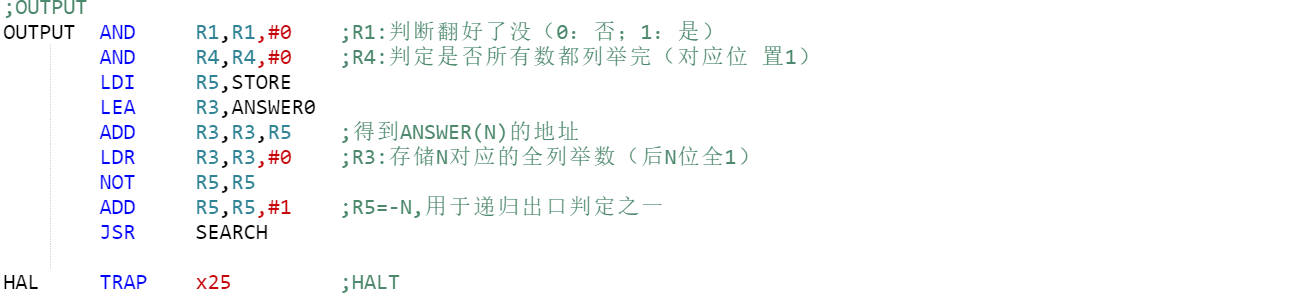
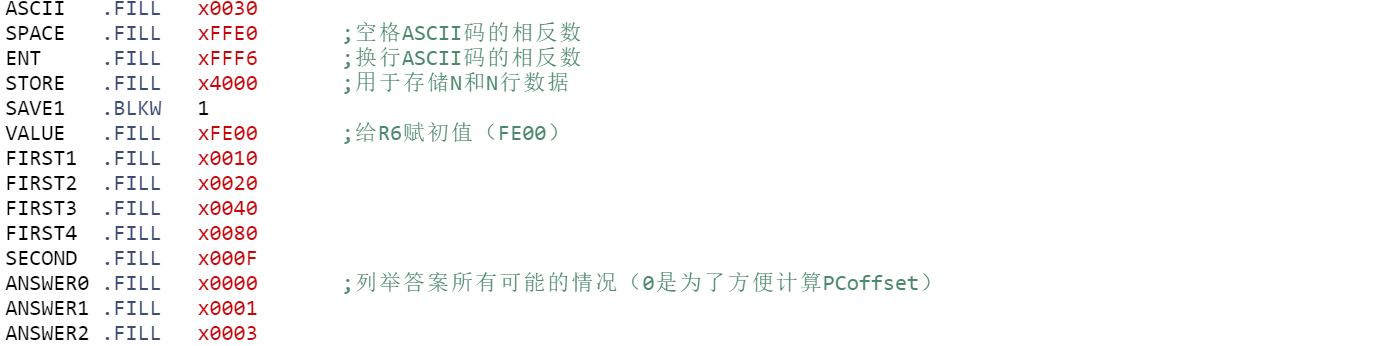
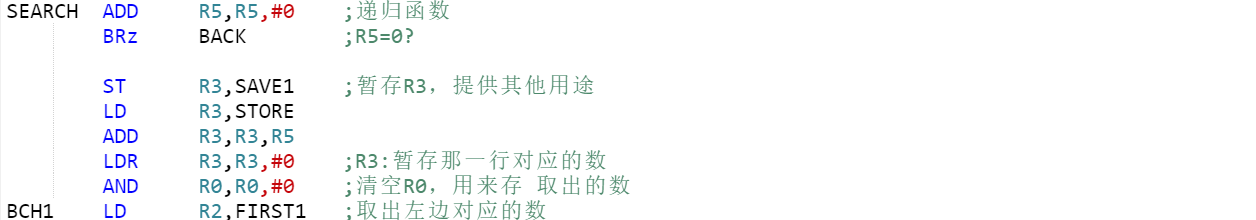
**Algorithm explanation**

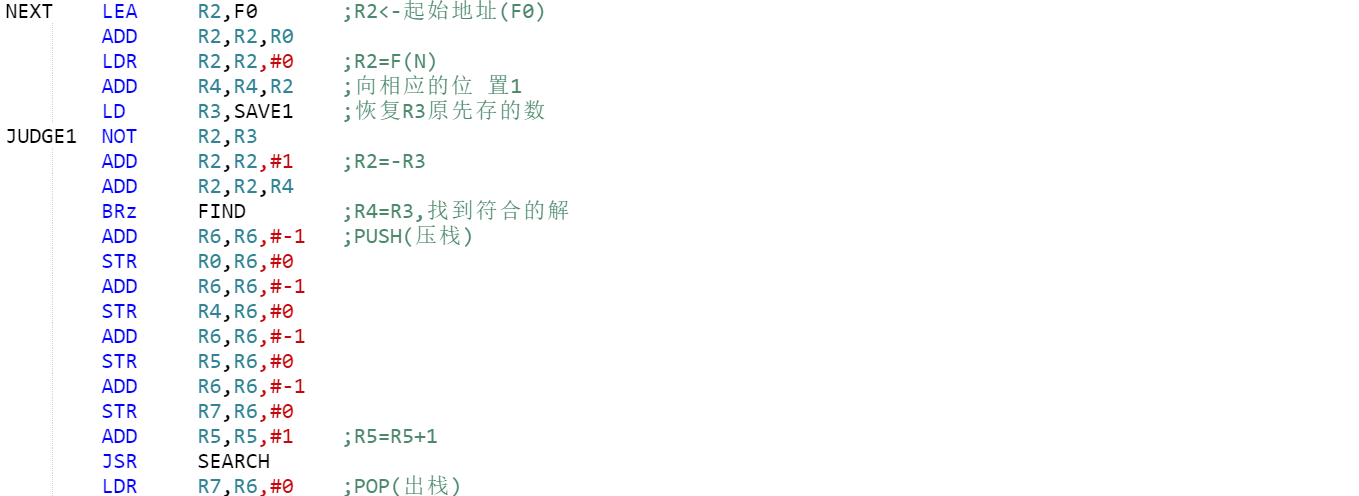
First, I store the integer N and N rows of data at a specific area one by one. Then I initialize R0, R1, R3,R4,R5, and JSR SEARCH(my recursive function) to find the answer. The R0 is used to store the left number or right number in each line, the R1 is used to judge if the answer is found(1:found,0:not), the R3 is used to store the ending flag(a 16-bit binary number whose last N bits are all ones), the R4 is used to store match marks and the R5 is used as recycling symbol. In my recursive function, I first judge if R5 is 0(if is, execute RET instruction), then I read the number on the left of the line and set 1 in R4’s corresponding bit and judge if R4 equals to R3(if yes, output the digit and return). If not, I push R0,R4,R5,R7 into stack and JSR SEARCH(next recursive function). Based on the value of the returned R1, I determine whether to read the right number and jump into the next recursive function. If both recursive functions return R1 with a value of 0, then I return to the previous recursive level and perform another operation(read the number on the right).

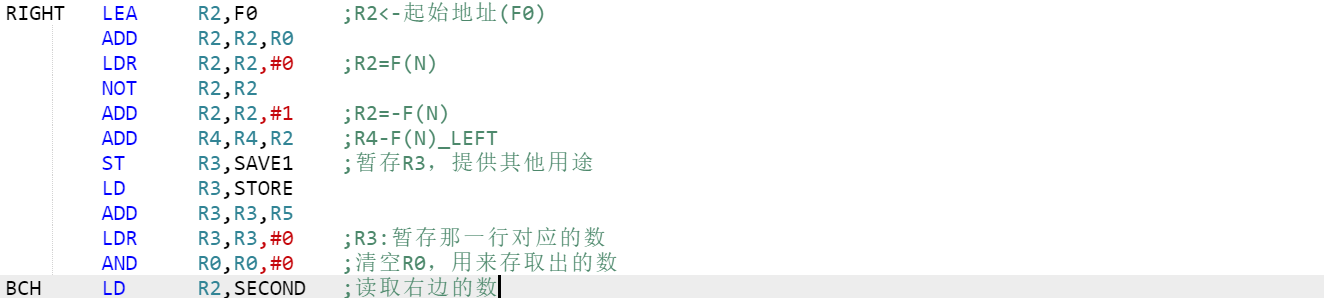
**Essential parts of your code with sufficient comments**

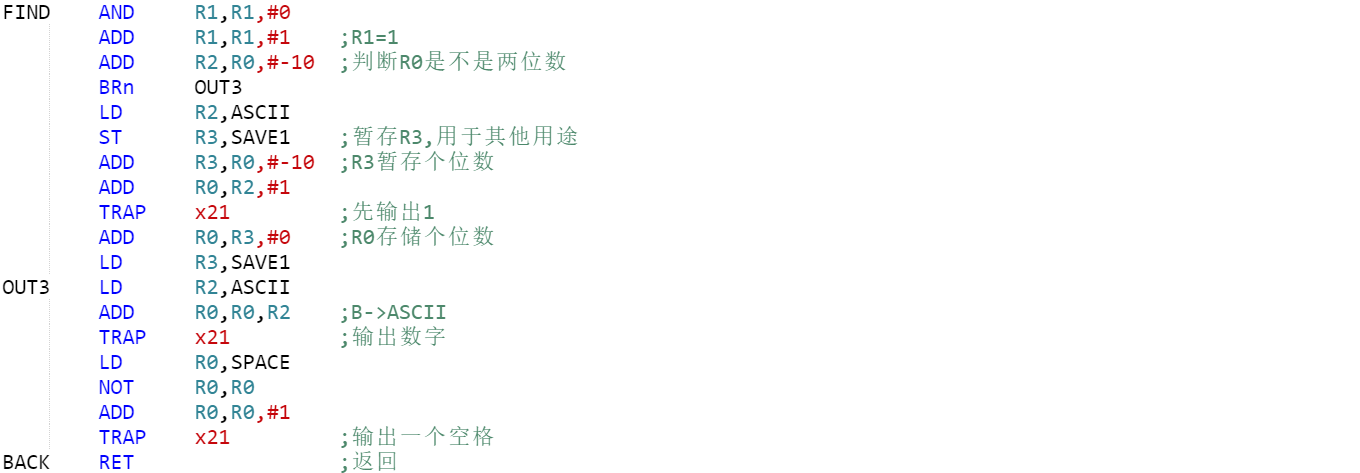












**Questions TA asked you and your answer in Check**

**TA:**What is your approach to this program（needn’t to mention the input part）?

**ME:**First, I initialize R0, R1, R3, R4, R5. The R0 is used to store the left number or right number in each line, the R1 is used to judge if the answer is found(1:found,0:not), the R3 is used to store the ending flag(a 16-bit binary number whose last N bits are all ones), the R4 is used to store match marks and the R5 is used as recycling symbol.In my recursive function, I first check if R5 equals to 0. If it is, return; if not continue. Then I get the left number and set 1 in R4’s corresponding bit and judge if R4 equals to R3. If it is, output the digit and return;if not, I push R0,R4,R5,R7 into stack and jump to the next recursive function. Based on the value of the returned R1, I determine whether to read the right number. If R1 didn’t equal to 1, I get the right number and set 1 in R4’s corresponding bit and judge if R4 equals to R3. If not, jump into next recursive function. If both recursive functions return R1 with a value of 0, then I return to the previous recursive level and perform another operation(read the number on the right).