Assignment 1

This assignment is based on gdb and objdump.

1) Write a program that includes the following:
a) a recursive function that has 2 local variables, and takes 1 input argument
b) 4 initialized global variables,
c) 3 uninitialized global variables,
d) uses at least 2 malloc statements to dynamically allocate space of 10 bytes to char*
e) reads a command-line argument from the user (i.e. in argv) and passes it to the recursive function

The program should output:
1) address of the main function and the recursive function
`e.g. printf("_main  @ %lx\n",(long unsigned int)&main);`
2) address of the initialized static variables and the uninitialized static variables
3) address of local variable at each recursive call

```
gcc –o recurse <program.c>
./recurse
objdump –D –S recurse
```

A) Based on the above answer the following questions:
1) Which of the addresses printed by your program are defined in the output of objdump?
2) Based on the output shown by the program, indicate what direction is the stack growing in [lower to higher addresses or vice-versa]?
3) How large is the stack frame for each recursive call? [Google and find out what is a stackframe]
4) Where is the heap? What direction is it growing in?
5) Are the two malloc()ed memory areas contiguous? (e.g. is there any extra space between their addresses?)
6) Load up your program executable in gdb, set a breakpoint at main, start your program, and continue one line at a time until you are in the middle of your program's execution (i.e. after few recursive calls have been made). Take a look at the stack using backtrace (bt). While you are looking through gdb, answer the following questions: (hint: print var-name)

(a) Value of argv?
(b) Value pointed to by argv?
(c) Address of the function main? Type "info address main"
(d) Type "info stack" in gdb. Explain what you see.
(e) Type "info frame". Explain what you see.

7) In the output of objdump,

(a) What segment contains main function and what is the address of main? (is it same as what you saw in gdb)
(b) Do you see the stack segment anywhere? What about the heap? Explain.