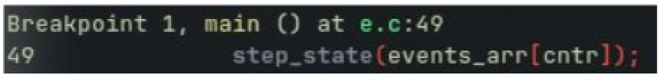
1) Consider the program in Assign3.It is a simple state machine.

A) Put a breakpoint in line 49.

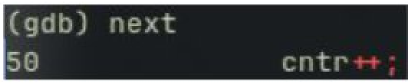
(Ans): break 49



```
Breakpoint 1, main () at e.c:49
49              step_state(events_arr[cntr]);
```

B) try next command

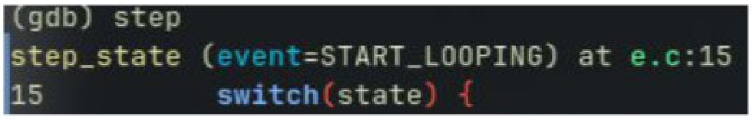
(Ans): next



```
(gdb) next
50              cntr++;
```

C) how will you get inside the function without using breakpoint?

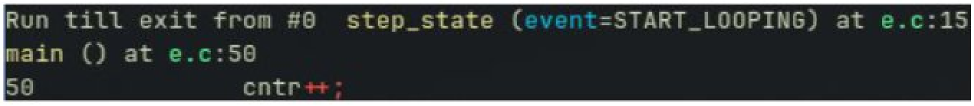
(Ans): step



```
(gdb) step
step_state (event=START_LOOPING) at e.c:15
15          switch(state) {
```

D) How will you get out of the function without using next and continue?

(Ans): finish

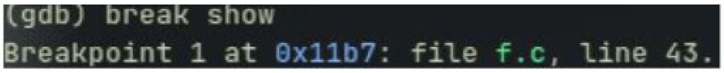


```
Run till exit from #0  step_state (event=START_LOOPING) at e.c:15
main () at e.c:50
50              cntr++;
```

2) Consider the program in Assign4 .It is also a simple state machine.If you provide user id and password properly account details will be displayed. The basic rule is user id should be positive and less than 20 .password is userid *b1000 .The loop will terminate after 10 iteration. It works fine if you provide valid user id and password.It works fine for invalid userid. But it goes to infiniteloop for invalid password.Run the program .It goes into infinite loop.you need to kill the program by [ctrl^c].

A) Set a suitable breakpoint in gdb in the routine show.give valid input and run:

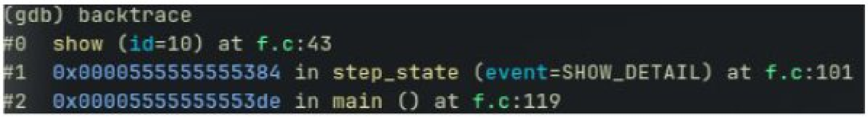
(Ans): break show



```
(gdb) break show
Breakpoint 1 at 0x11b7: file f.c, line 43.
```

B) How you can see the call stack of the routine.

(Ans): backtrace



```
(gdb) backtrace
#0  show (id=10) at f.c:43
#1  0x0000555555555384 in step_state (event=SHOW_DETAIL) at f.c:101
#2  0x00005555555553de in main () at f.c:119
```

C) Which commands will help you to see each value change of variable “event”?

(Ans): watch event (inside foo)

D) Correct the program so that it doesn’t go to infinite loop for wrong password. Rather main iteration restarts . [follow the value change path of event for wrong password]

(Ans): we change state = END in line 96