During this assignment, we need to create two programs. One of them is used to compress the input file and another is used to decompress the compressed file.

For the first program, there are two things we need to do. We need to create the tree used to compress the file. The basic topic is to use Huffman Coding. The way I do it is a little bit complex. I make the node structure a little different than we normally use. I add another information -character- inside of it. So the node can save int and char at same time. Than using a for loop to save each character and count the time they appeared. Than using another loop to move the node that have smallest int number to the end of the array and combine them. By this way I create the tree used for Huffman Coding. One important thing is I need to print the tree structure into the output file. I choose to use the 1, 0 and char structure to do it. 1 and 0 show the way to move in a tree. The char is the character saved in that node. By this way I can rebuild the tree by just read the information one times. I choose to use a space character to separate the tree and compressed information since there will have no space in the information. So the space can work good. Another part in this program is just compress the file by using the tree we build before. It is sample binary tree search and I need to use 1 and 0 to show the node is pass. I also need to print them out into the output file.

The second program is to decompress the output file create by first program and rebuild the origin file. I need to read the first part of the file to create the tree we needed to decompress. It is by read 1 and 0, control the node to move and when reach a character. Save it into the node and start form header of the tree again. When reading a space character, stop to create the tree building. I should can get the tree. The left thing is to read the 1 and 0 information. Every time and only when reaching the leaf node of the tree. Print the character saved in it and return to the begin of the tree and start the new search. I also made a EOF character at the end of this information. It is a 2. When to decompress program read a 2 character. It will stop and return.

I think the most difficult part in this program is to build the tree used to decompress and compress. My program should work but I don’t know the actual performance right now since my program still not finished. If the performance is not good I may change the way I do it.