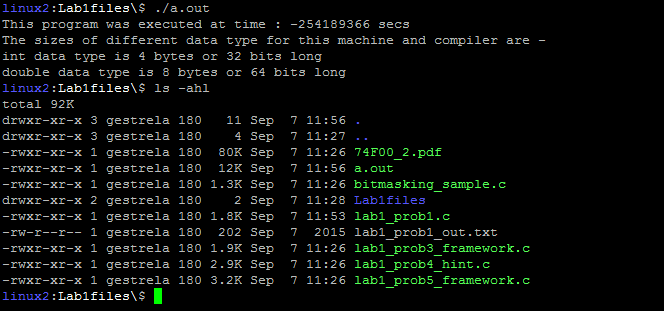
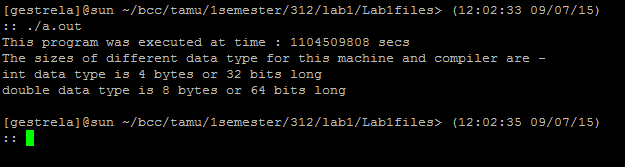
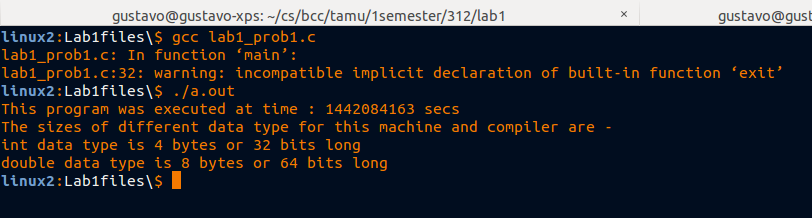
**Gustavo Estrela de Matos CSCE312 LAB1 7 September 2015**

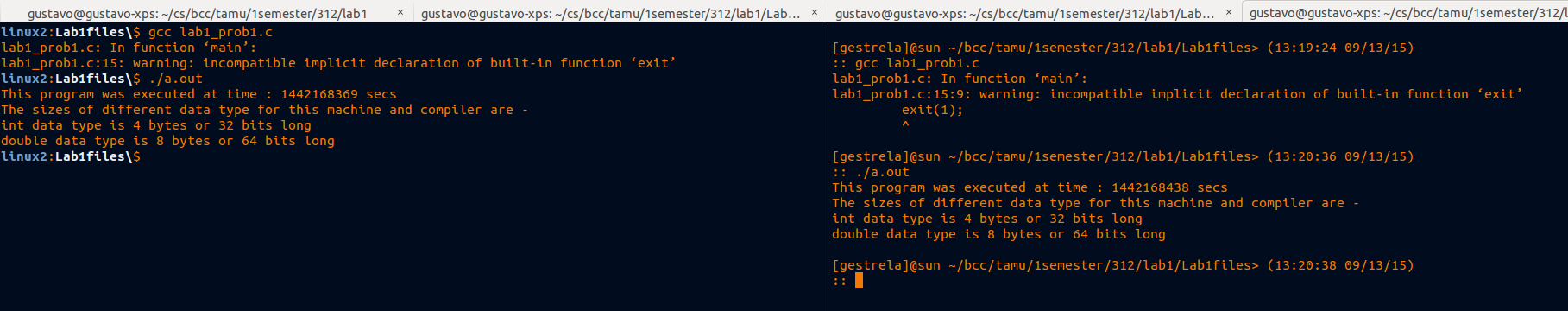
**Problem 1**

1. The statement in “Tag1” creates an integer variable named “int\_var” and the statement in “Tag2” prints the size of “int\_var”, in other words how many bits the variable occupies in the memory
2. The sizeof() function receives as parameters an identifier name or a datatype and give as output the size of the datatype or the size of the datatype of the variable associated to the identifier. This is a standard library, I recompiled the program without including sys libraries and the sizeof function still worked
3. Print:
4. Print:
5. The value was negative because we used double variable and tried to print it as an integer, but integer and double variables have different representations in memory. A possible fix for that is using a casting “(int)” before the variable, which will put the double value in a temporary place using the int data representation. The problem is that, this fix wont work for numbers bigger then the biggest positive number an int can represent (2^31 – 1). We can also try using “unsigned” and “long”, but the type double would still be able to represent numbers bigger then the biggest integer.

Using only casting we get the result:



1. The values are consistent with the one observed i­n the fixed version of the algorithm seen on question e).



1. The timeval structure may have different implementations or different plataforms. On windows, timeval is defined as:

*typedef struct timeval {*

*long tv\_sec;*

*long tv\_usec;*

*} timeval;*

While in linux:

*struct timeval {*

*time\_t tv\_sec;*

*suseconds\_t tv\_usec;*

*};*

**Problem 2**