# ASSIGNMENT-1 ON GDB

## NAME: SAMUDRA ROY   ROLL:002211001114     SEC-A3

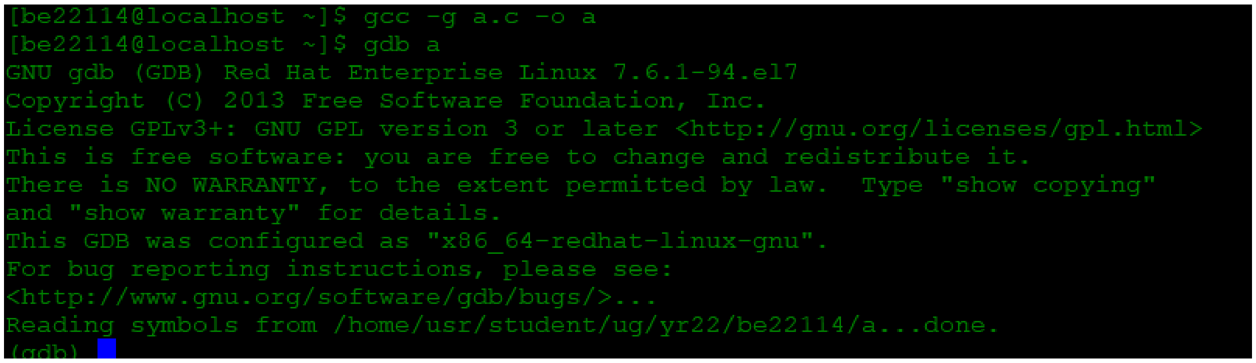
## 1. Consider the program in folder assign1

### a>Compile it so that it compiles with debugging

### symbols [using proper option]

ans: [be22114@localhost ~]$ gcc  -g a.c -o a

[be22114@localhost ~]$ gdb a



```
[be22114@localhost ~]$ gcc -g a.c -o a
[be22114@localhost ~]$ gdb a
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/yr22/be22114/a...done.
(gdb)
```

### b> Put breakpoint to function f1.

(gdb) b f1

Breakpoint 1 at 0x4005fb: file b.c, line 3.

### c> Put breakpoint to line 10 of b.c

(gdb) b b.c:10

Breakpoint 2 at 0x400657: file b.c, line 10.

### d> Run the program until it finishes. Which

**commands are you using to take it to completion?**

**Command:** r (to start execution of the program)

c (continue until the program finishes)

```
(gdb) r

Starting program: /home/usr/student/ug/yr22/be22114/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4


Breakpoint 1, f1 (x=50, y=163) at b.c:3

3         printf("The numbers are : ");

Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7_3.2.x86_64

(gdb) c

Continuing.

The numbers are : < 50, 163>


Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10         *p = (*p) - (*q);

(gdb) c

Continuing.


Breakpoint 1, f1 (x=163, y=50) at b.c:3
```

```
3	printf("The numbers are : ");

(gdb) c

Continuing.

After operation 1 The numbers are : < 163, 50>


Breakpoint 1, f1 (x=33, y=109) at b.c:3

3	printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 33, 109>


Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10	 *p = (*p) - (*q);

(gdb) c

Continuing.


Breakpoint 1, f1 (x=109, y=33) at b.c:3

3	printf("The numbers are : ");

(gdb) c

Continuing.

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

```
Breakpoint 1, f1 (x=25, y=81) at b.c:3

3          printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 25, 81>


Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10          *p = (*p) - (*q);

(gdb) c

Continuing.


Breakpoint 1, f1 (x=81, y=25) at b.c:3

3          printf("The numbers are : ");

(gdb) c

Continuing.

After operation 3 The numbers are : < 81, 25>


Breakpoint 1, f1 (x=20, y=65) at b.c:3

3          printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 20, 65>
```

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10	*p = (*p) - (*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=65, y=20) at b.c:3

3	printf("The numbers are : ");

(gdb) c

Continuing.

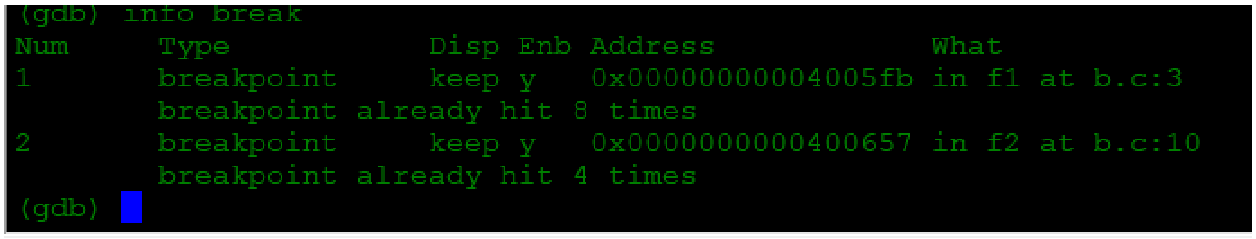
After operation 4 The numbers are : < 65, 20>

[Inferior 1 (process 9856) exited with code 04]

**e> How many times breakpoint "1" is hit in one run of**

**the program ?**

**ans:** 8 times breakpoint 1 hit

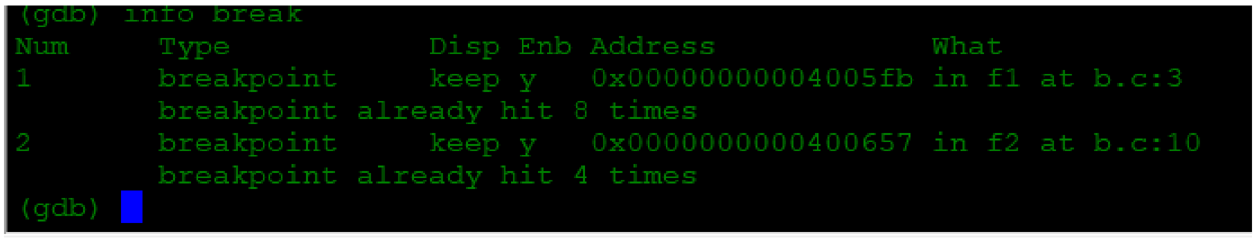


```
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004005fb in f1 at b.c:3
        breakpoint already hit 8 times
2       breakpoint     keep y   0x0000000000400657 in f2 at b.c:10
        breakpoint already hit 4 times
(gdb)
```

**f> How many times breakpoint "2" is hit in one run of**

**the program**

**ans:** breakpoint 2 hit 4 times.



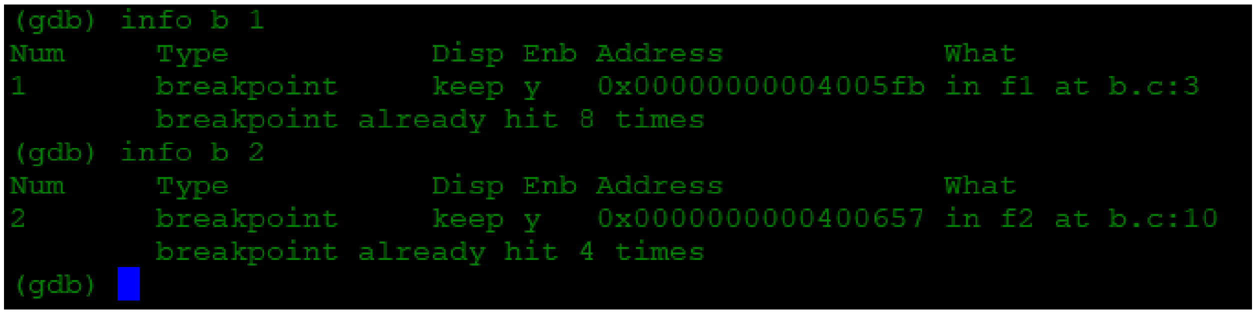
```
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004005fb in f1 at b.c:3
        breakpoint already hit 8 times
2       breakpoint     keep y   0x0000000000400657 in f2 at b.c:10
        breakpoint already hit 4 times
(gdb)
```

**g> How you can see details about a breakpoint ?**

**ans:** info break N (where N is the breakpoint number)

info b 1

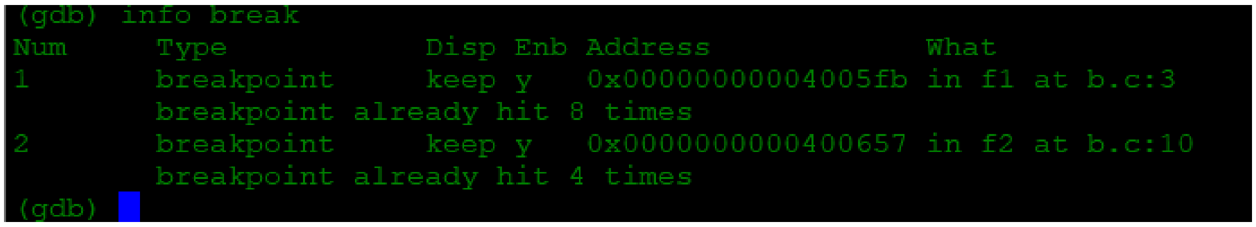
info b 2



```
(gdb) info b 1
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004005fb in f1 at b.c:3
        breakpoint already hit 8 times
(gdb) info b 2
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x0000000000400657 in f2 at b.c:10
        breakpoint already hit 4 times
(gdb)
```

**h> How you can see details about all breakpoints ?**

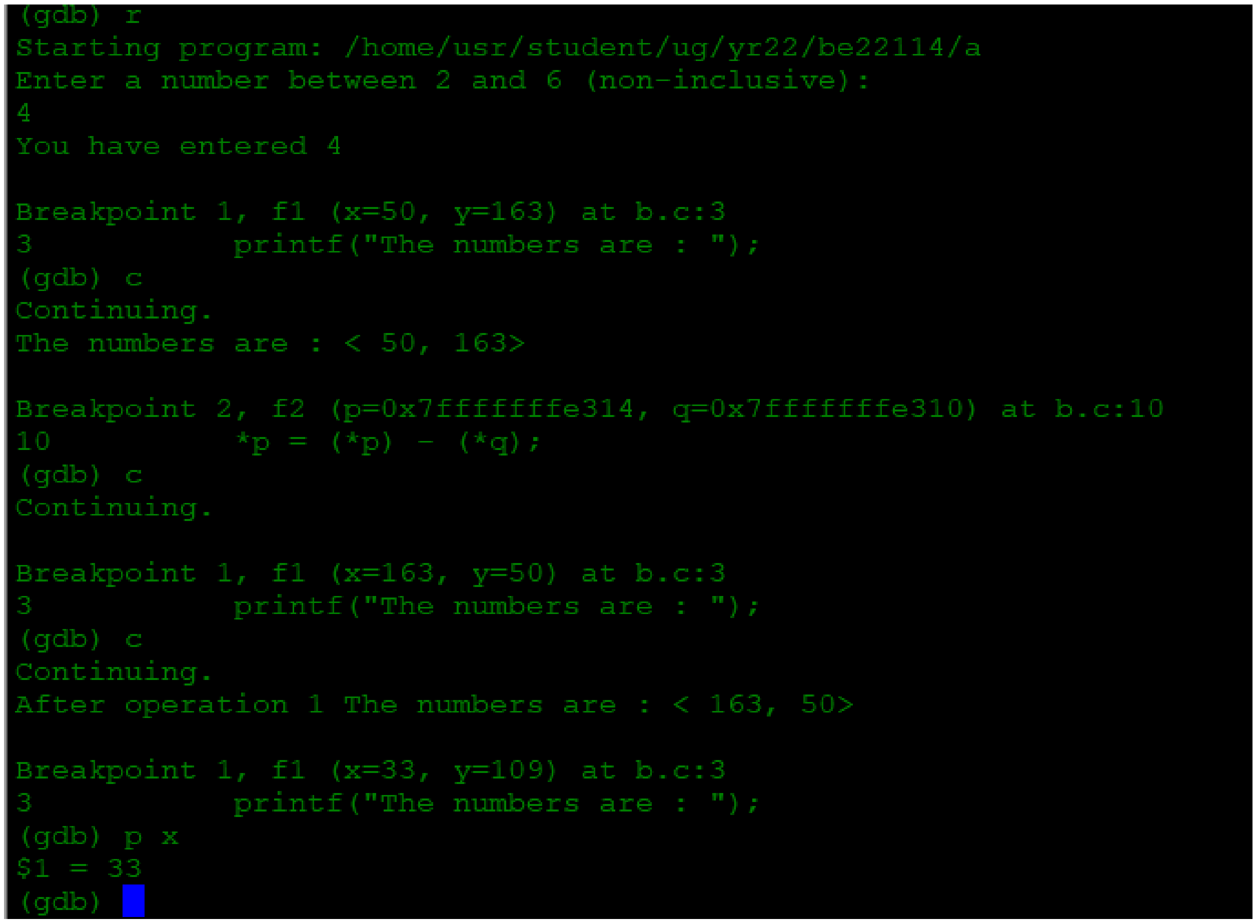
**command:** info break



```
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004005fb in f1 at b.c:3
        breakpoint already hit 8 times
2       breakpoint     keep y   0x0000000000400657 in f2 at b.c:10
        breakpoint already hit 4 times
(gdb)
```

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

**Ans:** Value of x is 33 when breakpoint hits 3$^{rd}$ time



```
(gdb) r
Starting program: /home/usr/student/ug/yr22/be22114/a
Enter a number between 2 and 6 (non-inclusive):
4
You have entered 4

Breakpoint 1, f1 (x=50, y=163) at b.c:3
3           printf("The numbers are : ");
(gdb) c
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10
10          *p = (*p) - (*q);
(gdb) c
Continuing.

Breakpoint 1, f1 (x=163, y=50) at b.c:3
3           printf("The numbers are : ");
(gdb) c
Continuing.
After operation 1 The numbers are : < 163, 50>

Breakpoint 1, f1 (x=33, y=109) at b.c:3
3           printf("The numbers are : ");
(gdb) p 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 firsttime.**

**Ans: command:**

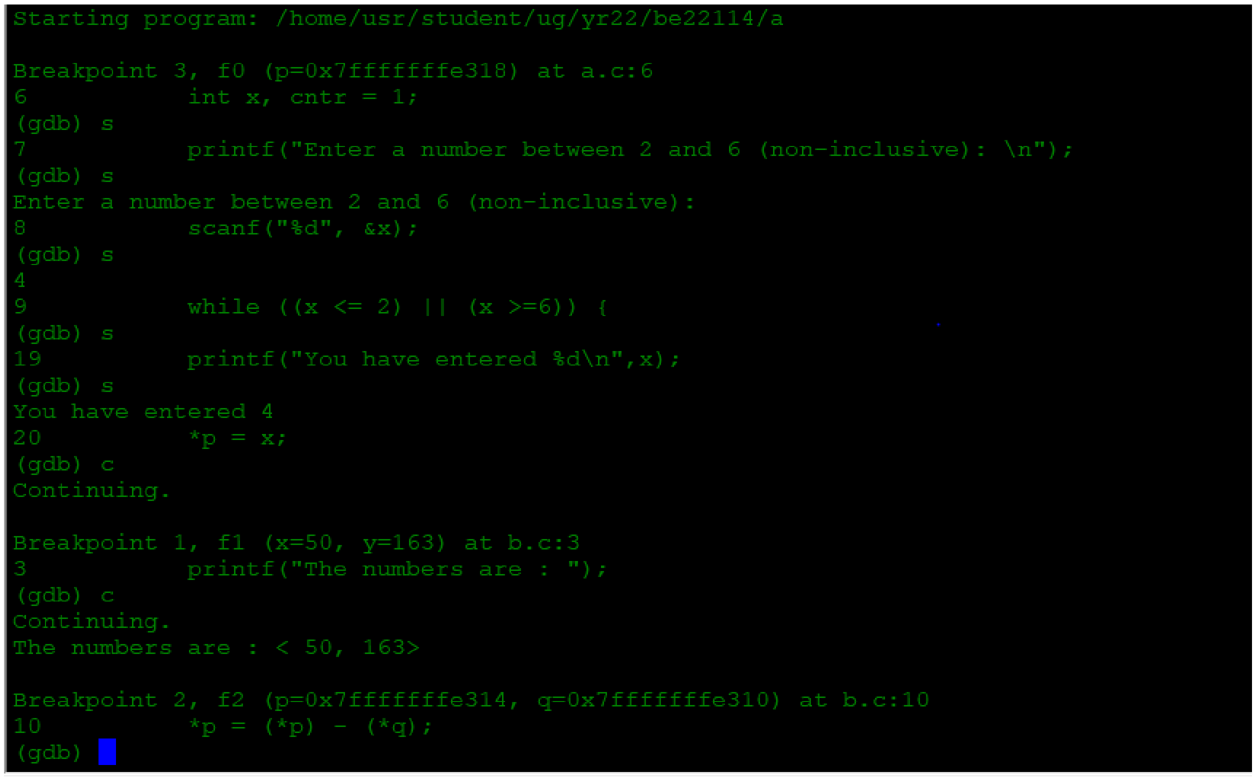
**For breakpoint:** b f0

Breakpoint 3 at 0x400679: file a.c , line 6.

**Rerun**: r (rerun if require press y again)

c (keep executing until the finishing of the program)

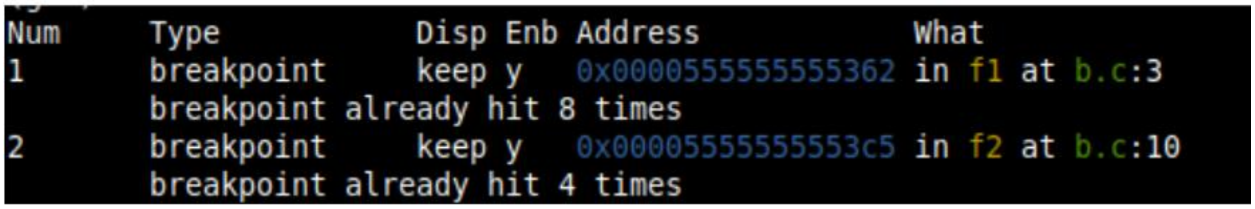
List 5 lines : s (executed n times for n lines here 5 times



```
Starting program: /home/usr/student/ug/yr22/be22114/a

Breakpoint 3, f0 (p=0x7fffffffe318) at a.c:6
6           int x, cntr = 1;
(gdb) s
7           printf("Enter a number between 2 and 6 (non-inclusive): \n");
(gdb) s
Enter a number between 2 and 6 (non-inclusive):
8           scanf("%d", &x);
(gdb) s
4
9           while ((x <= 2) || (x >=6)) {
(gdb) s
19          printf("You have entered %d\n",x);
(gdb) s
You have entered 4
20          *p = x;
(gdb) c
Continuing.

Breakpoint 1, f1 (x=50, y=163) at b.c:3
3           printf("The numbers are : ");
(gdb) c
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10
10          *p = (*p) - (*q);
(gdb) 
```

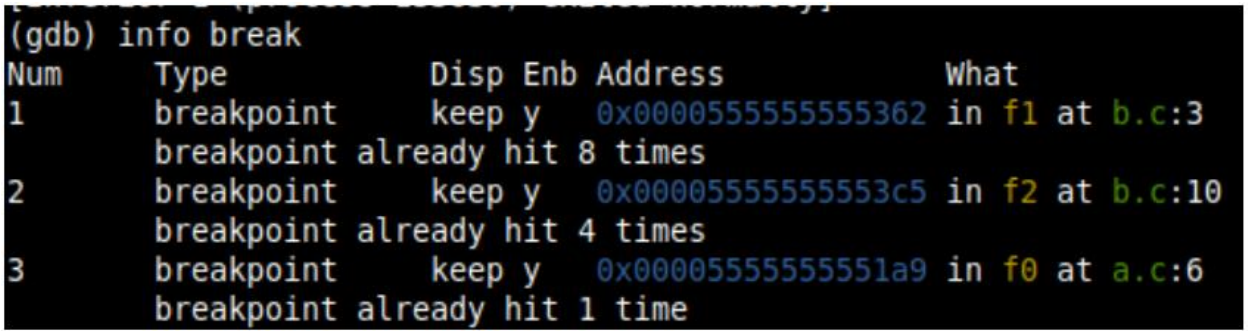
**>> Explore : Complete this rerun. Now see what is the change in details of breakpoint s by using command used in "h"(input is 4)**

Original : the additional breakpoint f3 is hit once



```
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555555362 in f1 at b.c:3
        breakpoint already hit 8 times
2       breakpoint     keep y   0x00005555555553c5 in f2 at b.c:10
        breakpoint already hit 4 times
```

Changed :



```
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555555362 in f1 at b.c:3
        breakpoint already hit 8 times
2       breakpoint     keep y   0x00005555555553c5 in f2 at b.c:10
        breakpoint already hit 4 times
3       breakpoint     keep y   0x00005555555551a9 in f0 at a.c:6
        breakpoint already hit 1 time
```