Program Execution

* Considering the process of preparing a C program for execution, describe the function of each of the following pieces of software plays in that process. Limit your description of each to 20 words or less.
  + Linker 4
    - Takes object (.O) and creates executable file
  + Assembler 3
    - Puts compiled assembly file (.S) together and outputs object (.O)
  + Compiler 2
    - Compiles preprocessed source into assembly
  + Loader 5
    - Takes executable file and puts it in memory (and often runs it)
  + macro preprocessor 1
    - Prepares source to be compiled
* Within the context of the process of preparing a C program for execution, draw a block diagram showing the program flow, in order, for each of the following stages of that process. For each block, show the form of the program entering that block, and the form of the program leaving that block.
  + Macro Preprocessor
    - Source goes in
    - Preprocessed source comes out
  + Compiler
    - Preprocessed source goes in
    - ASM .S comes out
  + Assembler
    - ASM .S
    - Object .O
  + Linker
    - Object .O
    - Executable a.out
  + Loader
    - Executable a.out
    - Executable Program in memory
* Draw, and label a diagram that indicates the logical division of memory among program code, stack, and heap.
  + Stack. Variables go here (I; roll; counts;)
  + Stack and heap push here.
  + Heap. Variable memory assignment here (Space for 13 ints – pointed to by counts)
  + Static space. Static stuff here (Num “There were %d rolls of %d\n”)
* Describe how, during program execution, the function of the stack and the heap differ.
* Given a short C program, consisting only of the function main, indicate which variables will reside in the runtime stack during execution and which will be stored in the heap.
  + Variables go in stack
  + Variable pointers go in heap
* Explain, in 20 words or less, why a runtime stack is needed to support execution of a C program.
  + Purpose of a runtime stack is to *store the return addresses.*
  + Stores locals and such as well.