	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor : Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 10

Integrante(s): Meraz Dionicio Israel

No. de Equipo de cómputo empleado: 2

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 28 de Octubre de 2019

Observaciones:

CALIFICACIÓN: _____

OBJETVO:
Aprender las técnicas básicas de depuración 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.

INTRODUCCION:
GDB es un depurador desarrollado por la FSF (Free Software Foundation) que se usa principalmente en Linux y bajo la licencia publica gratis GPL (GNU Public Licence), le permite ver lo que sucede `dentro 'de otro programa mientras se ejecuta, o lo que otro programa estaba haciendo en el momento en que se bloqueó.

GDB puede hacer cuatro tipos principales de cosas (además de otras cosas en apoyo de estas) para ayudarlo a detectar errores en el acto:

- Inicie su programa, especificando cualquier cosa que pueda afectar su comportamiento.
- Haga que su programa se detenga en condiciones específicas.
- Examine lo que sucedió cuando su programa se detuvo.
- Cambie las cosas en su programa, para que pueda experimentar corrigiendo los efectos de un error y continuar aprendiendo sobre otro.

EJEMPLO 1.

```c
#include <stdio.h>

int main(int argc, char * argv[]) {

	// Asignamos variables
	int numero = 10;
	int lista[numero];
	char caracter = 'B';
	float numeroReal = 89.8;
	long int suma = 0;
	double promedio;

	// Mostramos texto y valores
	printf("Primero texto solo\n");
	printf("Luego podemos poner un entero: %i\n", numero);
	printf("También podemos poner un caracter: %c\n", caracter);
	printf("Un numero real: %.2f\n", numeroReal);

	// Podemos llenar la lista con valores
	for(int i = numero ; i >= numero ; i++){
		lista[i] = i;
	}

	// Y ahora podemos hacer calculos con la lista
	for(int i = numero ; i >= numero ; i++){
		suma += lista[i];
	}
	promedio = suma / numero;
	printf("La suma es: %li\n", suma);
	printf("El promedio es: %lf\n",promedio);

	return 0;
}
```

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./ejemplo1...Reading symbols from /Users/fp03alu29/Download
s/ejemplo1.dSYM/Contents/Resources/DWARF/ejemplo1...done.
done.
(gdb) run
Starting program: /Users/fp03alu29/Downloads/ejemplo1
Unable to find Mach task port for process-id 1237: (os/kern) failure (0x5).
 (please check gdb is codesigned - see taskgated(8))
(gdb)
```

Al principio las computadoras Mac y gdb no son compatibles, ytuvimos que cambiarnos a una terminal de linux para que pudiera funcionar. Primero nos conectamos a un servidor de linux, despues ssh, la contraseña y hasta que apareciera samba

```
Last login: Mon Oct 14 09:16:07 on ttys000
Kuwait02:~ fp03alu29$ servidor
Kuwait02:~ fp03alu29$ ssh fp03alu29@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:jTgFsbnvP7IaIpwchV27DaUa9i2pvAVVZwZzbIneOF8.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
fp03alu29@192.168.2.200's password:
```

```
-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[fp03alu29@samba ~]$ ls
Escritorio  ejemplo1.c
[fp03alu29@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[fp03alu29@samba ~]$ $
```

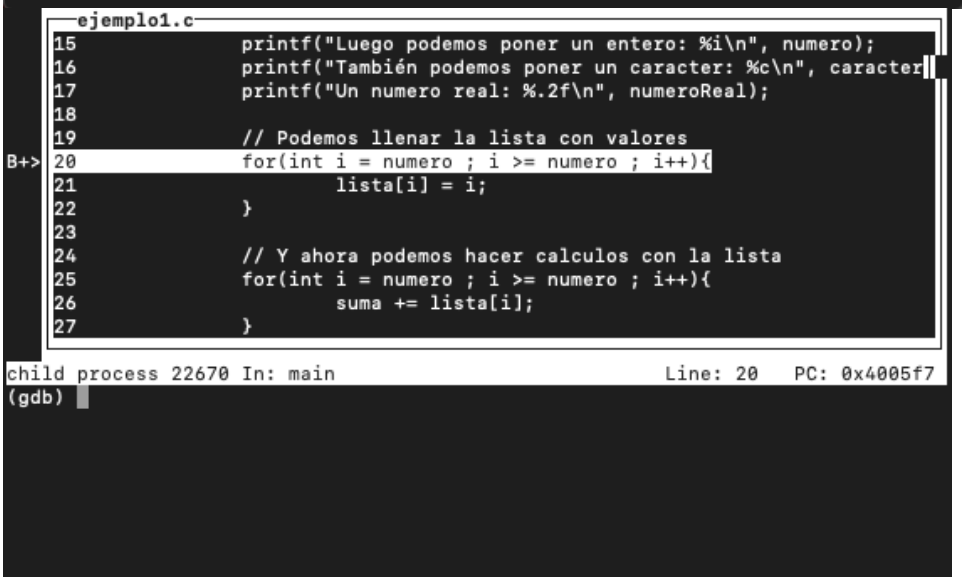
```
[[fp03alu29@samba ~]$ $gdb ./ejemplo1
 Primero texto solo
 Luego podemos poner un entero: 10
 También podemos poner un caracter: B
 Un numero real: 89.80
 Violación de segmento (`core' generado)
[[fp03alu29@samba ~]$ gdb ./ejemplo1
 GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
 Copyright (C) 2012 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/fp03alu29/ejemplo1...done.
 (gdb)
```

Ya entrando en gdb, se compila gcc –std=c99 –g … y luego se corre, se genero un core, para invocar a gdb es gdb y el ejecutable hasta que aparezaca (gdb) sólo va a recibir órdenes de gdb.
Se corre con run bajo el programa gdb.
La lista (list o l) te mostrara algunas partes del código. Y te muestra en donde esta el error
El break es detener teporalmente la ejecución para verificar la variables de entrada o para averiguar donde falla el programa.

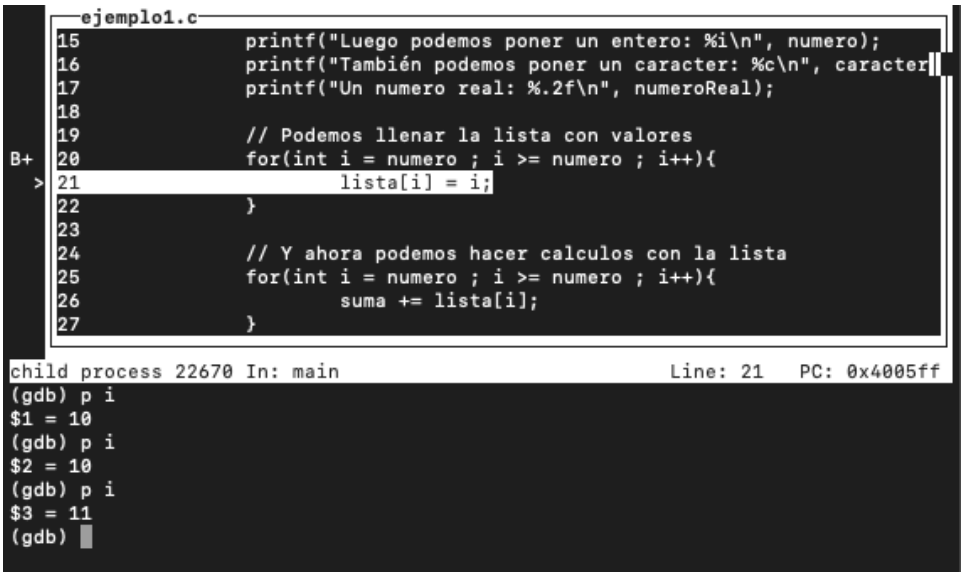
```
(gdb) run
Starting program: /users/fp03/fp03alu29/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x000000000040060c 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) l
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)
```

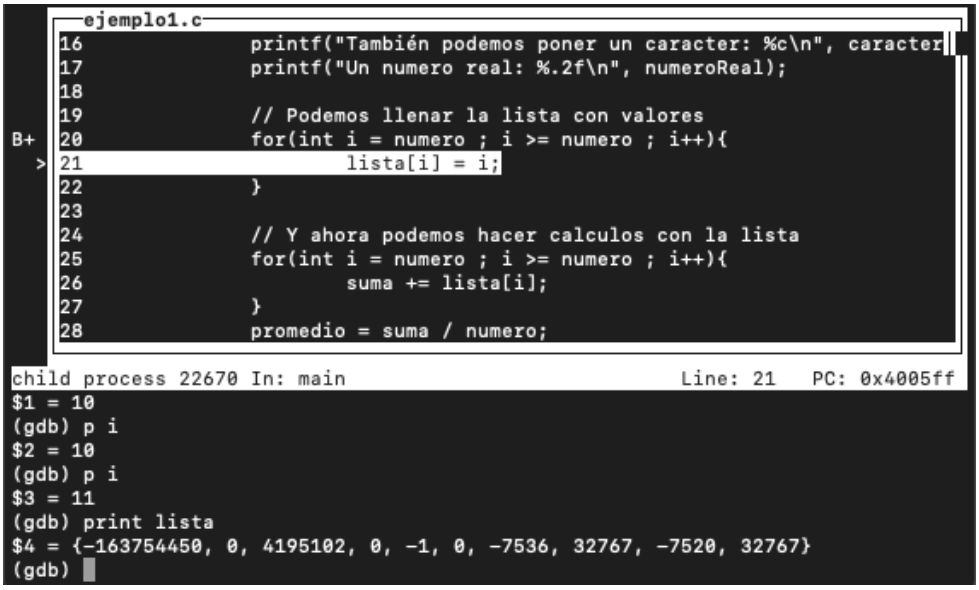
```
(gdb) q
A debugging session is active.

        Inferior 1 [process 21798] will be killed.

Quit anyway? (y or n) y
[fp03alu29@samba ~]$
```

Para salir de gdb se escribe la orden quit o q.
Vueves a entrar al programa de gdb y con las teclas presiona CTRL+x+A y entra al modo de la Interfaz de usuario de texto (TUI)

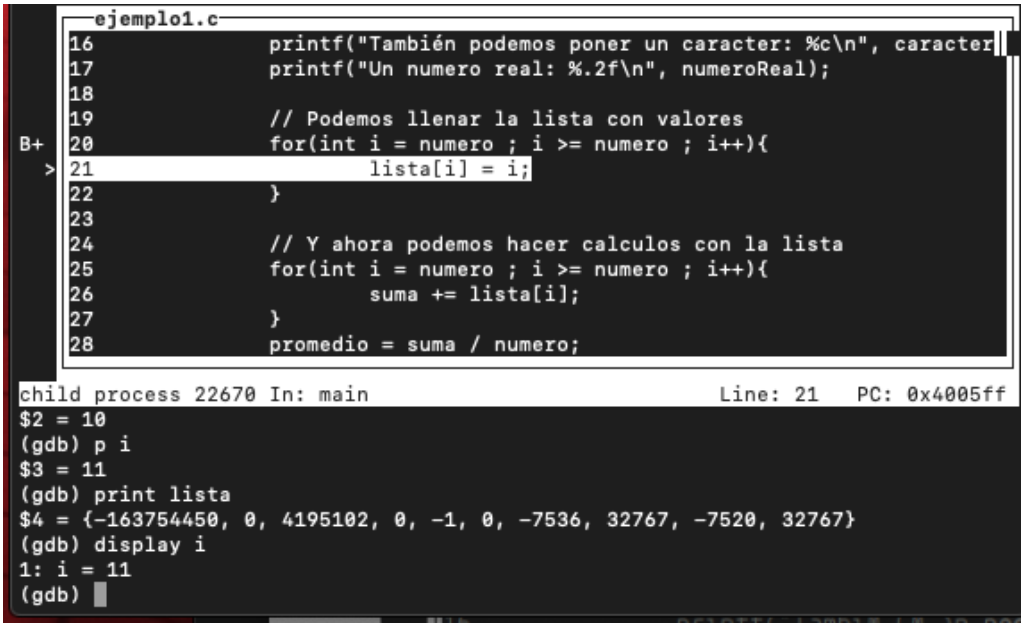


Para comenzar el gdb TUI se escribe start y te muestra el programa
Con next o n funciona para que se mueva a la siguiente linea no puede retroceder .
Con CTRL-L sirve en ocasiones cuando la pantalla se “desordena”y forzara a volver a dibujar la pantalla terminal

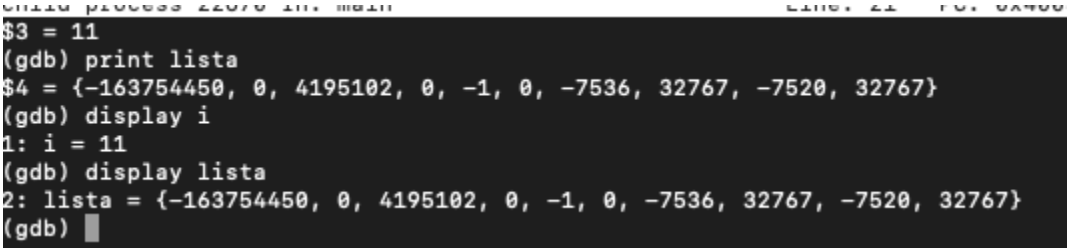


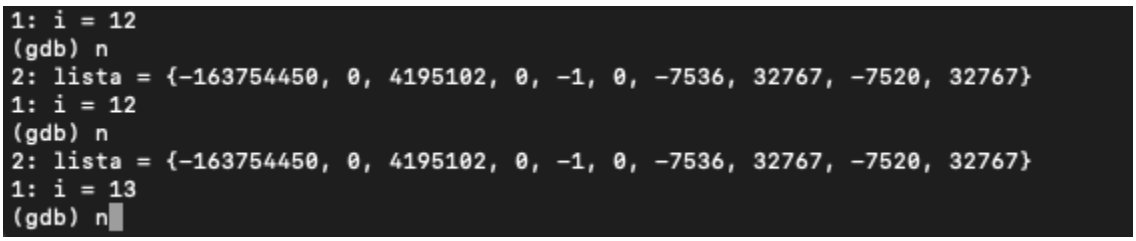


El print o p funciona para imprimir el valor de la variable o la lista en la función actual.
Como se muestra se imprimio i y mostro sólo un valor, si imprimes el nombre la lista que en este caso es “lista” imprime valores



El display muestra el valor de la variiable o el de la lista cada vez que el programa se detiene o sigues explorando en las lineas del programa.





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

```
Archivo  Edición  Formato  Ver  Ayuda
#include <stdio.h>

void main()
{
        int N, CONT, AS;
        AS=0;
        CONT=1;
        printf("Ingresa un número: ");
        scanf("%i",&N);
        while(CONT<=N)
        {
                AS=(AS+CONT);
                CONT=(CONT+2);
        }
        printf("\nEl resultado es: %i\n", AS);
}
```

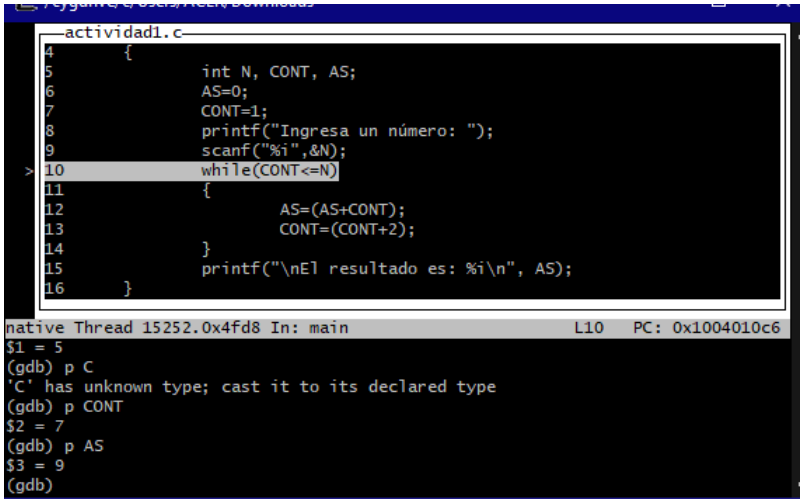
```
/cygdrive/c/Users/ACER/Downloads
También podemos poner un caracter: B
Un numero real: 89.80

Thread 1 "ejemplo1" hit Breakpoint 1, main (argc=1, argv=0xffffcc30)
    at ejemplo1.c:20
20              for(int i = numero ; i >= numero ; i++){
(gdb)
ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad1.c -o actividad1

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad1
Ingresa un número: 6

El resultado es: 9

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad1
Ingresa un número: 2

El resultado es: 1

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$
```

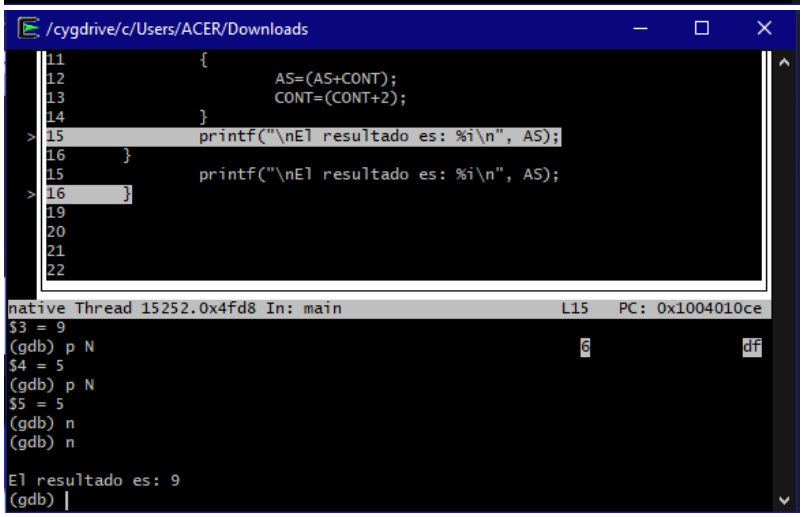
```
/cygdrive/c/Users/ACER/Downloads
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-pc-cygwin".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
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 ./act1...done.
(gdb) run
Starting program: /cygdrive/c/Users/ACER/Downloads/act1
[New Thread 23024.0x3228]
[New Thread 23024.0x572c]
[New Thread 23024.0x53bc]
[New Thread 23024.0x3bdc]
Ingresa un número: 3

El resultado es: 4
[Thread 23024.0x3bdc exited with code 20]
[Thread 23024.0x53bc exited with code 20]
[Inferior 1 (process 23024) exited with code 024]
(gdb)
```

Primero compilamos el programa y lo corremos, el resultado es que no tiene fallas el programa pero queremos conocer su funcionalidad, que hace.

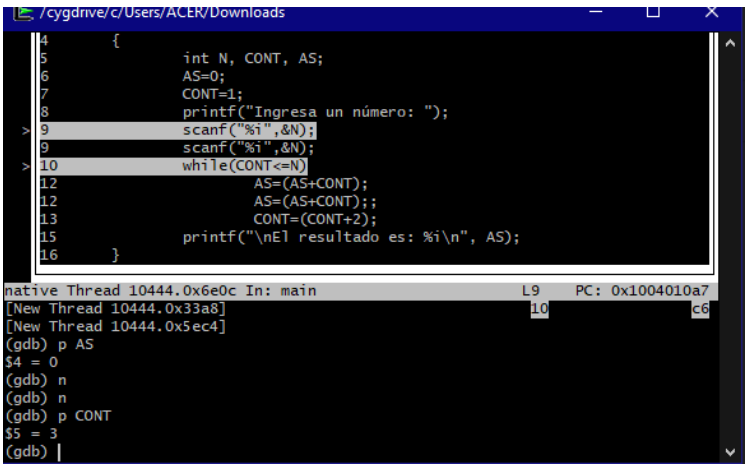
Entramos en el depurador de GDB, y aplicamos lo mismo que el ejemplo 1, escribi run, luego lista, y vi que estaba funcionado bien, entonces entre con CTRL-X-A para el interfaz y tener una mejor visualización.

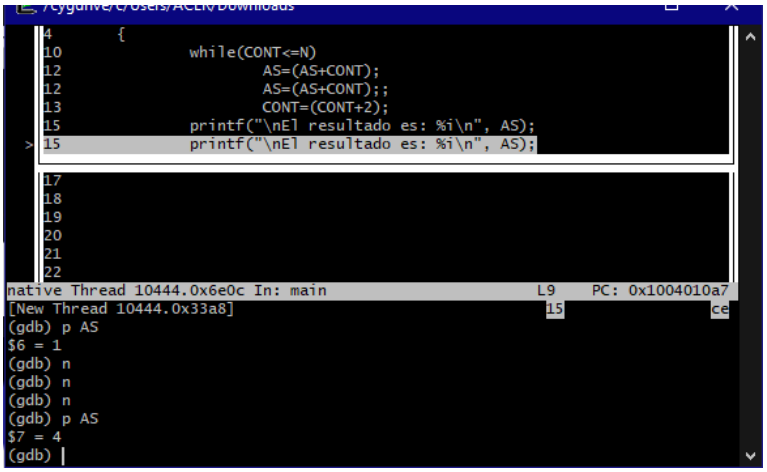
```
/cygdrive/c/Users/ACER/Downloads
7               CONT=1;
8               printf("Ingresa un número: ");
9               scanf("%i",&N);
10              while(CONT<=N)
(gdb) q

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gdb act1
GNU gdb (GDB) (Cygwin 8.1.1-1) 8.1.1
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-pc-cygwin".
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 act1...done.
(gdb)
```

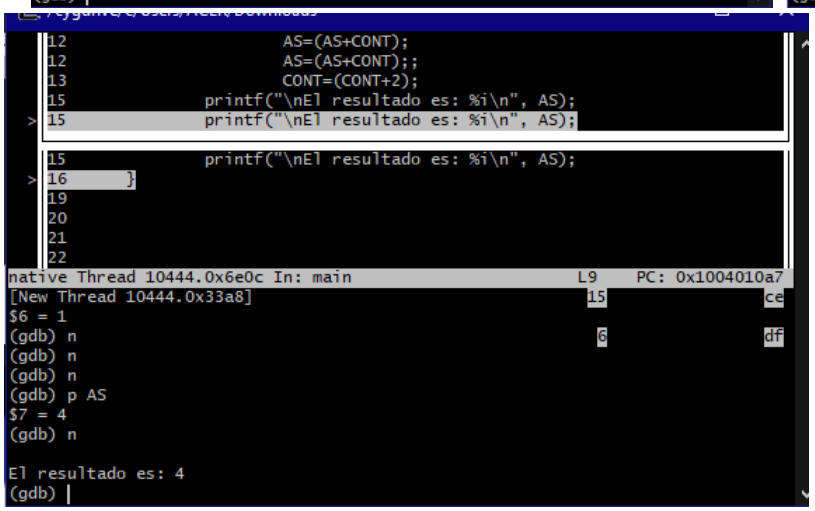




Entrando al interfaz e iniciando con start, veo el código del programa, para navegar escribi n.
Me doy cuenta cuando me piden un número y hay dos variables más que son el CONT y el AS con inicializaciones de 1 y 0 respectivamente.
Lo que hace el programa es: mientras el cont sea menor o igual al número que ingrese hace que guarde valores en as con as+contador, y luego cont tambien guarda valores sumandole mas dos y asi se va a detenr cuando el cont sea mayor al número ingresado.
Como se muestra la variable cont se detuvo en 7 y as tenia el valor de 9







Con otro ejemplo el número ingresado ahora es 3, As es igual a 1 y cont es igual a 3, cont es menor al número, si se repetía; as ahora es igual a 4 y cont igual a 5, por lo tanto cont ya era mayor al numero e imprimia el valor de as que era 4

- Utilizar GDB para coregir el programa

```c
#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);
}
```

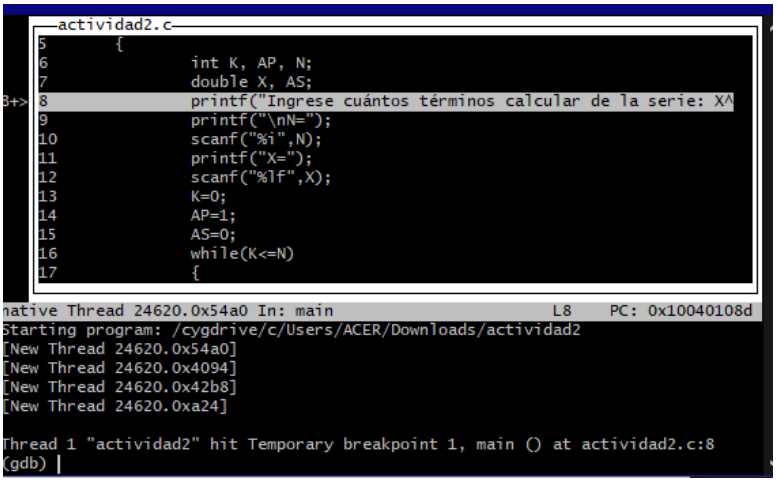
```
/cygdrive/c/Users/ACER/Downloads
$ gcc actividad
actividad.exe              actividad2.c
actividad.exe.stackdump    actividad2.exe
actividad1.c               actividad-formularios.docx
actividad1.exe

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad2.c -o acti

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./acti
Ingrese cuántos términos calcular de la serie: X^K/K!
N=4

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc -w actividad2.c -o actividad2 -lm

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=5

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$
```

```
[New Thread 19532.0x6fc0]
[New Thread 19532.0x402c]
[New Thread 19532.0x7144]
Ingrese cuántos términos calcular de la serie: X^K/K!
N=[New Thread 19532.0x5c28]
[New Thread 19532.0x1d44]
5
[Thread 19532.0x6fc0 exited with code 3221225477]
[Thread 19532.0x1d44 exited with code 3221225477]
[Thread 19532.0x5c28 exited with code 3221225477]
[Thread 19532.0x7144 exited with code 3221225477]
[Inferior 1 (process 19532) exited with code 030000000005]
(gdb) run
Starting program: /cygdrive/c/Users/ACER/Downloads/actividad2
[New Thread 20176.0x7020]
[New Thread 20176.0x6b08]
[New Thread 20176.0x6d44]
[New Thread 20176.0x5784]
Ingrese cuántos términos calcular de la serie: X^K/K!
N=5
[Thread 20176.0x6b08 exited with code 3221225477]
[Thread 20176.0x5784 exited with code 3221225477]
[Inferior 1 (process 20176) exited with code 030000000005]
(gdb)
```

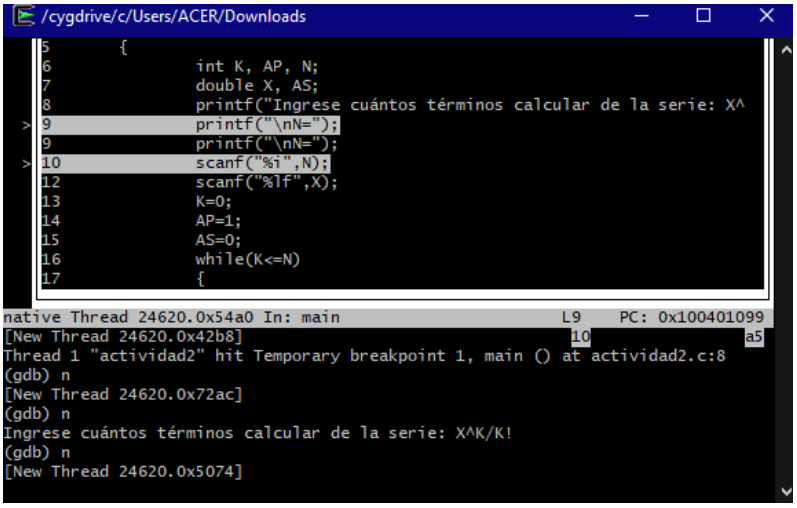
Para saber cuál es el error es compilarlo pero esta vez lo compile con ggc –w actividad2.c –o actividad2 –lm para que pudiera correr el programa, lo corri y el problema termina al escribir el primer valor.

```
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);
(gdb) l
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;
(gdb) l
21              }
22              printf("Resultado=%le",AS);
23      }
(gdb)
```

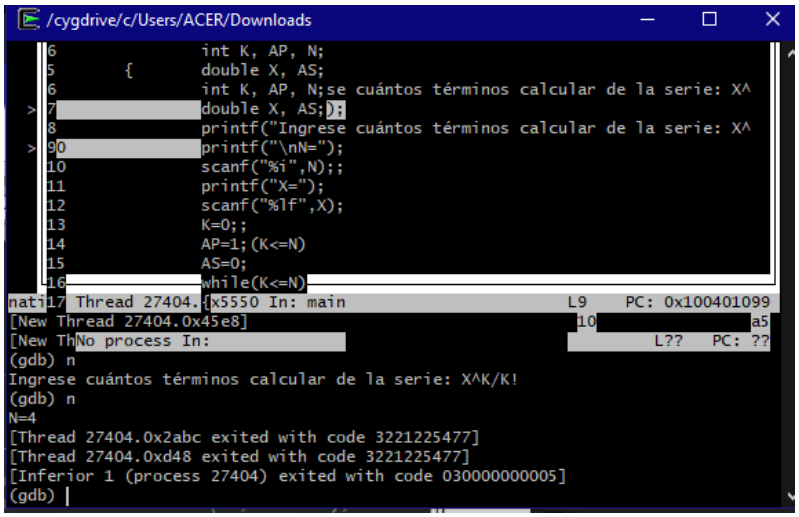
Ingrese a GDB, lo corri y me aparecio un problema después de ingresa el valor, escribi l para ver el código completo.

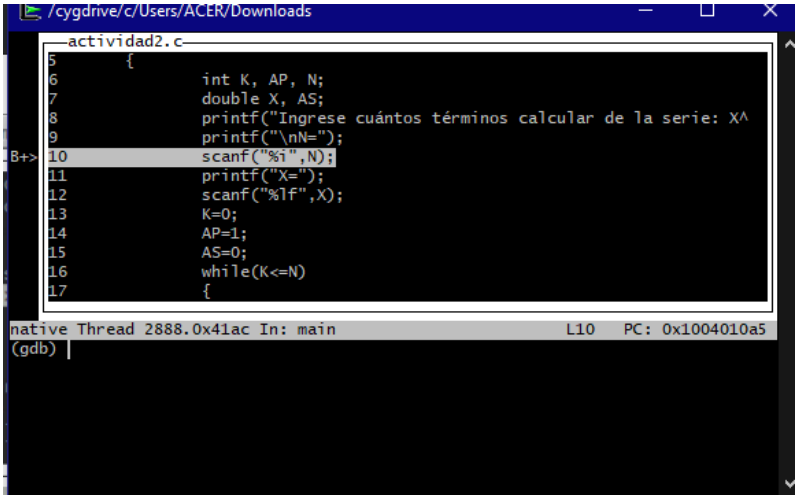


Para ver mejor el problema ingrese al interfaz Igualente lo empece y a partir de ese momento me di cuenta que el error es que estaba imcompleto el código y justamente el error son los scanf, les faltaba & para asignarles el valor.



Ya que siempre terminaba la ejecución en el primer scanf





Cuando lo correji sólo agregue lo que le faltaba y al final, lo compile, lo ejecute y corrio el programa con normalidad

```
Archivo  Edición  Formato  Ver  Ayuda
#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);
}
```

```
/cygdrive/c/Users/ACER/Downloads
(gdb) break 10
Breakpoint 1 at 0x1004010a5: file actividad2.c, line 10.
(gdb) run
Starting program: /cygdrive/c/Users/ACER/Downloads/actividad2
[New Thread 2888.0x41ac]
[New Thread 2888.0x1ce4]
[New Thread 2888.0x6f48]
[New Thread 2888.0x710c]
Ingrese cuántos términos calcular de la serie: X^K/K!

Thread 1 "actividad2" hit Breakpoint 1, main () at actividad2.c:10
10              scanf("%i",N);
(gdb)
ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc -w actividad2.c -o actividad2 -lm

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad2
Ingrese cuántos términos calcular de la serie: X^K/K!
N=4
X=3
Resultado=1.637500e+01
ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$
```

- Utilizar GDB para coregir el programa

```
#include <stdio.h>

int main()
{
        int numero;

        printf("Ingrese 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;
}
```

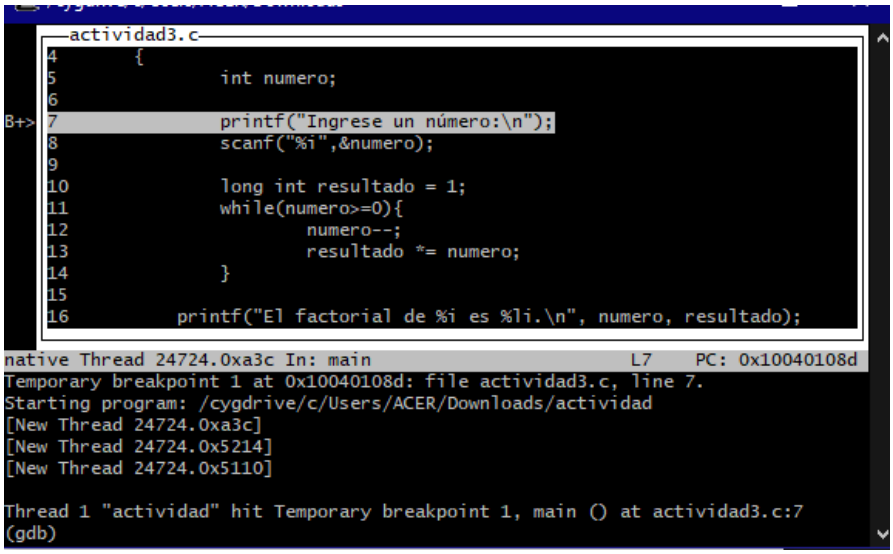
```
7               double X, AS;
8               printf("Ingrese cuántos términos calcular de la serie: X^K/K!");
9               printf("\nN=");
10              scanf("%i",N);
(gdb) q

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad
actividad.exe               actividad2.c
actividad.exe.stackdump     actividad2.exe
actividad1.c                actividad3.c
actividad1.exe              actividad-formularios.docx

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad3.c -o actividad

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad
Ingrese un número:
4
El factorial de -1 es 0.

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$
```

```
/cygdrive/c/Users/ACER/Downloads
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-pc-cygwin".
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 ./actividad...done.
(gdb) run
Starting program: /cygdrive/c/Users/ACER/Downloads/actividad
[New Thread 17976.0x524c]
[New Thread 17976.0x1218]
[New Thread 17976.0x688c]
[New Thread 17976.0x3364]
Ingrese un número:
4
El factorial de -1 es 0.
[Thread 17976.0x3364 exited with code 0]
[Thread 17976.0x688c exited with code 0]
[Inferior 1 (process 17976) exited normally]
(gdb)
```

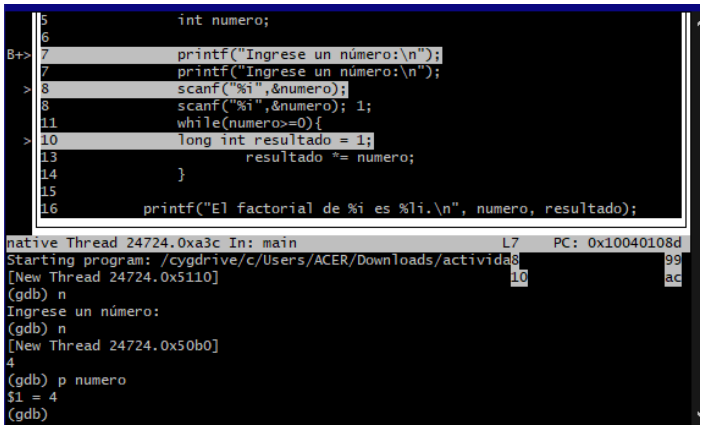
Al compilar y ejecutar el programa me daba otro resultado al que yo esperaba, se supone que el programa me tenia que dar el factorial de un número pero intentando con otros vaores aún me daban cero.

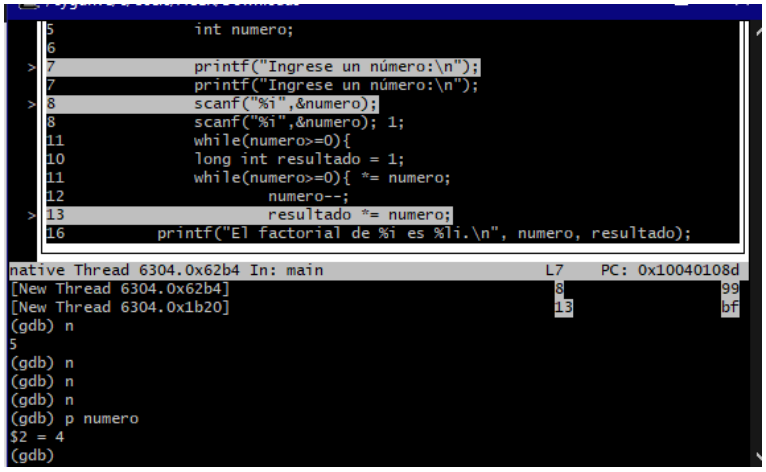
Entonces entrando en el depurador de GDB lo corri y no mostraba un error en el código, un core, hasta que presiones las teclas CTRL+X+A

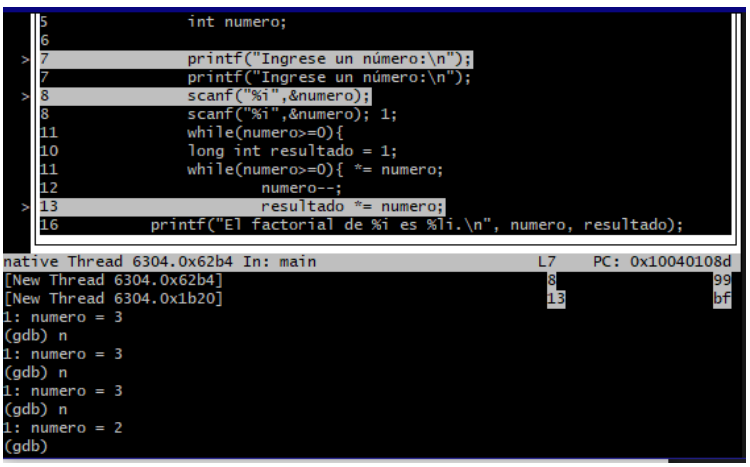
```
/cygdrive/c/Users/ACER/Downloads
[Inferior 1 (process 17976) exited normally]
(gdb) list
1       #include <stdio.h>
2
3       int main()
4       {
5               int numero;
6
7               printf("Ingrese un número:\n");
8               scanf("%i",&numero);
9
10              long int resultado = 1;
(gdb) l
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
(gdb)
```

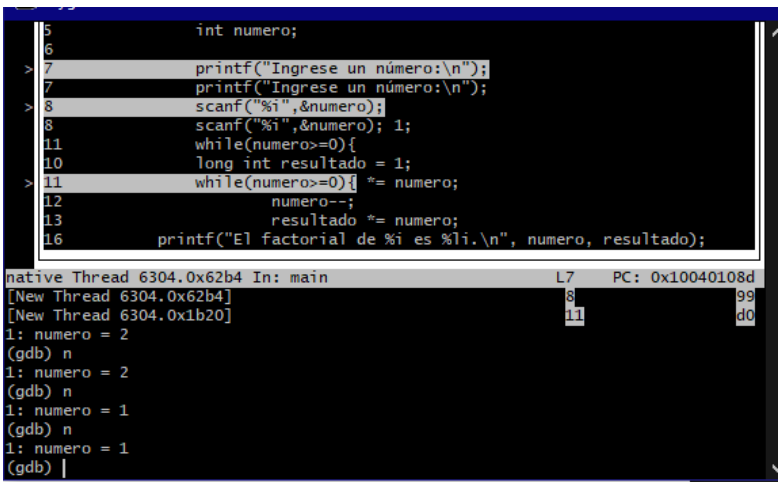


Entrando al interfaz:
Lo comienzo y con n me doy cuenta que no estaba mal escrito el código sino es el procedimiento
Como se muestra las siguientes capturas primero la inicializacion de la variable resultado es cero.
Entrando al while primero reducian la variable numero y no lo multiplicaba por el resultado.
Dentro del While, la condición obligaba a que el numero legara a -1 y no a 1.
Y siempre el resultado va a ser cero por quesimepre se multiplica por cero.









```
5               int numero;
6
7               printf("Ingrese un número:\n");
7               printf("Ingrese un número:\n");
8               scanf("%i",&numero);
8               scanf("%i",&numero); 1;
11              while(numero>=0){
10              long int resultado = 1;
11              while(numero>=0){ *= numero;
12                      numero--;
13                      resultado *= numero;
16          printf("El factorial de %i es %li.\n", numero, resultado);
native Thread 6304.0x62b4 In: main                          L7    PC: 0x10040108d
[New Thread 6304.0x62b4]                                    8                 99
[New Thread 6304.0x1b20]                                    13                bf
1: numero = 0
(gdb) n
1: numero = 0
(gdb) n
1: numero = 0
(gdb) n
1: numero = -1
(gdb)
```

Hasta que lo arregle dentro del código

```
/cygdrive/c/Users/ACER/Downloads
4
El factorial de 1 es 24.

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad3.c -o actividad

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad
Ingrese un número:
4
el factorial de 4
es 24.

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ gcc actividad3.c -o actividad

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$ ./actividad
Ingrese un número:
4
el factorial de 4 es 24.

ACER@EQUIPO /cygdrive/c/Users/ACER/Downloads
$
```

Archivo Edición Formato Ver Ayuda

```c
#include <stdio.h>

int main()
{
        int numero;

        printf("Ingrese un número:\n");
        scanf("%i",&numero);
    printf("el factorial de %i", numero);
        long int resultado = 1;
        while(numero>1){
                resultado *= numero;

                numero--;
        }

    printf(" es %li.\n", resultado);

    return 0;
}
```

Lo primero que arregle fue la condición del While, el número ingresado tiene que ser mayor a 1 ya que el resultado multiplicado por 1 da el mismo resultado, el resultado lo inicialize con valor a 1 para cuando multiplique no de siempre cero.
Dentro del While primero hice que mutiplicara el número con el resultado y asi lo vaya guardando y luego reducir el valor del número.

CONCLUSIONES:

Los depuradores como el GDB nos funciona cuando queremos realizar varias series de inspecciones y modificaciones en tiempo de ejecución en el código, también hay que tomar en cuenta que se puede encontrar en Linux o incluso en Windows si tienes instalado cygwin y esto hace que sea gratis