1. Consider the program in folder assign

A. Compile it so that it compiles with proper debugging symbols.

(Ans.): gcc a.c b.c –g –o out –I./

B. Put breakpoint to function f1.

(Ans): break f1

```
Breakpoint 1 at 0x1177: file b.c, line 4.
```

C. Put breakpoint to line 10 of b.c.

(Ans): break b.c:10

```
Breakpoint 2 at 0x11ca: file b.c, line 10.
```

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

(Ans): continue or c

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

(Ans): 6 times

```
1       breakpoint     keep y   0x0000555555555177 in f1 at b.c:4
        breakpoint already hit 6 times
```

F. How many times breakpoint “2” is hit in one run of the program ?

(Ans): 3 times

```
2       breakpoint     keep y   0x00005555555551ca in f2 at b.c:10
        breakpoint already hit 3 times
```

G. How can you see details about a breakpoint

(Ans): info breakpoint <breakpointNumber>

```
(gdb) info breakpoints 2
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x00005555555551ca in f2 at b.c:10
        breakpoint already hit 3 times
```

H. How can you see details about all breakpoints

(Ans): info breakpoints

```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555555177 in f1 at b.c:4
        breakpoint already hit 6 times
2       breakpoint     keep y   0x00005555555551ca in f2 at b.c:10
        breakpoint already hit 3 times
```

I. What is value of variable x in f1 when breakpoint “1” is hit for the third time? How can you examine it?

(Ans): 33; ignore 1 2, run, continue, print x

```
(gdb) info breakpoints 1
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000555555555177 in f1 at b.c:4
        breakpoint already hit 3 times
(gdb) print x
$1 = 33
```

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

(Ans): set listsize 5, break f0, continue, list

```
(gdb) list
3
4       int f0(int *p)
5       {
6           int x, cntr = 1;
7           printf("Enter a number between 2 and 6 (non-inclusive): \n");
```