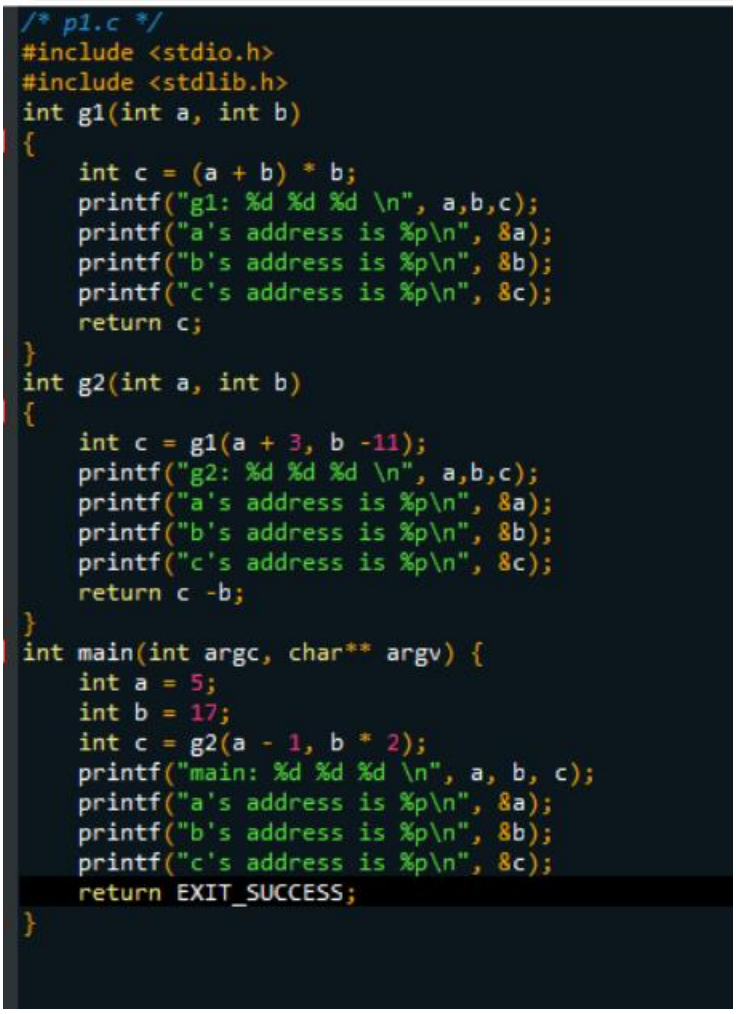
Lab 2

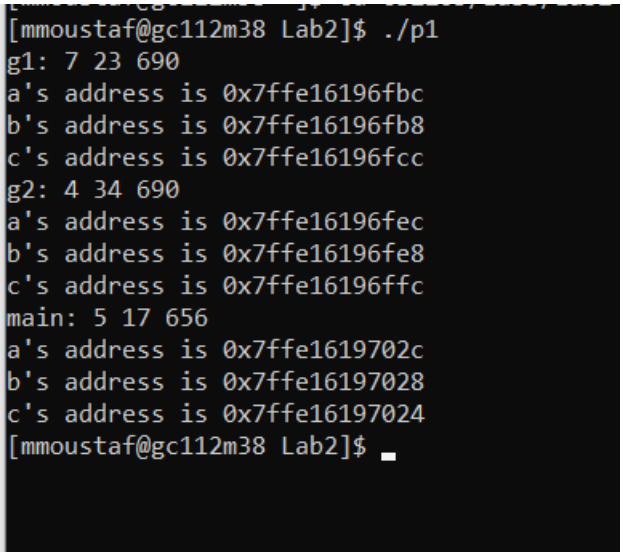
Exercise One

p1.c source code



```c
/* p1.c */
#include <stdio.h>
#include <stdlib.h>
int g1(int a, int b)
{
    int c = (a + b) * b;
    printf("g1: %d %d %d \n", a,b,c);
    printf("a's address is %p\n", &a);
    printf("b's address is %p\n", &b);
    printf("c's address is %p\n", &c);
    return c;
}
int g2(int a, int b)
{
    int c = g1(a + 3, b -11);
    printf("g2: %d %d %d \n", a,b,c);
    printf("a's address is %p\n", &a);
    printf("b's address is %p\n", &b);
    printf("c's address is %p\n", &c);
    return c -b;
}
int main(int argc, char** argv) {
    int a = 5;
    int b = 17;
    int c = g2(a - 1, b * 2);
    printf("main: %d %d %d \n", a, b, c);
    printf("a's address is %p\n", &a);
    printf("b's address is %p\n", &b);
    printf("c's address is %p\n", &c);
    return EXIT_SUCCESS;
}
```

Output



Q&A

1. The values of the variables are the same for all of us because the variables are hard coded in the program, and we all had the same values for their corresponding variables
2. The addresses printed from the programs we different for all of us because the addresses depend on a lot of things such as whether the computer has other processes running and so on.
3. main calls g2 and g2 calls g1. So g2 is put in the call stack first, therefore, the variables in the g2 frame have the bigger addresses, since the addresses are numbered in a way that the largest are at the bottom of the frame.

Exercise two

Backtrace after reaching g2

```
#0  g2 (a=4, b=34) at p1.c:15
#1  0x000000000040066d in main (argc=1, argv=0x7fffffffdee8) at p1.c:25
```

It is showing 2 frames

Backtrace after reaching g1

```
#0  g1 (a=7, b=23) at p1.c:6
#1  0x00000000004005d0 in g2 (a=4, b=34) at p1.c:15
#2  0x000000000040066d in main (argc=1, argv=0x7fffffffdee8) at p1.c:25
```

It is showing 3 frames

Q&A

- They are related. The backtrace command shows the call stack. This call stack has the addresses of the functions where the breakpoints exist (in this case the all the function that is called by the main then that called function calls another one). The frames of these functions have the addresses of the variables. Therefore, they are related.

Exercise Three

Backtrace after reaching isFib

```
#0  isFib (i=1597) at isfib.c:7
#1  0x0000000000400712 in main () at fibprimes.c:21
```

Backtrace after reaching isPrime

```
#0  isPrime (i=1597) at isprime.c:6
#1  0x0000000000400728 in main () at fibprimes.c:21
```