

Belinda Huang & Mugdha Danda

Project 1 – Assembler Overview:

This assembler first starts by opening the input file. It reads the contents and stores any labels if they exist. Here, it checks if the label is a duplicate or if it's invalid. Then it starts over at the beginning of the input file and reads line by line. For each line, it checks what the instruction is. It translates the opcode and the fields into machine language according to what type of instruction it is. The translated line will then be written to the output file. If there is an error, the program will stop running and nothing will be written to the output file.

Difficulties:

- Coding in C, without having any prior knowledge of it.
- Storing the label in the array, preventing opcodes and decimals from entering the array.
- Error checking for unrecognized opcodes.
- Figuring out and dealing with the segmentation error, when we were using file pointers to open a single file.
- Adding extra conditions for offset values in I-type instructions without getting segmentation error and error checking, whether the offset fits in 16 bits.
- Adding a special condition in I-type instructions when the offset is negative value was also challenging.
- Using so many if-statements, made us feel that our code is not efficient and coming in terms with that was hard.