Sinem Koçoğlu – 2020400339

Huzeyfe Bahadıroğlu – 2018400114

Firstly, registers are set to zero. Input is read char by char in read_input label. If the input is an operator, code jumps to corresponding operation's label to make the necessary calculation. If the input is space character, it pushes the last input to the stack in the label 'pushStack' If the input is enter character, value in the stack is popped to be printed after new line printed. Otherwise, code jumps to label 'numberInput' to calculate the value of hexadecimal number given in the input. CheckValue indicates that lastly read character is number and is used in order to prevent any unintentional push operation. If the number given in the input has more than one digit, when it reads the next digit, it shifts the previous character's bits to the right and adds the current digit.

When 'enter' character is read in the input, the code jumps to label 'output' to pop the result in the stack. To convert the result to hexadecimal form, 'myLoop' iterates four times because the output must be 4 digits. By dividing the result by 16, the digits are pushed to stack from right to left. Then, the digits in the stack are popped in 'myLoop2' four times to print the output. If the hexadecimal value of the digit in the stack is greater than 9, by addition and subtraction operations corresponding symbol is obtained in the label 'printLetter'. If not, conversion is done in the label 'print'.