Lab Assignment - 3;

Name - Soham Rajesh Pawar;
Roll number - CS22BTECH11055;
Course ID - CS2323;
Computer Architecture;

General train of thought :

1) Reading the input from the input file one line at a time till empty;
2) Converting the hexadecimal string acquired in step 1 to a binary string using a custom converter function;
3) Obtaining the opcode from the binary string using a custom substring function;
4) Using the opcode acquired in step 3 to identify the type of instruction for the purpose of separating the string into different parts according to the RISC V cheat sheet provided by the teacher;
5) Constructing the assembly instruction using the cheat sheet;
6) Appending the instruction thus created to an array of character pointers for the purpose of printing;
7) Printing the final result and checking the result by entering in RIPES simulator and comparing the relevant machine code created in the debug window;

Problems faced :

1) How to position the labels in B/J instructions?
-> Using a program counter(that increments every time a statement is appended to the array) and the offset(calculated using the immediate value) to get the relevant position of the label in the assembly code. Then if the position already had a label(i.e the label was already created by another statement pointing to the same line in the assembly code) we don’t do anything. If not present we create a custom label using an arbitrary convention(hard coded) and insert it at the beginning of the string at that position;