## Name:- ATRIJ ROY

## Roll no:- 002311001086

## Class:- IT UG2 A3

### 1. Consider the program in folder assign1

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

Code:-

**gcc -g a.c b.c -o prog**

**gdb prog**

*Output:-*

**GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 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 prog...**

**(gdb)**

### 

### b> Put breakpoint to function f1.

Code:-

**b f1**

*Output:-*

**Breakpoint 1 at 0x1362: file b.c, line 4.**

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

Code:-

**b b.c:10**

*Output:-*

**Breakpoint 2 at 0x13b1: file b.c, line 10.**

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

Code:-

**(gdb)run**

*Output:-*

**Starting program: /home/adminpc/Documents/atr86/Assignments/assign1/prog**

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

**4**

**You have entered 4**

**Breakpoint 1, f1 (x=1766484395, y=882909184) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

**The numbers are : < 50, 163>**

**Breakpoint 2, f2 (p=0x7fffffffdeec, q=0x7fffffffdef0) at b.c:10**

**10 \*p = (\*p) + (\*q);**

**(gdb)c**

**Continuing.**

**Breakpoint 1, f1 (x=32767, y=-8468) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

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

**Breakpoint 1, f1 (x=163, y=50) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

**The numbers are : < 33, 109>**

**Breakpoint 2, f2 (p=0x7fffffffdeec, q=0x7fffffffdef0) at b.c:10**

**10 \*p = (\*p) + (\*q);**

**(gdb) c**

**Continuing.**

**Breakpoint 1, f1 (x=32767, y=-8468) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

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

**Breakpoint 1, f1 (x=109, y=33) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

**The numbers are : < 25, 81>**

**Breakpoint 2, f2 (p=0x7fffffffdeec, q=0x7fffffffdef0) at b.c:10**

**10 \*p = (\*p) + (\*q);**

**(gdb) c**

**Continuing.**

**Breakpoint 1, f1 (x=32767, y=-8468) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

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

**Breakpoint 1, f1 (x=81, y=25) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

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

**Breakpoint 2, f2 (p=0x7fffffffdeec, q=0x7fffffffdef0) at b.c:10**

**10 \*p = (\*p) + (\*q);**

**(gdb) c**

**Continuing.**

**Breakpoint 1, f1 (x=32767, y=-8468) at b.c:4**

**4 {**

**(gdb) c**

**Continuing.**

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

**[Inferior 1 (process 14725) exited normally]**

**(gdb) c**

**The program is not being run.**

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

Code:-

**(gdb) info b 1**

*Output:-*

**Num Type Disp Enb Address What**

**1 breakpoint keep y 0x0000555555555362 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?

Code:-

**gdb) info b 2**

*Output:-*

**Num Type Disp Enb Address What**

**2 breakpoint keep y 0x00005555555553b1 in f2 at b.c:10**

**breakpoint already hit 4 times**

### g> How can you see details about a breakpoint ? Code:-

**info b N**

*Output:-*

**Where N is the no of the particular breakpoint**

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

Code:-

**info b**

*Output:-*

**Num Type Disp Enb Address What**

**1 breakpoint keep y 0x0000555555555362 in f1 at b.c:4**

**breakpoint already hit 8 times**

**2 breakpoint keep y 0x00005555555553b1 in f2 at b.c:10**

**breakpoint already hit 4 times**

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

Code:-

**Breakpoint 1, f1 (x=163, y=50) at b.c:4**

**4 {**

**(gdb) p x**

*Output:-*

**$1 = 163**

**(gdb)**

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

Code:-

**(gdb) list a.c:2, a.c:7**

*Output:-*

**(gdb) list a.c:2, a.c:7**

**2 #include "f.h"**

**3**

**4 int f0(int \*p)**

**5 {**

**6 int x, cntr = 1;**

**7 printf("Enter a number between 2 and 6 (non-inclusive): \n");**

**(gdb)**

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

Code:-

**(gdb) info b**

*Output:-*

**Num Type Disp Enb Address What**

**1 breakpoint keep y 0x0000555555555362 in f1 at b.c:4**

**breakpoint already hit 8 times**

**2 breakpoint keep y 0x00005555555553b1 in f2 at b.c:10**

**breakpoint already hit 4 times**

**3 breakpoint keep y 0x00005555555551a9 in f0 at a.c:5**

**breakpoint already hit 1 times**