

1. Mostrar el valor de la variable final2 justo antes de que se le asigne ningún valor dentro del bucle for.

Break 21 //Punto de ruptura en la línea antes del bucle

run //Iniciamos el programa

info locals //Mostramos las variables

```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
javi5454@javi5454:~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2$ ls
codigo_c++  codigo_c++.cpp
javi5454@javi5454:~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2$ g
db cod
code      codepage  codigo_c++
javi5454@javi5454:~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2$ g
db codigo_c++
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 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-linux-gnu".
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 codigo_c++...
```

```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
Reading symbols from codigo_c++...
(gdb) list 1,50
1      #include <iostream>
2      /* Incrementa en 2 una variable */
3      int cuenta (int y) {
4          int tmp;
5          tmp = y + 2;
6          return tmp;
7      }
8      /* Calcula la multiplicación de dos números */
9      int multiplica (int x, int y) {
10         int final;
11         int i;
12         final = 0;
13         for (i = 0; i < x; i++) {
14             final = final + y;
15         }
16         return final;
17     }
18     int main (void) {
19         int final1;
20         int final2;
21         int i;
22         final1 = multiplica(3, 2);
```

```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
18     int main (void) {
19         int final1;
20         int final2;
21         int i;
22         final1 = multiplica(3, 2);
23         for (i = 0; i < 100; i++)
--Type <RET> for more, q to quit, c to continue without paging--
24         final2 = cuenta(i);
25         std::cout << final1 << "\n";
26         return 0;
27     }
(gdb) break 21
Breakpoint 1 at 0x1203: file codigo_c++.cpp, line 22.
(gdb) run
Starting program: /home/javi5454/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2/codigo_c++

Breakpoint 1, main () at codigo_c++.cpp:22
22     final1 = multiplica(3, 2);
(gdb) info locals
final1 = 0
final2 = 0
i = 32767
(gdb)
```

2. Haz todas las configuraciones necesarias utilizando el depurador gdb para obtener el valor de la variable final2 cuando i vale 50. Muestra todos los comandos utilizados y el valor de las variables final2 e i

```
break 24 //Punto de ruptura dentro del bucle
run //Iniciamos el programa
info locals //Mostramos las variables
set variable i=50 //Establecemos el valor de la variable i a 50.
info locals //Comprobamos
continue //Continuamos
info locals //Observamos el cambio de valor en la variable i
```

```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
(gdb) break 24
Breakpoint 1 at 0x1222: file codigo_c++.cpp, line 24.
(gdb) run
Starting program: /home/javi5454/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2/codigo_c++

Breakpoint 1, main () at codigo_c++.cpp:24
24      final2 = cuenta(i);
(gdb) info locals
final1 = 6
final2 = 0
i = 0
(gdb) set variable i=50
(gdb) info locals
final1 = 6
final2 = 0
i = 50
(gdb) continue
Continuing.

Breakpoint 1, main () at codigo_c++.cpp:24
24      final2 = cuenta(i);
(gdb) info locals
final1 = 6
```

```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
Starting program: /home/javi5454/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2/codigo_c++

Breakpoint 1, main () at codigo_c++.cpp:24
24      final2 = cuenta(i);
(gdb) info locals
final1 = 6
final2 = 0
i = 0
(gdb) set variable i=50
(gdb) info locals
final1 = 6
final2 = 0
i = 50
(gdb) continue
Continuing.

Breakpoint 1, main () at codigo_c++.cpp:24
24      final2 = cuenta(i);
(gdb) info locals
final1 = 6
final2 = 52
i = 51
(gdb)
```

3. Con el mismo código del ejercicio anterior no parece que la variable final2 pueda ser mayor de 101. Utilizando el depurador gdb haz, que sin tocar la variable final2, esta variable tenga un valor por encima de 1000 al final del programa. Describe todos los pasos que has seguido para conseguirlo e incluye la captura de pantalla correspondiente.

Break 24 //Punto de ruptura línea 24

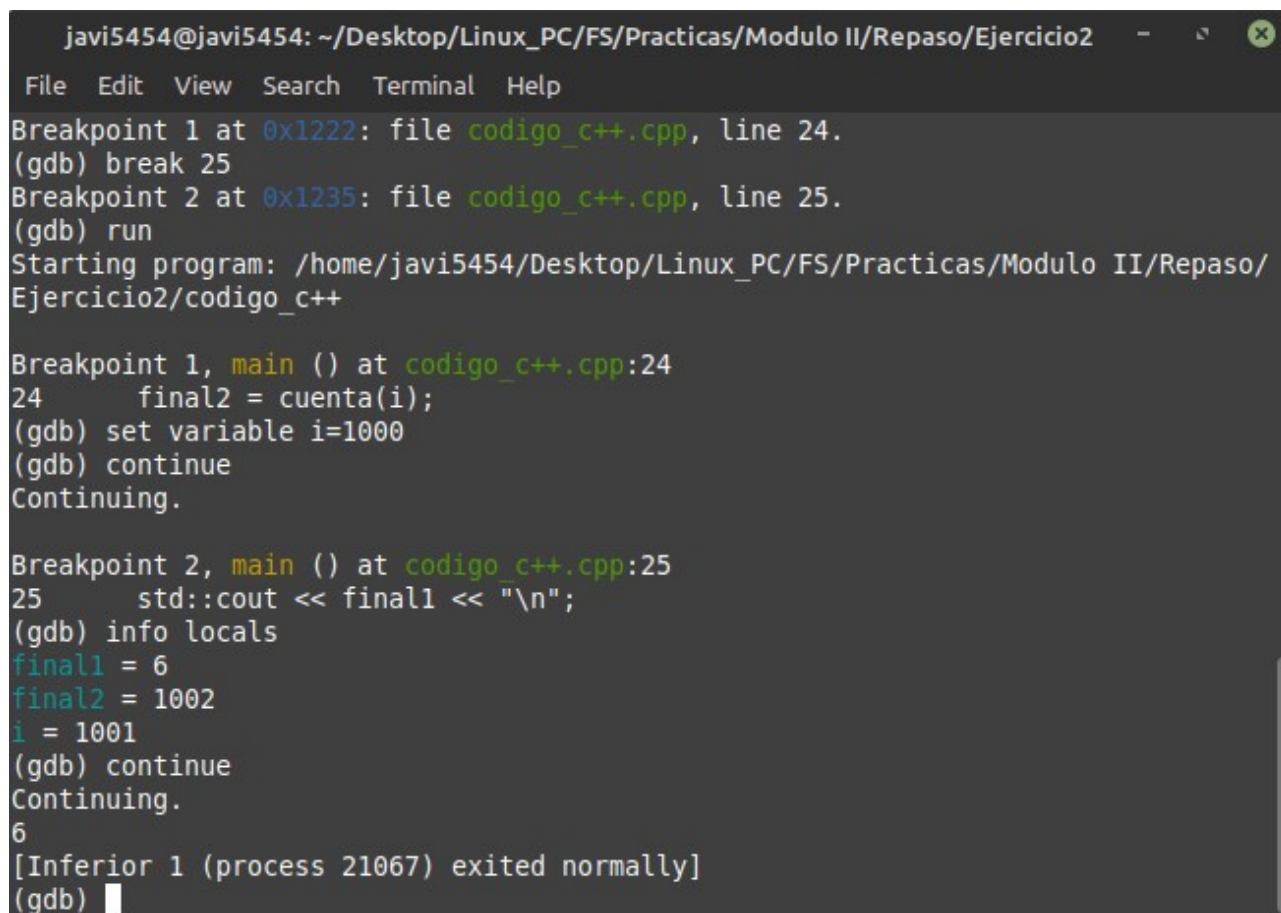
break 25 //Punto de ruptura línea 25

run //Ejecutamos el programa

set variable i=1000 //Ponemos el valor de la variable i a 1000.

continue //Al pasar al siguiente punto de ruptura, se sale del bucle puesto que no se cumple la condición $i < 100$ ($i=1000$).

info locals //Mostramos el valor de las variables.



```
javi5454@javi5454: ~/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2
File Edit View Search Terminal Help
Breakpoint 1 at 0x1222: file codigo_c++.cpp, line 24.
(gdb) break 25
Breakpoint 2 at 0x1235: file codigo_c++.cpp, line 25.
(gdb) run
Starting program: /home/javi5454/Desktop/Linux_PC/FS/Practicas/Modulo II/Repaso/Ejercicio2/codigo_c++

Breakpoint 1, main () at codigo_c++.cpp:24
24     final2 = cuenta(i);
(gdb) set variable i=1000
(gdb) continue
Continuing.

Breakpoint 2, main () at codigo_c++.cpp:25
25     std::cout << final1 << "\n";
(gdb) info locals
final1 = 6
final2 = 1002
i = 1001
(gdb) continue
Continuing.
6
[Inferior 1 (process 21067) exited normally]
(gdb)
```