
11        AS= (AS+CONT);
12        CONT=(CONT+2);
13    }
14    printf("Resultado= %i", AS);
15 }
```

input

Debug Console

Temporary breakpoint 1 at 0x4005ae: file main.c, line 6.  
Starting program: /home/a.out

Temporary breakpoint 1, main () at main.c:6  
(gdb) n  
6 AS=0;  
(gdb) n  
7 CONT=1;  
(gdb) n  
8 printf("Ingresar un número: ");  
(gdb) n  
9 scanf("%i", &N);  
(gdb) n  
Ingresar un número: 9  
10 while (CONT<=N)  
(gdb) n  
12 AS= (AS+CONT);  
(gdb) n  
13 CONT=(CONT+2);  
(gdb) n  
10 while (CONT<=N)  
(gdb) n  
12 AS= (AS+CONT);

Call Stack

#	Function	File:Line
0	main	main.c:12

Local Variables

Variable	Value
N	9
CONT	3
AS	1

Display Expressions

Expression	Value
Enter expression to watch	

Breakpoints and Watchpoints

#	Description
---	-------------

La funcionalidad del programa es recibir un número, número, después en otra variable llamada AS se va guardando el nuevo resultado de la suma de AS más el contador que igual es una variable que aumentará de dos en dos e inicia desde 1

## ACTIVIDAD 2

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

onlinegdb.com

OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Sign Up

Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

```
main.c
2 #include <math.h>
3
4 void main()
5 {
6     int K, AP, N;
7     double X, AS;
8     printf("Ingresar 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 }
```

input

Debug Console

0x00007ffff762682 in \_IO\_vfscanf\_internal (s=<optimized out>, format=<optimized out>, argptr=argptr@entry=0x7fffffffeac8, errp=errp@entry=0x0) at vfscanf.c:1857  
1857 vfscanf.c: No such file or directory.  
(gdb)

Call Stack

#	Function	File:Line
0	_IO_vfscanf_internal	vfscanf.c:1857
1	__isoc99_scanf	isoc99_scanf.c:37
2	main	main.c:10

Local Variables

Variable	Value
map	<optimized out>
argpos	<optimized out>
_buffer	[__routine = 0x7ffff7f68cc0 <_funlockfile>, __arg = 0x7ffff7ace640 <_IO_2_1_stdin_>, __canceltype = -143584832, __prev = 0x3abf8d2]
_avail	<optimized out>
arg	{{gp_offset = 16, fp_offset = 48, overflow_arg_area = 0x7ffffffeb0, reg_save_area = 0x7ffffffeae0}}
f	0x400876
fc	<optimized out>
done	<optimized out>
read_in	<optimized out>
c	<optimized out>
width	10
flags	64

OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Sign Up

Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

```

main.c
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);

```

input

Debug Console

(gdb) n  
printf("X=");  
(gdb) n  
scanf("%lf", &X);  
(gdb) n  
X=

Call Stack

#	Function	File:Line
0	main	main.c:12

Local Variables

Variable	Value
K	4195776
AP	0
N	4
X	6.9533558075923193e-310
AS	0

Display Expressions

Expression	Value
Enter expression to watch	

Breakpoints and Watchpoints

#	Description
---	-------------

Escribe aquí para buscar

12:07 a. m.  
29/10/2019

El error era que a los scanf les faltaba "&" para que el programa estuviera bien ejecutado, y quitar el igual a el while

onlinegdb.com

Aplicaciones Sitios sugeridos Apple Bing Google Yahoo Importado de Inter... La corrosión atmosf... csxc Otros favoritos

OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE

My Projects

Learn Programming

Programming Questions

Sign Up

Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

```

main.c
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;

```

input

Debug Console

Reading symbols from a.out...done.  
/usr/share/gdb/gdbinit: No such file or directory.  
(gdb) start  
Temporary breakpoint 1 at 0x4006be: file main.c, line 8.  
Starting program: /home/a.out  
  
Temporary breakpoint 1, main () at main.c:8  
(gdb) n  
printf("Ingrese cuántos términos calcular de la serie: X^K/K!");  
(gdb) n  
printf("\nN=");  
(gdb) n  
Ingrese cuántos términos calcular de la serie: X^K/K!  
(gdb) n  
scanf("%i", &N);  
(gdb) n  
N=4  
(gdb) n  
printf("X=");  
(gdb) n  
scanf("%lf", &X);  
(gdb) n  
X=9  
(gdb) n  
K=0;  
(gdb)

Call Stack

#	Function	File:Line
0	main	main.c:13

Local Variables

Variable	Value
K	4195776
AP	0
N	4
X	9
AS	0

Display Expressions

Expression	Value
Enter expression to watch	

Breakpoints and Watchpoints

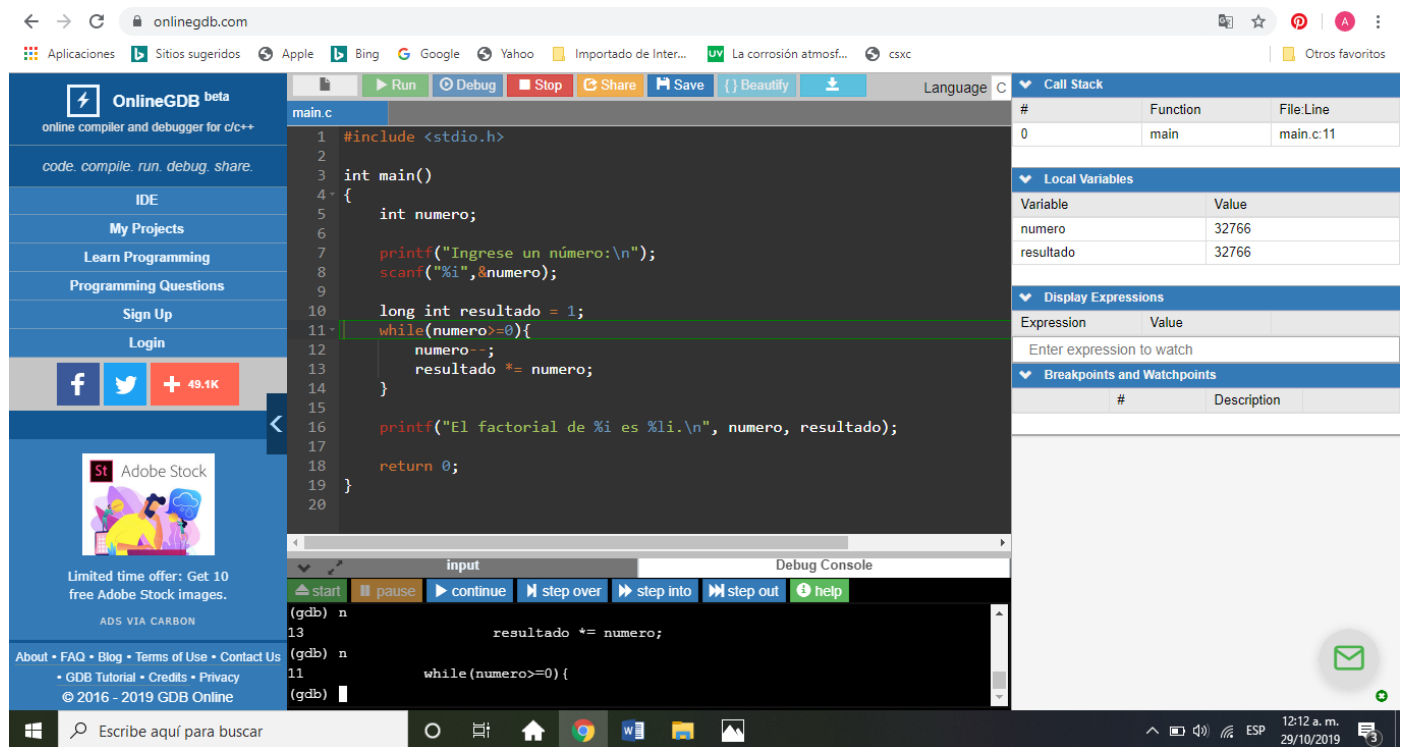
#	Description
---	-------------

Escribe aquí para buscar

12:09 a. m.  
29/10/2019

## ACTIVIDAD 3

Utilizar GDB para corregir el programa.



OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE  
My Projects  
Learn Programming  
Programming Questions  
Sign Up  
Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

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

input  
(gdb) start  
(gdb) n  
n  
resultado \*= numero;  
(gdb) n  
while(numero>=0){  
(gdb) |

Call Stack

#	Function	File:Line
0	main	main.c:11

Local Variables

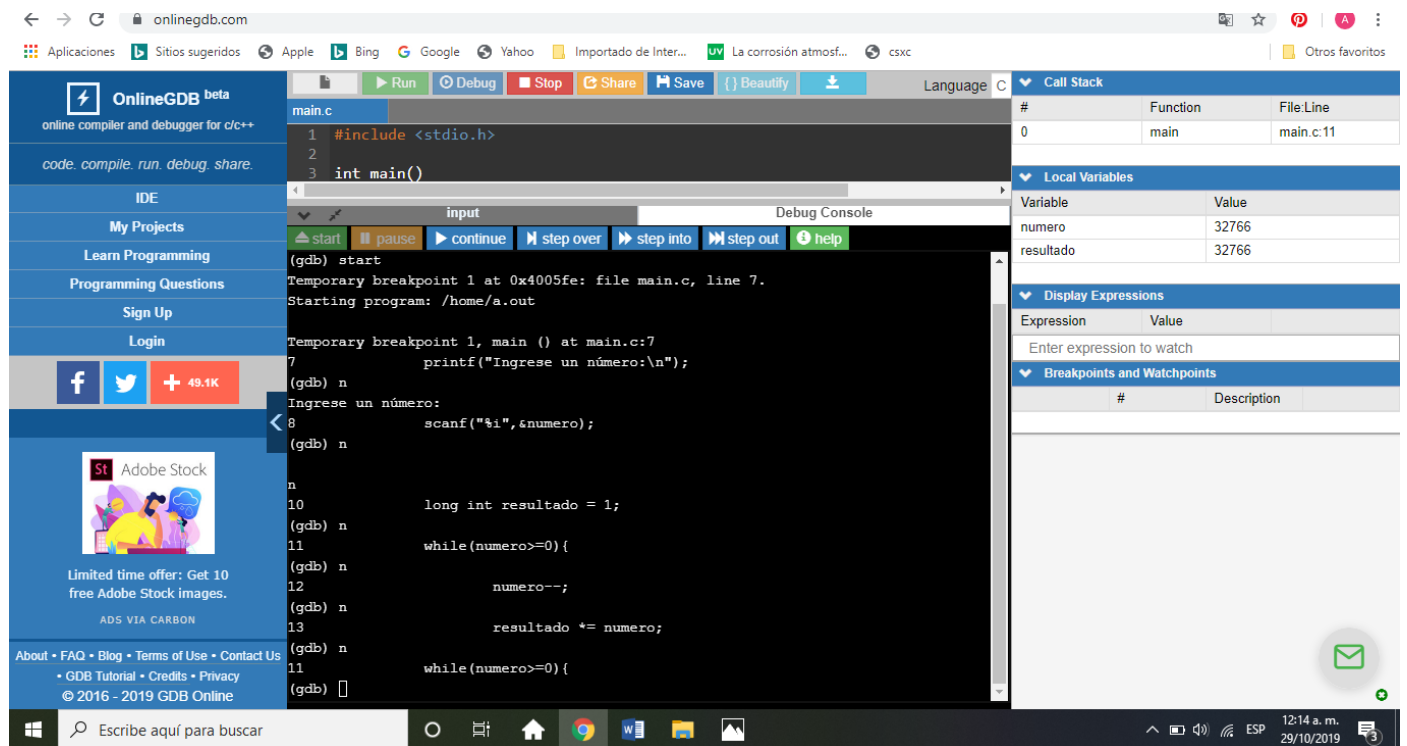
Variable	Value
numero	32766
resultado	32766

Display Expressions

Expression	Value
Enter expression to watch	

Breakpoints and Watchpoints

#	Description
---	-------------



OnlineGDB beta  
online compiler and debugger for c/c++  
code. compile. run. debug. share.

IDE  
My Projects  
Learn Programming  
Programming Questions  
Sign Up  
Login

Adobe Stock  
Limited time offer: Get 10 free Adobe Stock images.  
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us  
• GDB Tutorial • Credits • Privacy  
© 2016 - 2019 GDB Online

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

input  
(gdb) start  
Temporary breakpoint 1 at 0x4005fe: file main.c, line 7.  
Starting program: /home/a.out  
Temporary breakpoint 1, main () at main.c:7  
7 printf("Ingrese un número:\n");  
(gdb) n  
Ingrese un número:  
(gdb) n  
scanf("%i",&numero);  
(gdb) n  
n  
long int resultado = 1;  
(gdb) n  
while(numero>=0){  
(gdb) n  
numero--;  
(gdb) n  
resultado \*= numero;  
(gdb) n  
while(numero>=0){  
(gdb) |

Call Stack

#	Function	File:Line
0	main	main.c:11

Local Variables

Variable	Value
numero	32766
resultado	32766

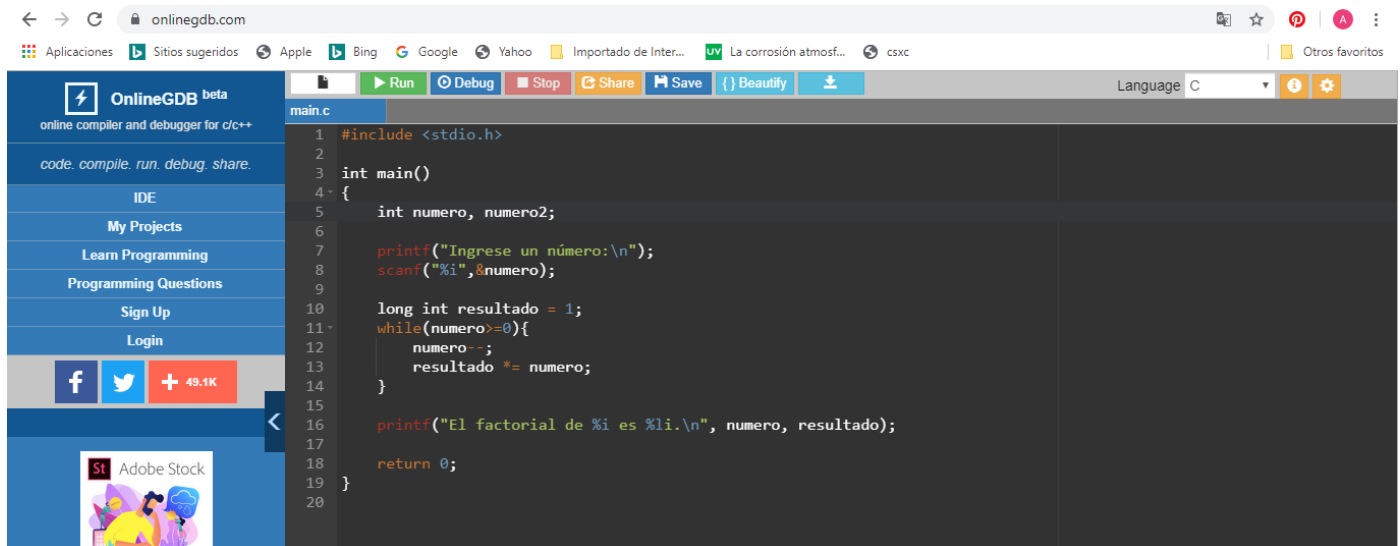
Display Expressions

Expression	Value
Enter expression to watch	

Breakpoints and Watchpoints

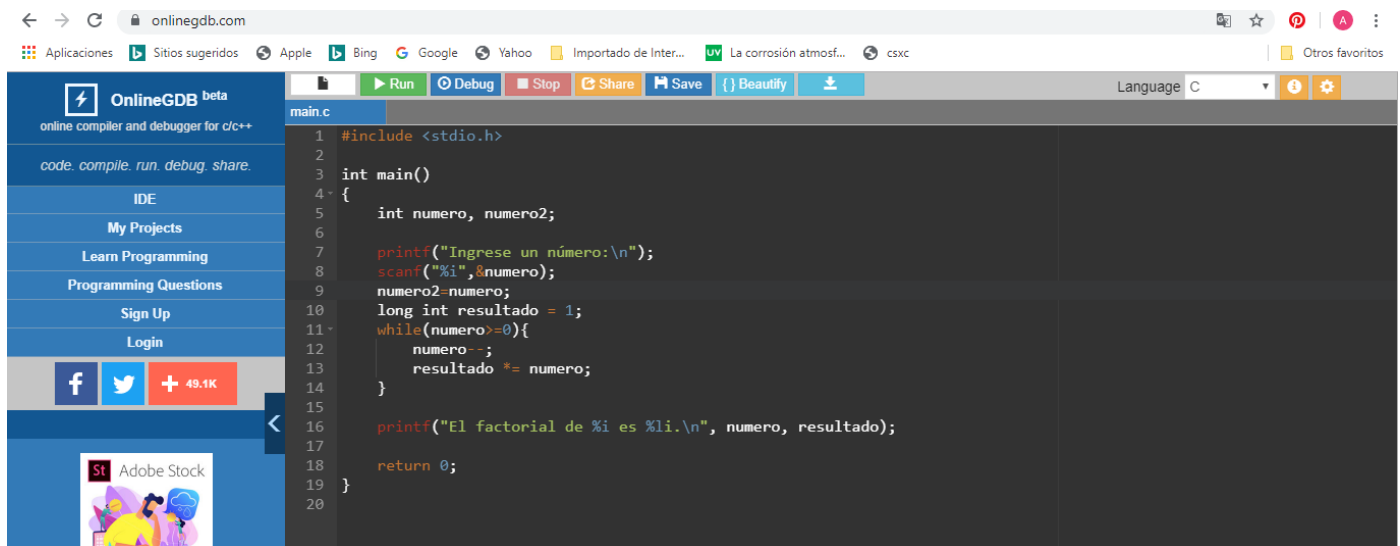
#	Description
---	-------------

Se le tiene que agregar una variable porque al final se imprime numero, pero el resultado final de numero es 0 así que se agrega numero2



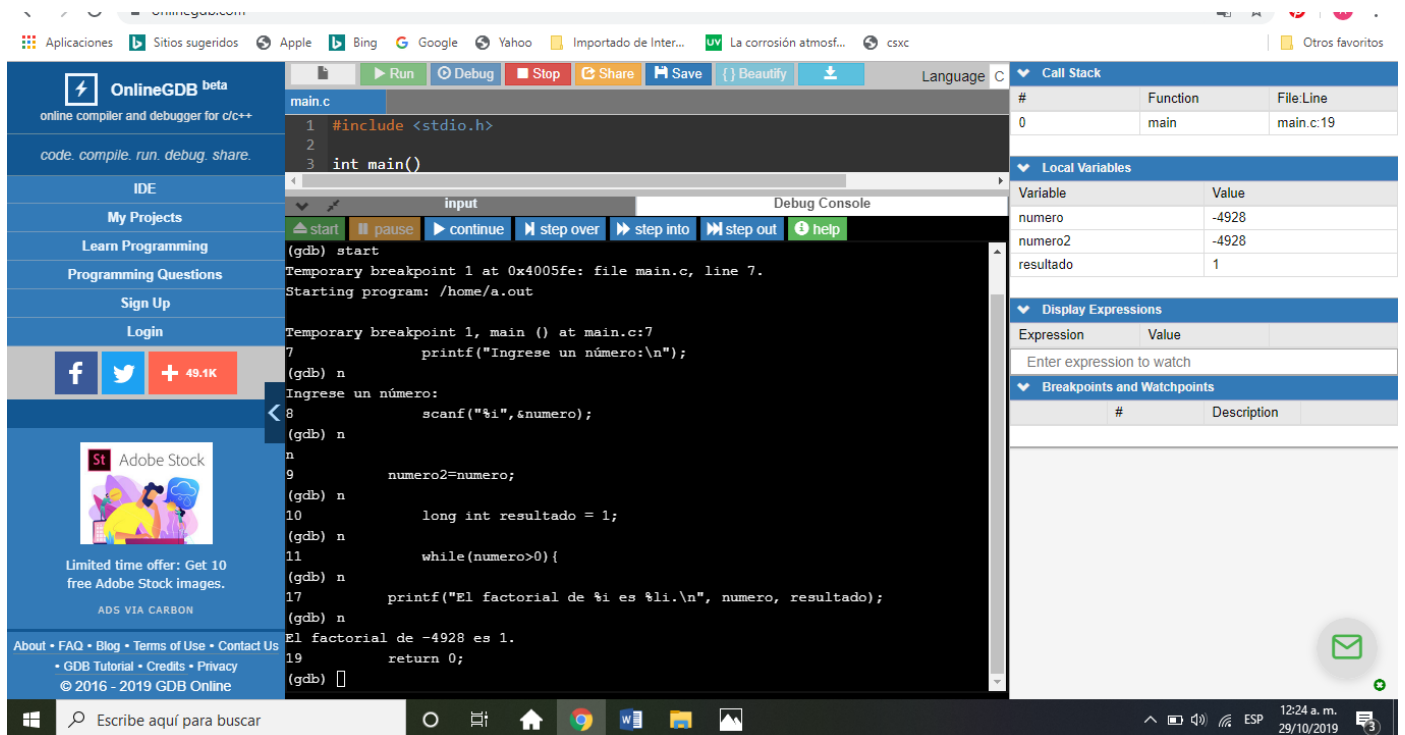
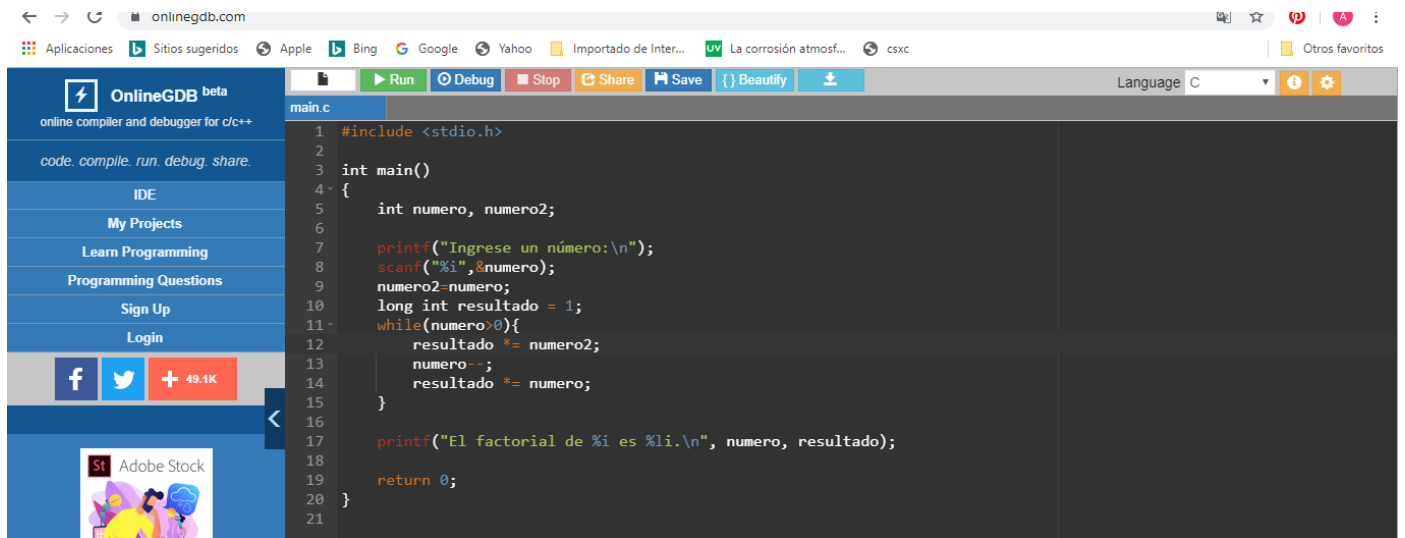
```
1 #include <stdio.h>
2
3 int main()
4 {
5     int numero, numero2;
6
7     printf("Ingrese un número:\n");
8     scanf("%i",&numero);
9
10    long int resultado = 1;
11    while(numero>=0){
12        numero--;
13        resultado *= numero;
14    }
15
16    printf("El factorial de %i es %li.\n", numero, resultado);
17
18    return 0;
19 }
20
```

Numero 2 es igual a numero para mantener su función



```
1 #include <stdio.h>
2
3 int main()
4 {
5     int numero, numero2;
6
7     printf("Ingrese un número:\n");
8     scanf("%i",&numero);
9     numero2=numero;
10    long int resultado = 1;
11    while(numero>0){
12        numero--;
13        resultado *= numero;
14    }
15
16    printf("El factorial de %i es %li.\n", numero, resultado);
17
18    return 0;
19 }
20
```

Se cambia el numero2>=0 a numero2>0 esto para evitar la multiplicación por 0  
Y se cambia la posición de numero2, para que se restara hasta el final y no al principio del proceso



## CONCLUSIONES

La depuración de programas es el proceso de identificar y corregir errores de programación. En inglés se conoce como debugging, porque se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.

Muchas veces se requiere incluir en el código fuente instrucciones auxiliares que permitan el seguimiento de la ejecución del programa, presentando los valores de variables y direcciones de memoria y ralentizando la salida de datos ("modo de depuración"). Dentro de un proceso formal de aseguramiento de la calidad, puede ser asimilado al concepto de "prueba unitaria".