Chris Mathew
y86 Emulator

1. Algorithm
    1. Started by first opening the file
    2. Then I parsed the file matching the components to their respective functions for example, if I encountered .text I would realize the first value after that would be the starting index for the pc
    3. I used string tokenizer to break the file into parts, and then made functions to break those parts down even further
    4. I then stored the components into memory which was a char array
    5. Then I proceeded with the fetch, decode, and execute model where I fetched the instruction form memory, I decoded it to the y86 instruction, and then I executed the function.
2. Problems
    1. I kept running into issues of accessing memory that wasn't there, trying to figure out where i had stored a set on instructions, how each instruction actually worked and the underlying mechanisms behind them. This was one very hard project. First of string manipulation in C is horrendous. It took me a while to finally be able to parse the document correctly. The directives, weren't bad but the .string directive was a challenge. I kept getting "???" before and after my string. The instructions were also very hard to code into C because I kept getting the registers, and the location of memory wrong. I also had issues with movsbl. I could not get it to work tho i made the program still work by substituting it for another instruction.