

Kartik Srivastava

Student ID – 3668516

LAB 2

Exercise One:

1. Yes, the values of the variables printed is the same because the data in the variables is the same for everyone.
2. No, the addresses were not the same because a different memory block is allocated for everyone. The addresses also change after each execution.
3. g2 will have a bigger address as it is called first. The variables in g2 will be put in the call stack first.

```
/* 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:

```
g1: 7 23 690  
a's address is 0061FED0  
b's address is 0061FED4  
c's address is 0061FEBC  
g2: 4 34 690  
a's address is 0061FF00  
b's address is 0061FF04  
c's address is 0061FEEC  
main: 5 17 656  
a's address is 0061FF1C  
b's address is 0061FF18  
c's address is 0061FF14
```

Question Two:

```
(gdb) f 1  
#1 0x00000000004005c0 in g2 (a=4, b=34) at p1.c:18  
(gdb) f 0  
#0 g1 (a=7, b=23) at p1.c:8  
(gdb) I
```

- g2 has 1 frame
- g1 has 0 frames.

Question Three:

id415m11.cs.unb.ca:1 (ksrivast) - TigerVNC

Applications DDD: /home1/ugrads/k... Assignment1 Terminal - ksrivast@id... 21:54 EN Kartik Srivastava

DDD: /home1/ugrads/ksrivast/Desktop/CS2263/Assignments/Assignment1/isFib.h

File Edit View Program Commands Status Source Data Help

0: main

```
#include <stdio.h>

int isFib(int num)
{
    int num1;
    int num2;
    int num3;

    if((num == 0) || (num == 1))
    {
        return 0;
        //printf("The number is a fibonacci number \n");
    }

    else
    {
        num1 = 0;
        num2 = 1;
        num3 = num1+num2;

        while(num3 < num)
        {
            num1 = num2;

```

Breakpoint 1, isFib (num=2) at isFib.h:9
(gdb) bt
#0 isFib (num=2) at isFib.h:9
#1 0x000000000040072b in main () at fibPrimes.c:20
(gdb) I

▲ #0 isFib (num=2) at isFib.h:9

Run Interrupt Step Next Step1 Next1 Until Finish Cont Kill Up Down Undo Redo Edit Make