	Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Esteban Pimentel Alarcón
<i>Asignatura:</i>	Fundamentos de Programación
<i>Grupo:</i>	Grupo #3
<i>No de Práctica(s):</i>	Práctica No.10
<i>Integrante(s):</i>	Gómez Matías Paola Donaji / Laureano González David / Ortiz Luciano Gerson Gael
<i>No. de Equipo de cómputo empleado:</i>	Equipo No. 28
<i>No. de Lista o Brigada:</i>	No. de Lista (Por orden de Nombre): 17 / 25 / 38 No. de Cuenta (Por orden de nombre): #9472 / #9519 / #1949
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	21 de Octubre de 2019
<i>Observaciones:</i>	Tarde entrega. No se cumple con el objetivo de la primer actividad Y en las otras dos actividades no se presentan evidencias de que los programas corregidos funcionen correctamente.

Depuración de Programas

- GDB

Las siglas GDB son una abreviatura para GNU Debugger y es uno de los depuradores más populares para programas en C y C++. GDB ofrece la posibilidad de trazar y modificar la ejecución de un programa.

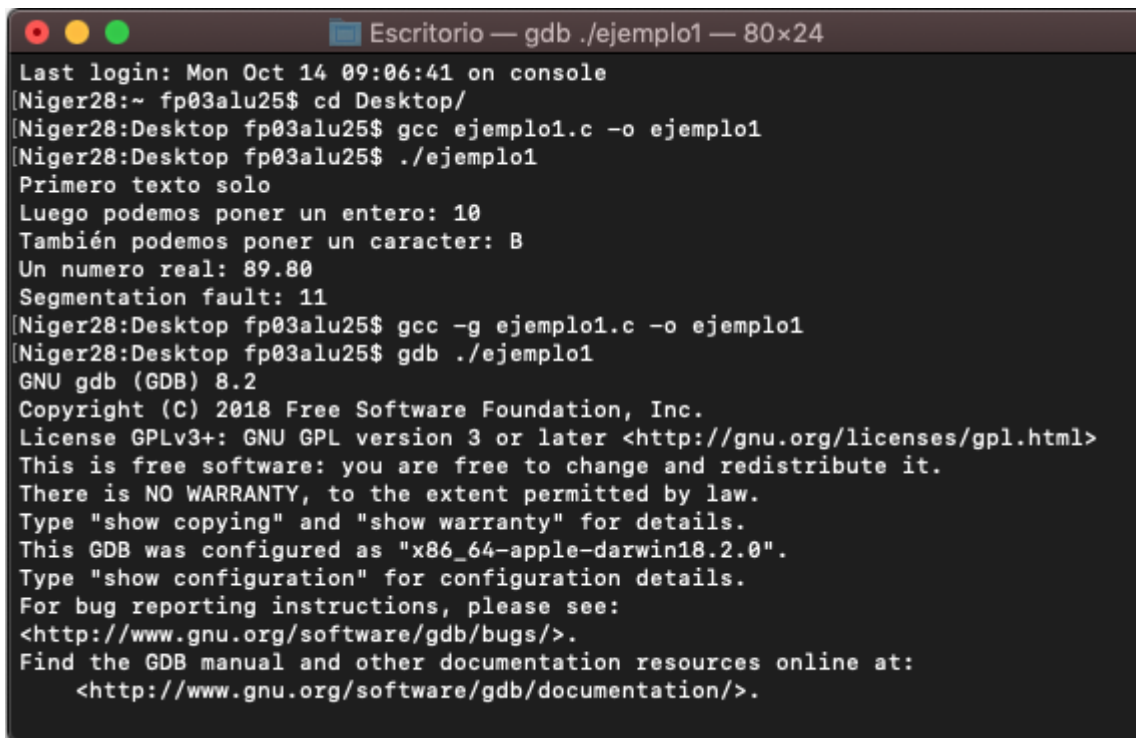
GDB no cuenta con una interfaz gráfica, propia, y esta se controla por medio de una interfaz de línea de comandos.

Objetivo

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

Ejemplo

Compilamos el archivo “ejemplo.c” y después lo corrimos, vimos que estaba corriendo bien, hasta que en la última línea notamos que decía “Segmentation fault: 11”. Después lo compilamos diferente, con un `-g` antes del nombre del archivo y para correrlo utilizamos gdb al principio del comando



```
Escritorio — gdb ./ejemplo1 — 80x24
Last login: Mon Oct 14 09:06:41 on console
[Niger28:~ fp03alu25$ cd Desktop/
[Niger28:Desktop fp03alu25$ gcc ejemplo1.c -o ejemplo1
[Niger28:Desktop fp03alu25$ ./ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un numero real: 89.80
Segmentation fault: 11
[Niger28:Desktop fp03alu25$ gcc -g ejemplo1.c -o ejemplo1
[Niger28:Desktop fp03alu25$ 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/>.
```

Al momento de que lo corrimos, nos salió un menu completo acerca de GDB

```
Escritorio — gdb ./ejemplo1 — 80x24
También podemos poner un caracter: B
Un numero real: 89.80
Segmentation fault: 11
[Niger28:Desktop fp03alu25$ gcc -g ejemplo1.c -o ejemplo1
[Niger28:Desktop fp03alu25$ 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/fp03alu25/Desktop/
ejemplo1.dSYM/Contents/Resources/DWARF/ejemplo1...done.
done.
(gdb) █
```

Tratamos de seguir con el ejemplo pero la terminal de Mac no tenia los elementos suficientes y nos conectamos a una terminal remota de Linux

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
[fp03alu25@samba ~]$ ls
Escritorio actividad1.c actividad2.c actividad3.c ejemplo1.c
[fp03alu25@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[fp03alu25@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/fp03alu25/ejemplo1...done.
(gdb) █
```

Lo compilamos en la terminal remota de Linux y después lo corrimos, nos salió el mismo menu de GDB y pusimos el comando “run” para correr el programa y nos indicó en que línea exactamente había un error en el programa.

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
[[fp03alu25@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[[fp03alu25@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/fp03alu25/ejemplo1...done.
((gdb) run
Starting program: /users/fp03/fp03alu25/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) list
```

Después ocupamos el comando “list”, este nos mostró las líneas donde se encontraba el error y nos mostró el código

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
Reading symbols from /users/fp03/fp03alu25/ejemplo1...done.
((gdb) run
Starting program: /users/fp03/fp03alu25/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) 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) █
```

Para salir de la interfaz de GDB usamos el comando "quit" o solo la letra "q"

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
Un numero real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x0000000040060c in main (argc=19, argv=0x110000010) 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) quit
A debugging session is active.

        Inferior 1 [process 21626] will be killed.

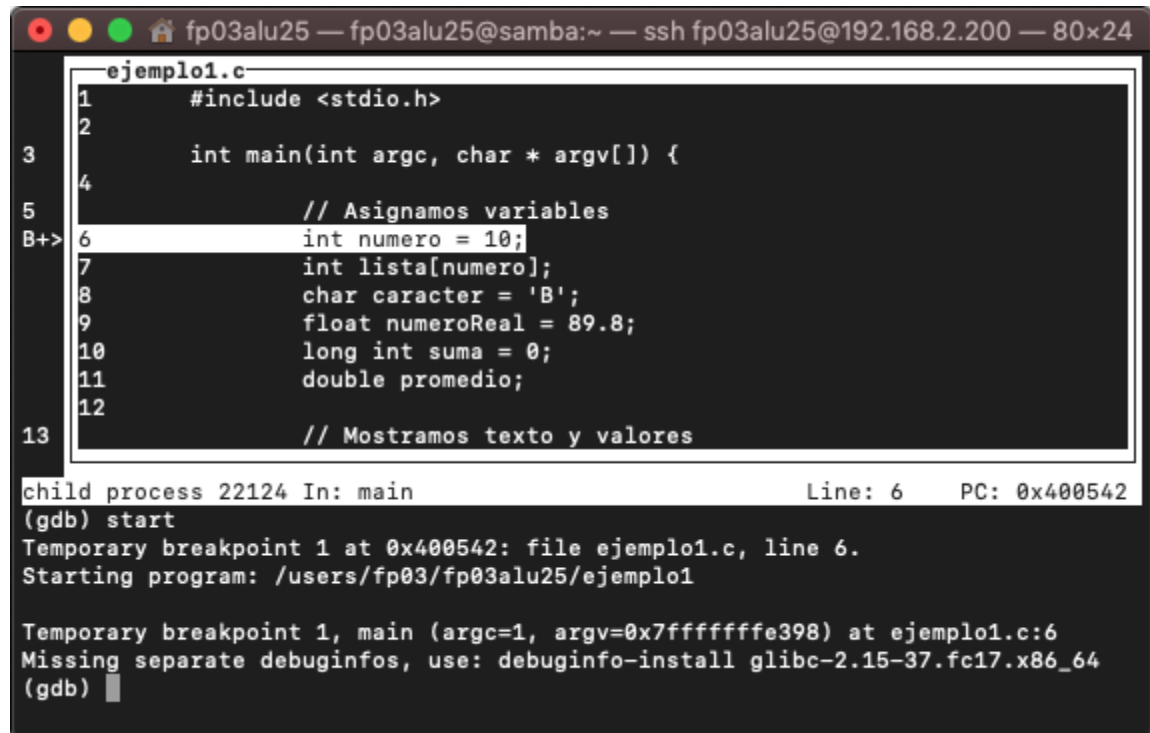
[Quit anyway? (y or n) y
[fp03alu25@samba ~]$
```

Al utilizar el GDB entramos a una terminal la cual al correr el programa nos muestra todas las líneas de este

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
[ No Source Available ]

exec No process In:                               Line: ??   PC: ??
(gdb)
```

Al ya haber compilado el programa, utilizamos el comando start para que nos muestre todo el programa, línea por línea

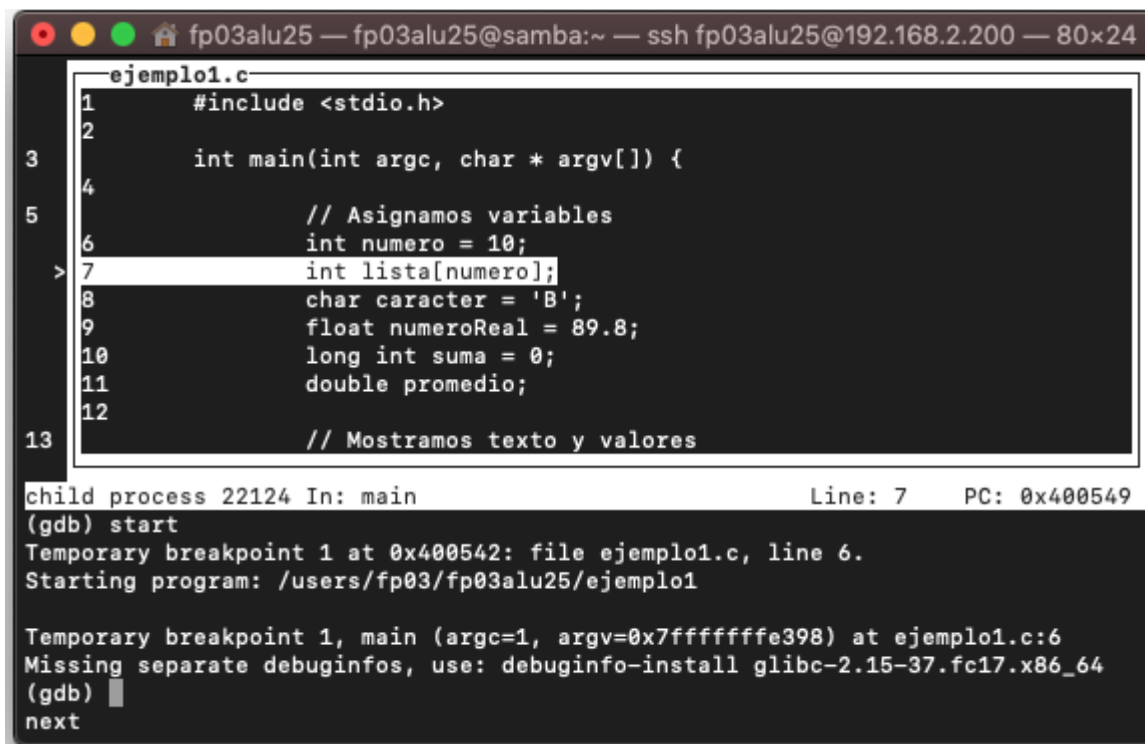


```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
ejemplo1.c
1      #include <stdio.h>
2
3      int main(int argc, char * argv[]) {
4
5          // Asignamos variables
6      B+> int numero = 10;
7          int lista[numero];
8          char caracter = 'B';
9          float numeroReal = 89.8;
10         long int suma = 0;
11         double promedio;
12
13         // Mostramos texto y valores

child process 22124 In: main                                Line: 6      PC: 0x400542
(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu25/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) █
```

Con el comando next podemos movernos entre líneas o podemos cambiar de líneas poniendo solo n



```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
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

child process 22124 In: main                                Line: 7      PC: 0x400549
(gdb) start
Temporary breakpoint 1 at 0x400542: file ejemplo1.c, line 6.
Starting program: /users/fp03/fp03alu25/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
```

Utilizamos el comando break para establecer un punto de parada para ver el programa en este caso fue la línea 20

```
fp03alu25 — fp03alu25@samba:~ — ssh fp03alu25@192.168.2.200 — 80x24
[[fp03alu25@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/fp03alu25/ejemplo1...done.
(gdb) break 20
Breakpoint 1 at 0x4005f7: file ejemplo1.c, line 20.
(gdb) run
Starting program: /users/fp03/fp03alu25/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=0x7fffffffe398) 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
[[fp03alu25@samba ~]]$
```

Por ultimo usamos el comando print, que nos muestra en este caso la lista en la línea en la que estamos

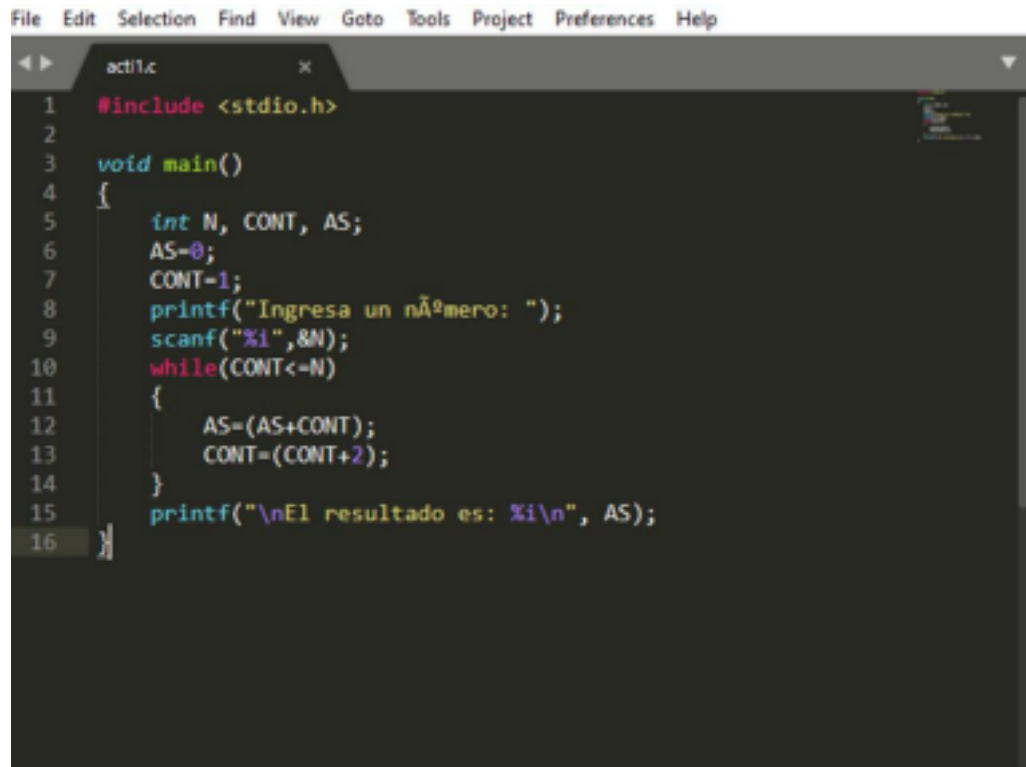
```
ejemplo1.c
15      printf("Luego podemos poner un entero: %i\n", numero);
16      printf("También podemos poner un caracter: %c\n", caracter);
17      printf("Un numero real: %.2f\n", numeroReal);
18
19      // Podemos llenar la lista con valores
20      for(int i = numero ; i >= numero ; i++){
21          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      }

child process 22838 In: main Line: 21 PC: 0x4005ff
(gdb) p i
$1 = 10
(gdb) print lista
$2 = {-163754450, 0, 4195102, 0, -1, 0, -7536, 32767, -7520, 32767}
(gdb) display i
1: i = 10
(gdb)
```


Actividades

- **Actividad No. 1**

Utilizamos GDB para checar el primer archivo, actividad1.c, pero al revisar, vimos que este no tenia ningún error así que lo dejamos como está



```
File Edit Selection Find View Goto Tools Project Preferences Help
activ1.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+1);
14     }
15     printf("\nEl resultado es: %i\n", AS);
16 }
```

Lean la actividad. Este programa no tenía ningún problema, se debía usar GDB para revisar su funcionamiento y describirlo

- **Actividad No. 2**

Utilizamos GDB para checar el segundo archivo, actividad2.c, y encontramos que tenia unos errores, así que corregimos el programa y quedó de la siguiente manera

```
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     //Aquí faltó el & en la variable
12     printf("X=");
13     scanf("%lf",X);
14     K=0;
15     AP=1;
16     AS=0;
17     while(K<N)
18     //Se cambia el = por un < para que el proceso se haga N veces
19     {
20         AS=AS+pow(X,K)/AP;
21         K=K+1;
22         AP=AP*K;
23     }
24     printf("Resultado=Xle",AS)
```

- **Actividad No.3**

Utilizamos GDB para checar el tercer archivo, actividad3.c y encontramos varios errores los cuales corregimos, y el programa queda de la siguiente forma:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int numero,numero2;
6      //Se agrega una variable, porque al final el valor de numero es 0 así que se agrega num
7      printf("Ingrese un número:\n");
8      scanf("%i",&numero);
9      numero2=numero
10     //Se iguala numero2 a numero para mantenet su funcion
11     long int resultado = 1;
12     while(numero2>0){
13         //Se cambia a solo > para evitar la multiplicación por 0
14         resultado *= numero2;
15         numero2--;
16         //Se cambia de posición las funciones para que se reste al final y no al principid
17     }
18
19     printf("El factorial de %i es %li.\n", numero, resultado);
20
21     return 0;
22 }
```