Logo, company name

Description automatically generated

CS2263

Lab 2

February 1st, 2024

UNB Fredericton

**Class** **Code**: Cs2263

**Document**: Lab 2

**Student** **Name**: Will Ross #3734692

**Due** **Date**: February 1st, 2024

**Exercise 1**

*Source Code:*

/\* p1.c \*/

#include <stdio.h>

#include <stdlib.h>

int g1(int a, int b)

{

int c = (a + b) \* b;

printf("g1:\na = %d\nb = %d\nc = %d\n", a, b, c);

printf("g1:\naddress of a = %p\naddress of b = %p\naddress of c = %p\n", &a, &b, &c);

return c;

}

int g2(int a, int b)

{

int c = g1(a + 3, b - 11);

printf("g2:\na = %d\nb = %d\nc = %d\n", a, b, c);

printf("g2:\naddress of a = %p\naddress of b = %p\naddress of c = %p\n",&a,&b,&c);

return c - b;

}

int main(int argc, char \* \* argv)

{

int a = 5;

int b = 17;

int c = g2(a - 1, b \* 2);

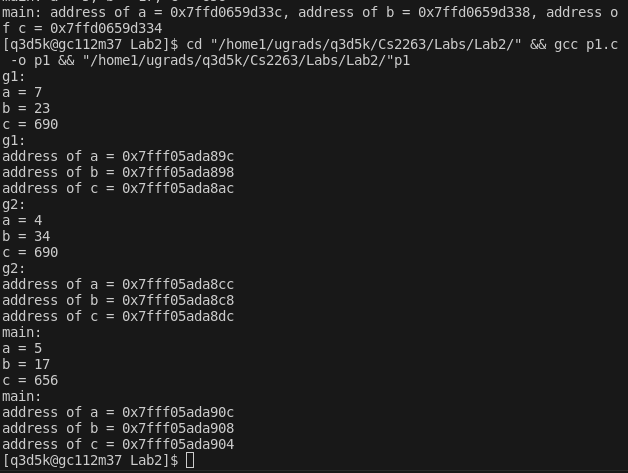
printf("main:\na = %d\nb = %d\nc = %d\n", a, b, c);

printf("main:\naddress of a = %p\naddress of b = %p\naddress of c = %p\n", &a,&b,&c);

return EXIT\_SUCCESS;

}

*Output:*



*Questions:*

**Are the values of the variables printed from your program the same as obtained by your colleagues? Why?**

Yes, the values are the same due to the values being gathered by our program and not hardware dependent.

**- Are the addresses printed from your program the same as obtained by your colleagues? Why?**

They are different from my colleagues due to c being a hardware dependent programming language and will set different variables to different memory locations.

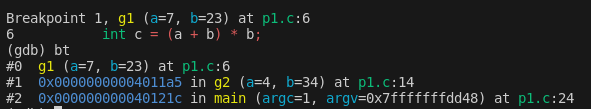
**- Are the addresses printed for the variables used in the function g1 higher or lower than the addresses printed from the function g2? Why?**

In G1 the addresses printed are lower in G1 compared to G2 due to the program when assigning variables to memory addresses, G1 has smaller memory address than G2 because G1 will be higher in the program.

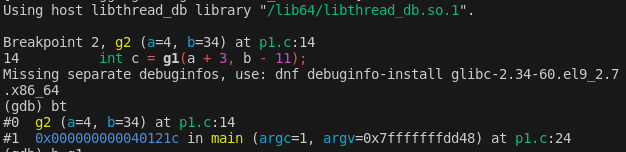
**Exercise 2**

*Output 2.1:*

**G1 bt**

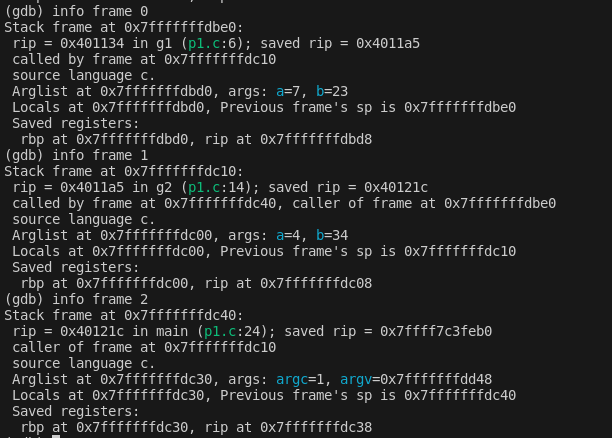


**G2 bt**

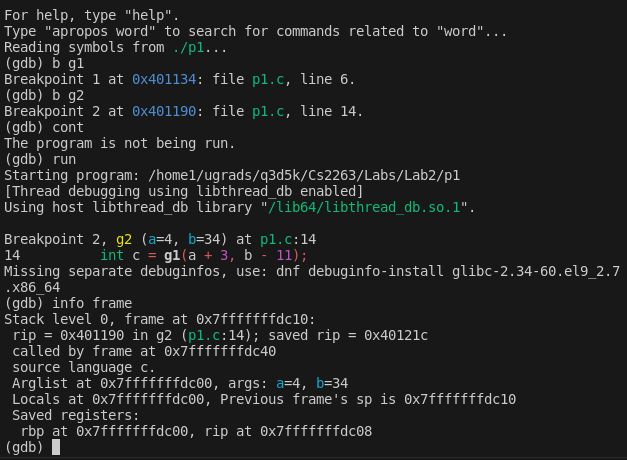


*Output 2.1:*

**Info frame 0,1,2**



**Info frame**



**Are the stack addresses of each frame related to the addresses of the variables a, b and c printed from functions main, g1 and g2?**

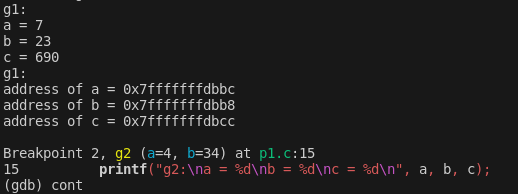
No, they are not the same as the functions from g1 and g2

**Show a memory map indicating the boundaries of each frame and the storage locations used by each variable**

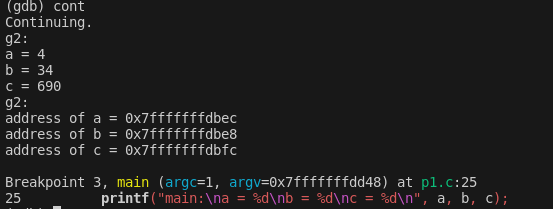
**Memory Map**

|  |  |  |  |
| --- | --- | --- | --- |
| **Frame** | **Symbol** | **Address** | **Value** |
| g1 | a | 0x7fffffffdbbc | 7 |
| g1 | b | 0x7fffffffdbb8 | 23 |
| g1 | c | 0x7fffffffdbcc | 690 |
| g1 | return | 0x7fffffffdbd8 | 0x4011a5 |
| g2 | a | 0x7fffffffdbbc | 4 |
| g2 | b | 0x7fffffffdbe8 | 34 |
| g2 | c | 0x7fffffffdbfc | 690 |
| g2 | return | 0x7fffffffdc08 | 0x40121c |
| main | a | 0x7fffffffdc2c | 5 |
| main | b | 0x7fffffffdc28 | 17 |
| main | c | 0x7fffffffdc24 | 656 |
| main | return | 0x7fffffffdc38 | 0x7ffff7c3feb0 |

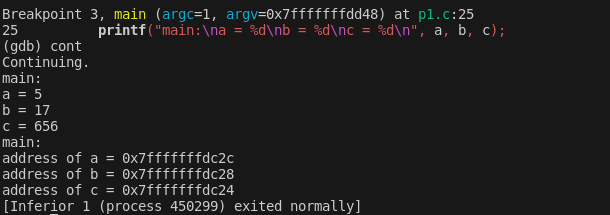
**Info frame G1**



**Info frame G2**



**Info frame main**



**Exercise 3**

*Source Code:*

#include <stdio.h>

int calcTrib(int n) {

printf("%d\n", n);

return n;

}

int main(void){

int numOfValues = 20, num1 = 0, num2 = 0, num3 = 1, num4 = 0;

for(int i = 0; i < numOfValues; i++){

if(i % 2 == 0){

//printf("%d\n", num1);

calcTrib(num1);

}

num4 = num1 + num2 + num3;

num1 = num2;

num2 = num3;

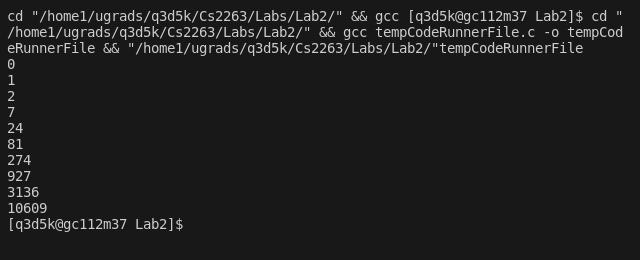
num3 = num4;

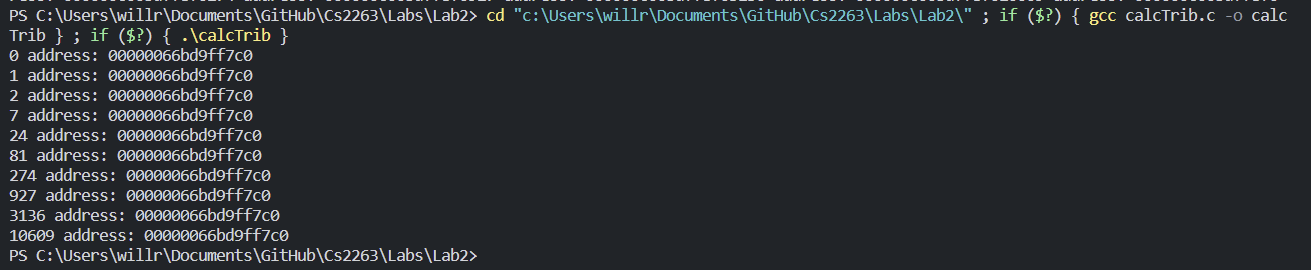
}

}

*Output*

Test results





*Note I complied this later to show that it is the same memory address ^*

breakpoints

