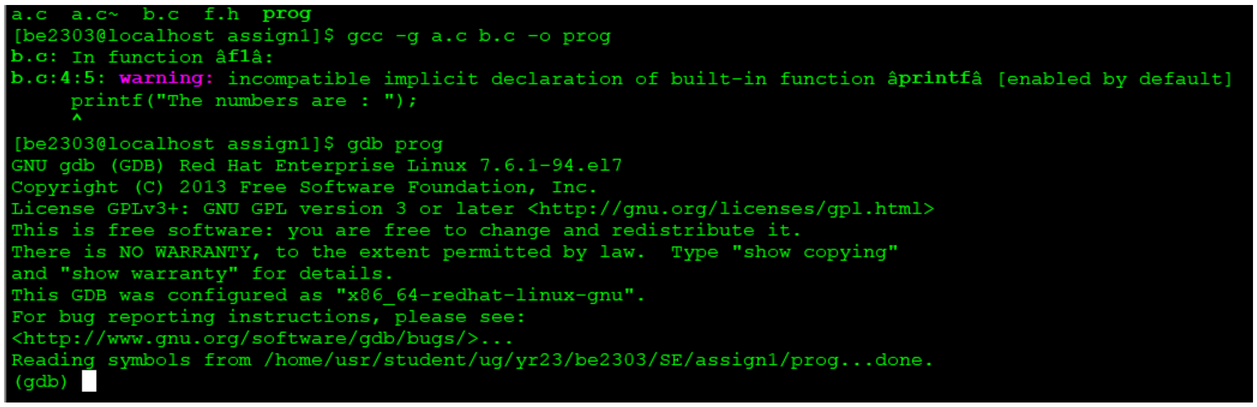
Assignments-1 on gdb

1. Consider the program in folder assign1

a>	Compile it so that it compiles with debugging symbols [using proper option]

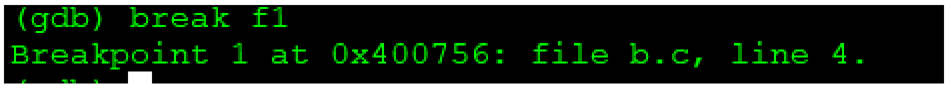
Ans> gcc –g a.c b.c –o prog



```
a.c  a.c~  b.c  f.h  prog
[be2303@localhost assign1]$ gcc -g a.c b.c -o prog
b.c: In function âf1â:
b.c:4:5: warning: incompatible implicit declaration of built-in function âprintfâ [enabled by default]
     printf("The numbers are : ");
     ^
[be2303@localhost assign1]$ gdb prog
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-94.el7
Copyright (C) 2013 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 /home/usr/student/ug/yr23/be2303/SE/assign1/prog...done.
(gdb)
```

b>	Put breakpoint to function f1.

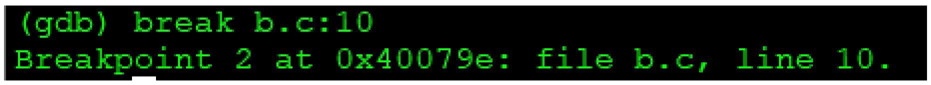
Ans> break f1



```
(gdb) break f1
Breakpoint 1 at 0x400756: file b.c, line 4.
```

c>	Put breakpoint to line 10 of b.c

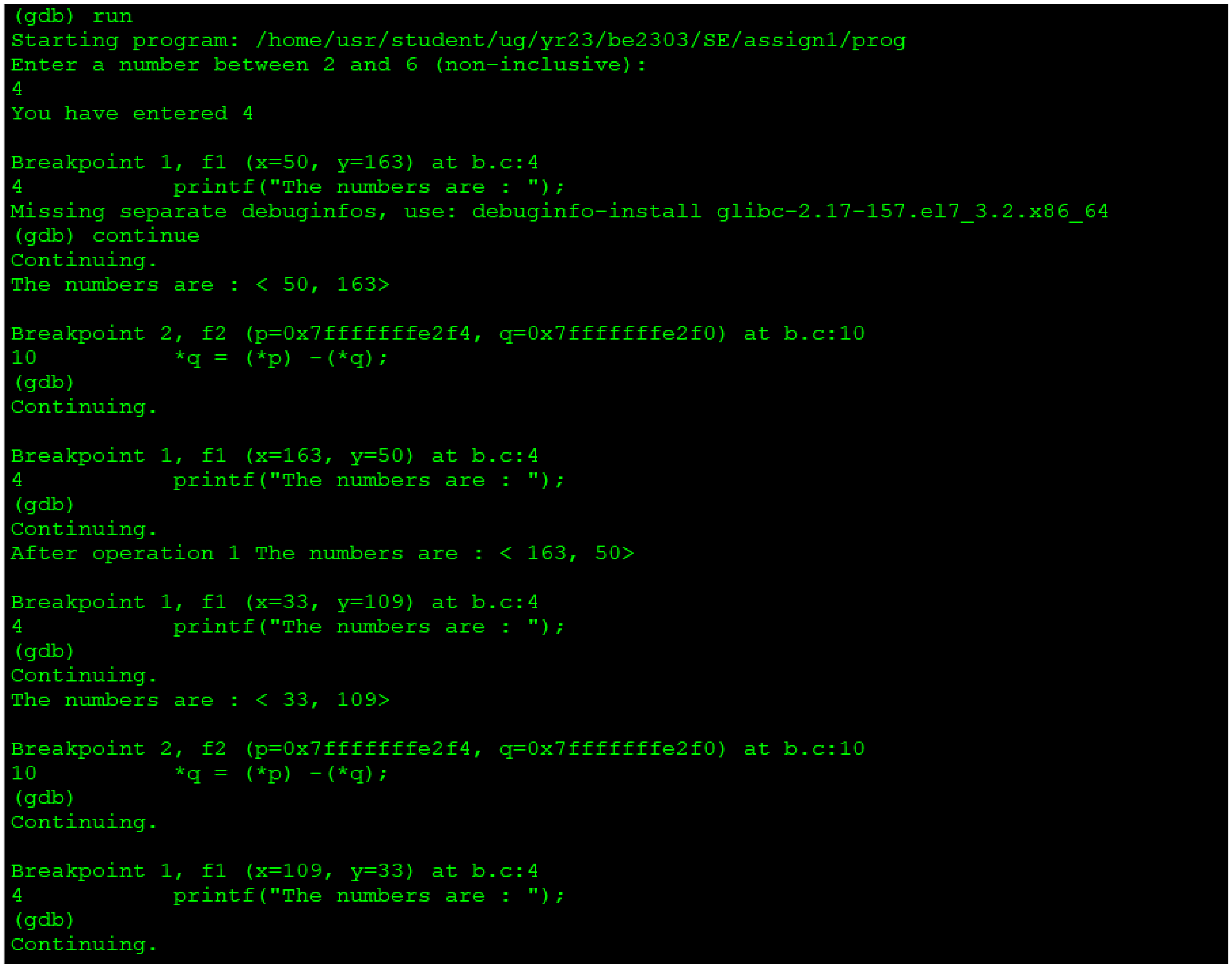
Ans> break b.c:10



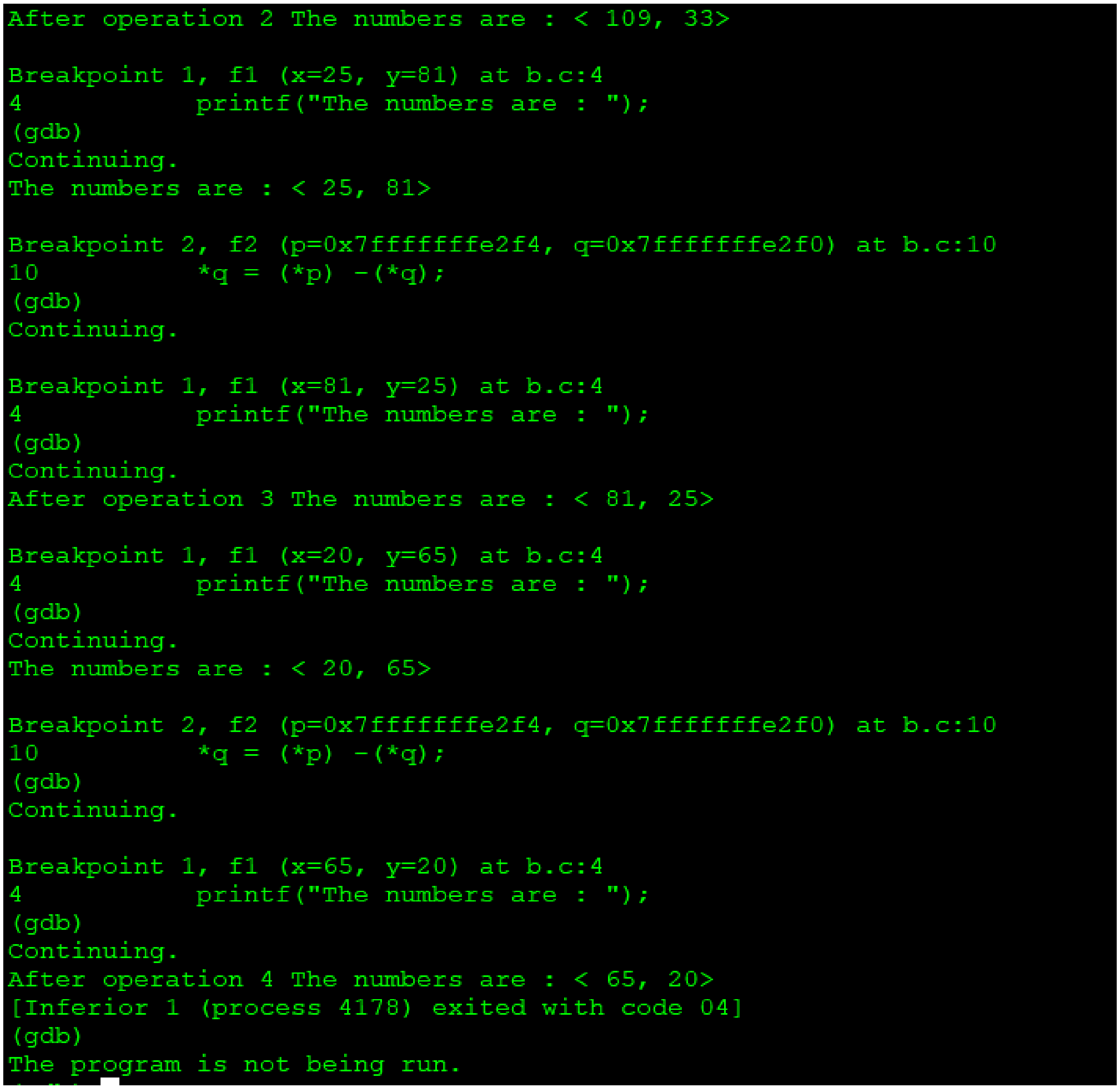
```
(gdb) break b.c:10
Breakpoint 2 at 0x40079e: file b.c, line 10.
```

d> Run the program until it finishes. Which commands are you using to take it to completion?

Ans> run , 4 (any no between 2 and 6 excluding both), continue , enter(till program is not being run)



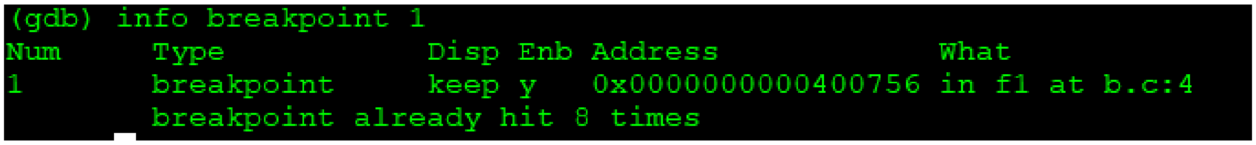
```
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign1/prog
Enter a number between 2 and 6 (non-inclusive):
4
You have entered 4

Breakpoint 1, f1 (x=50, y=163) at b.c:4
4           printf("The numbers are : ");
Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7_3.2.x86_64
(gdb) continue
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe2f4, q=0x7fffffffe2f0) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.

Breakpoint 1, f1 (x=163, y=50) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
After operation 1 The numbers are : < 163, 50>

Breakpoint 1, f1 (x=33, y=109) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 33, 109>

Breakpoint 2, f2 (p=0x7fffffffe2f4, q=0x7fffffffe2f0) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.

Breakpoint 1, f1 (x=109, y=33) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
```



```
After operation 2 The numbers are : < 109, 33>

Breakpoint 1, f1 (x=25, y=81) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 25, 81>

Breakpoint 2, f2 (p=0x7fffffffe2f4, q=0x7fffffffe2f0) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.

Breakpoint 1, f1 (x=81, y=25) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
After operation 3 The numbers are : < 81, 25>

Breakpoint 1, f1 (x=20, y=65) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 20, 65>

Breakpoint 2, f2 (p=0x7fffffffe2f4, q=0x7fffffffe2f0) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.

Breakpoint 1, f1 (x=65, y=20) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
After operation 4 The numbers are : < 65, 20>
[Inferior 1 (process 4178) exited with code 04]
(gdb)
The program is not being run.
```

e> How many times breakpoint “1” is hit in one run of the program ?

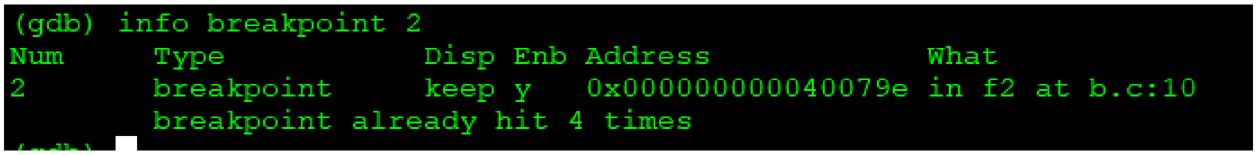
Ans> breakpoint 1 is hit “8” times.



```
(gdb) info breakpoint 1
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000400756 in f1 at b.c:4
        breakpoint already hit 8 times
```

f> How many times breakpoint “2” is hit in one run of the program

Ans> breakpoint 1 is hit “4” times.



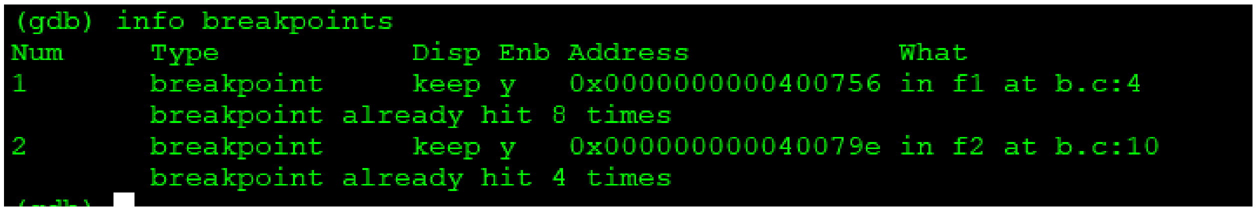
```
(gdb) info breakpoint 2
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x000000000040079e in f2 at b.c:10
        breakpoint already hit 4 times
```

g> How you can see details about a breakpoint ?

Ans> info breakpoint <breakpoint no>

h> How you can see details about all breakpoints ?

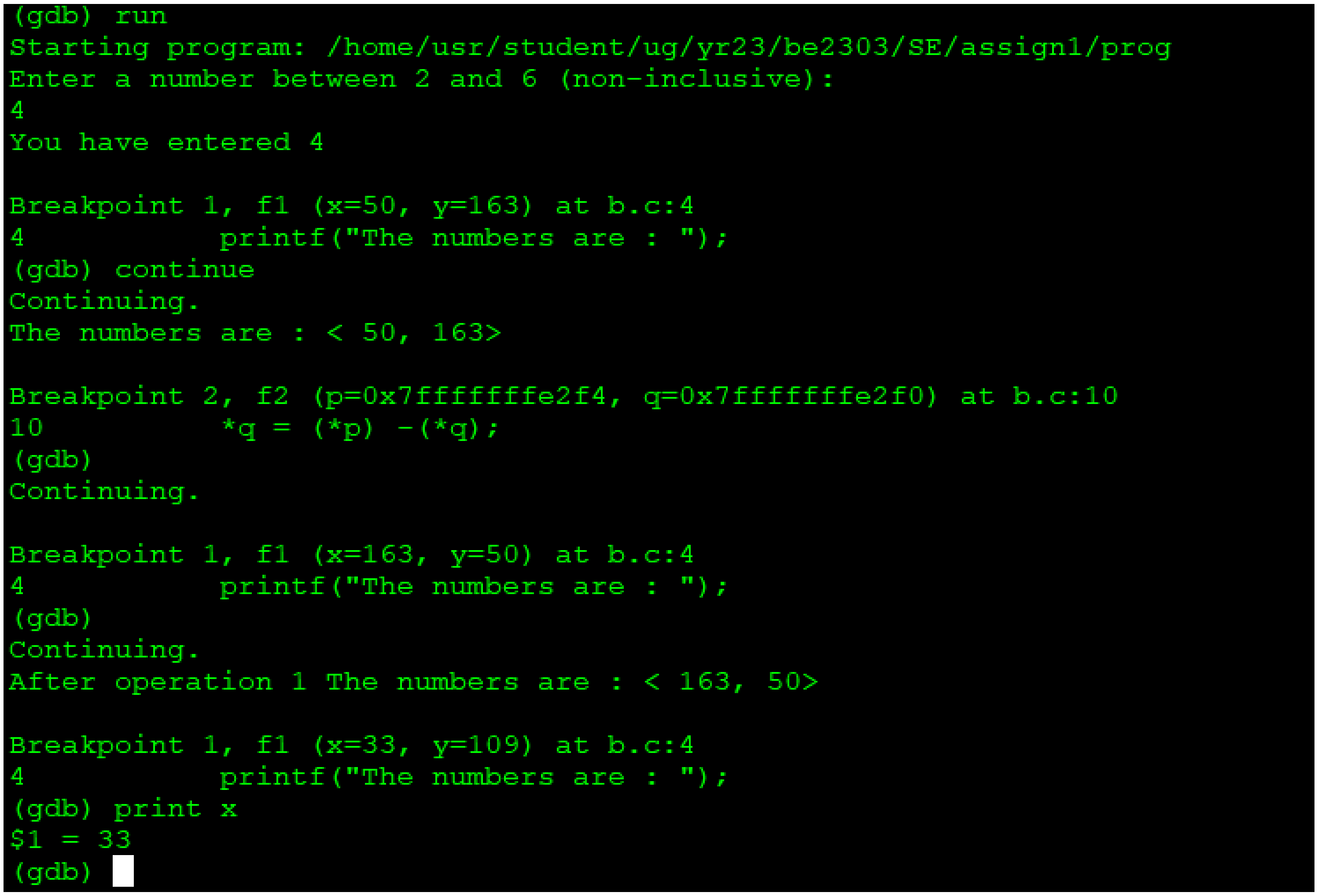
Ans> info breakpoints



```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000400756 in f1 at b.c:4
        breakpoint already hit 8 times
2       breakpoint     keep y   0x000000000040079e in f2 at b.c:10
        breakpoint already hit 4 times
```

i> What is value of variable x in f1 when breakpoint “1” is hit for 3rd time ? How you can examine it ?

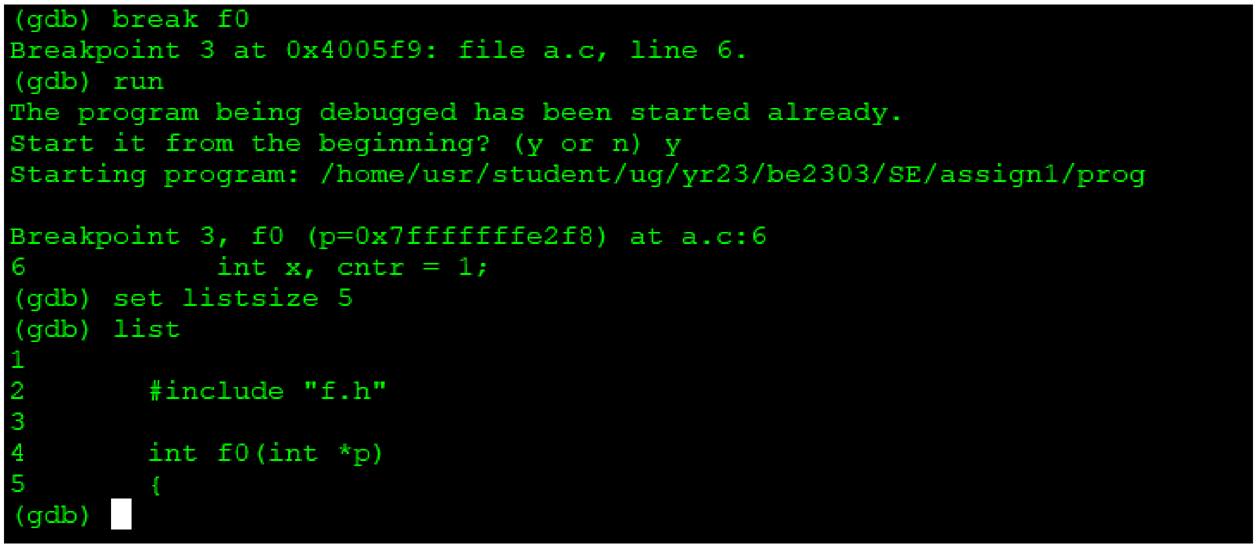
Ans> run, 4 (any number between 2 and 6), c, enter, enter, print x



```
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign1/prog
Enter a number between 2 and 6 (non-inclusive):
4
You have entered 4

Breakpoint 1, f1 (x=50, y=163) at b.c:4
4           printf("The numbers are : ");
(gdb) continue
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe2f4, q=0x7fffffffe2f0) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.

Breakpoint 1, f1 (x=163, y=50) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
After operation 1 The numbers are : < 163, 50>

Breakpoint 1, f1 (x=33, y=109) at b.c:4
4           printf("The numbers are : ");
(gdb) print x
$1 = 33
(gdb) █
```

j> Rerun the program.put a breakpoint at function f0. list 5 lines where it has stopped with breakpoint 3 for first time.

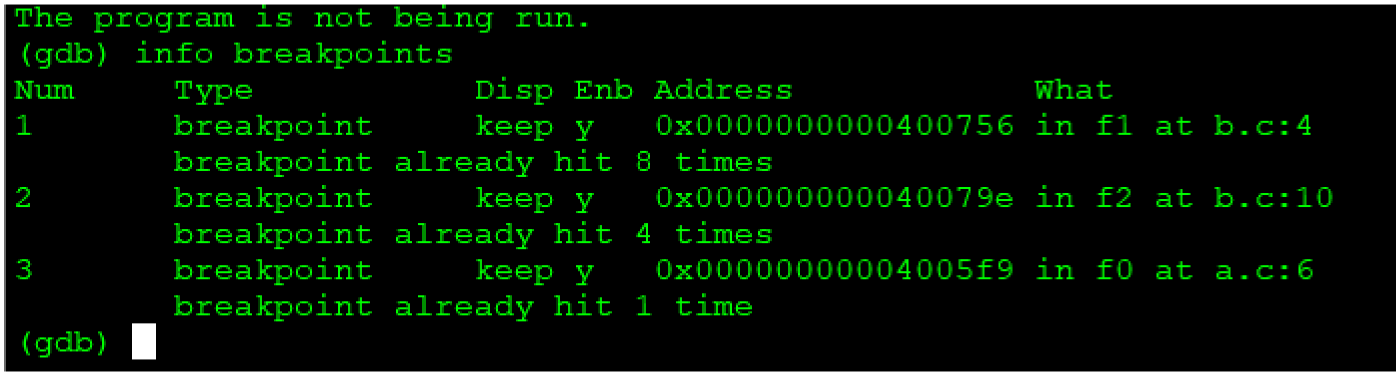
Ans> break f0, run, set listsize 5, list



```
(gdb) break f0
Breakpoint 3 at 0x4005f9: file a.c, line 6.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign1/prog

Breakpoint 3, f0 (p=0x7fffffffe2f8) at a.c:6
6           int x, cntr = 1;
(gdb) set listsize 5
(gdb) list
1
2       #include "f.h"
3
4       int f0(int *p)
5       {
(gdb)
```

Explore : Complete this rerun. Now see what is the change in details of breakpoint s by using command used in “h”

Ans> continue, enter (till cannot run anymore), info breakpoints



```
The program is not being run.
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000400756 in f1 at b.c:4
        breakpoint already hit 8 times
2       breakpoint     keep y   0x000000000040079e in f2 at b.c:10
        breakpoint already hit 4 times
3       breakpoint     keep y   0x00000000004005f9 in f0 at a.c:6
        breakpoint already hit 1 time
(gdb)
```