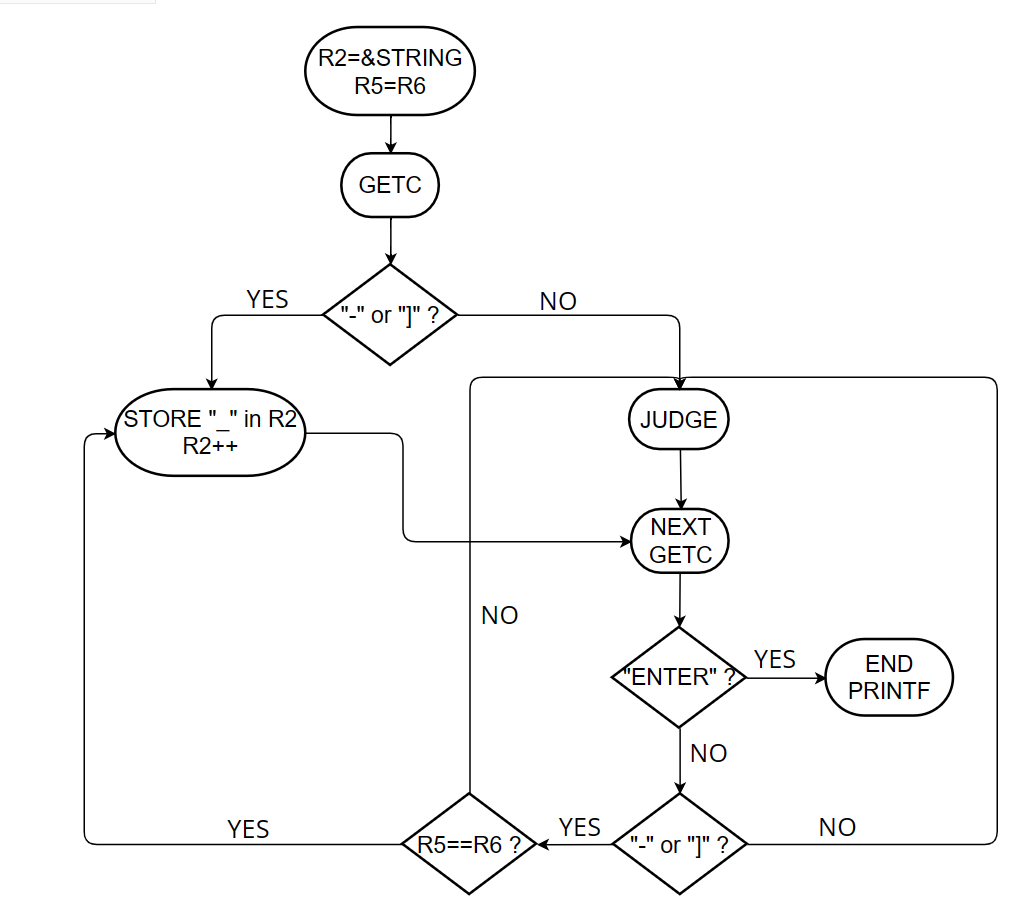
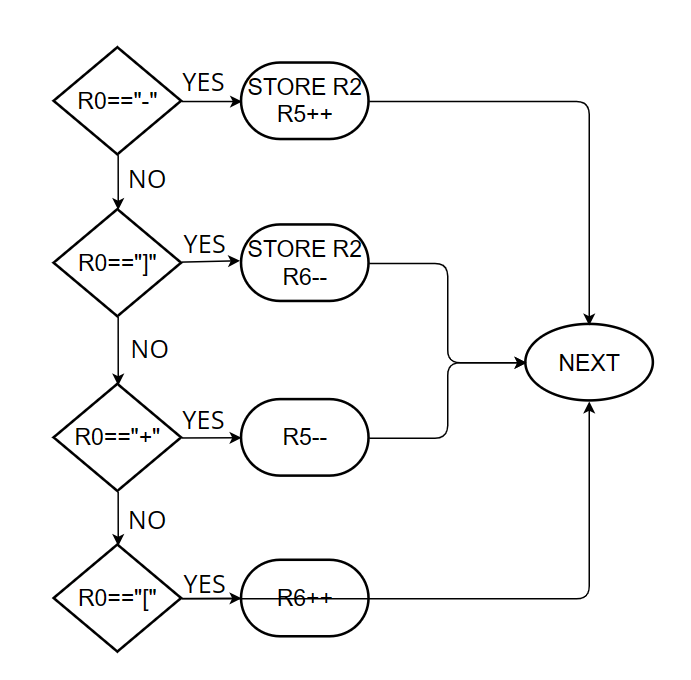
Lab3

1. Algorithm

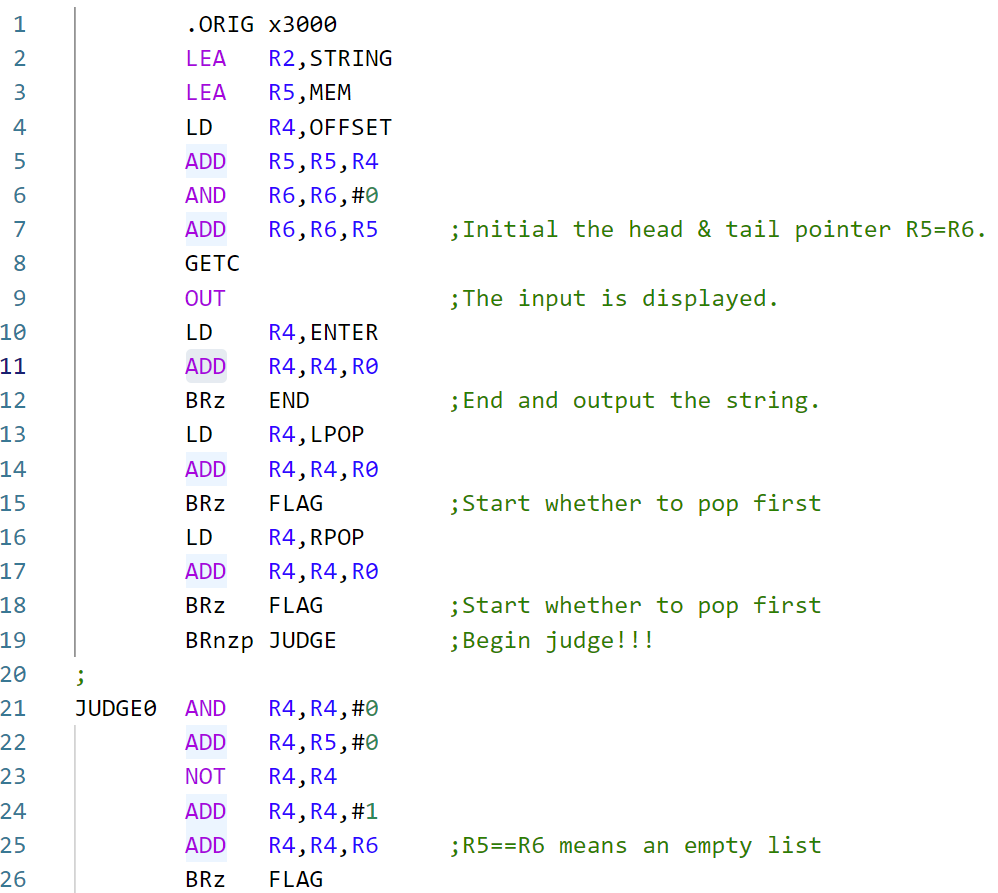


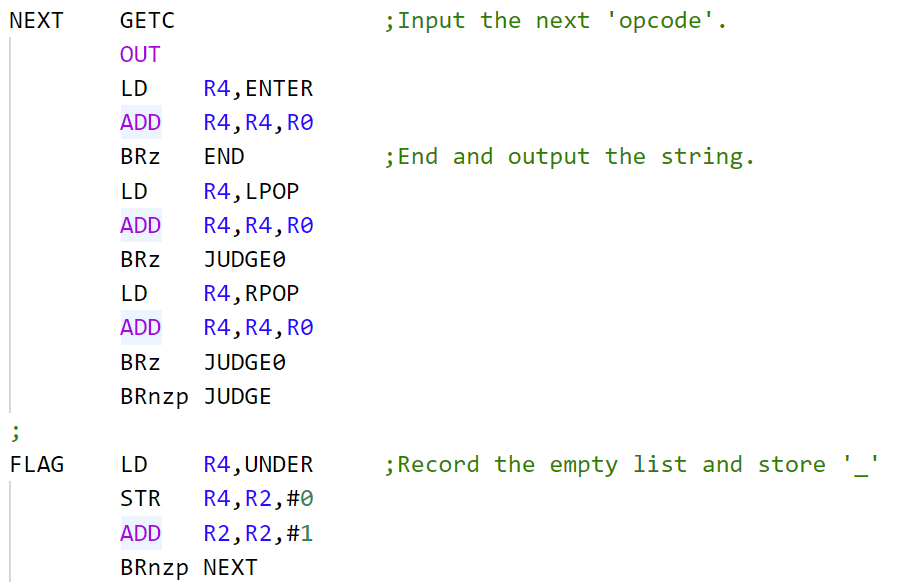
JUDGE：



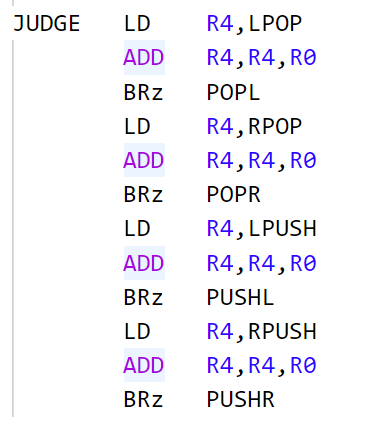
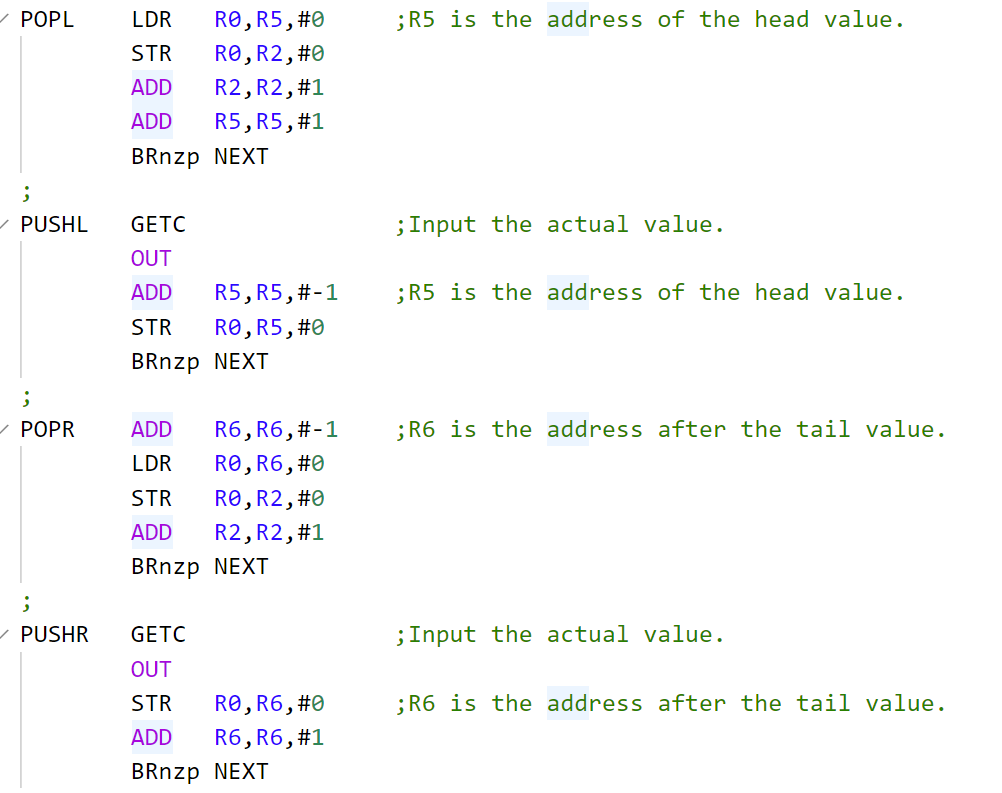
1. Essential parts of your code with sufficient comments

Assign R2 to the first address of STRING and R5=R6 to the address of the stack. Then start typing characters, if the first one is "-" or "] ", it means that the empty list has directly popped, so assign "\_" to the string, if not then proceed to the JUDGE phase. Then proceed to the next character input and judgment.





The JUDGE part is divided into four cases, where R5 points to the first address of the stack and R6 points to the last address of the stack. Move the address of R5 or R6 according to the entered character. If it is a pop character, the character is saved in the address of R2, and R2 is automatically increased.



1. Q&A
2. What is the effect of offset on line 88?

The string is required to be no more than 100, so I set the memory to 200, R5 and R6 point to the middle address, and I can push both left and right.

1. What’s the key parts of the program?

The JUDGE part is the key part. It contains four cases, where R5 points to the first address of the stack and R6 points to the last address of the stack. Move the address of R5 or R6 according to the entered character. If it is a pop character, the character is saved in the address of R2, and R2 is automatically increased.