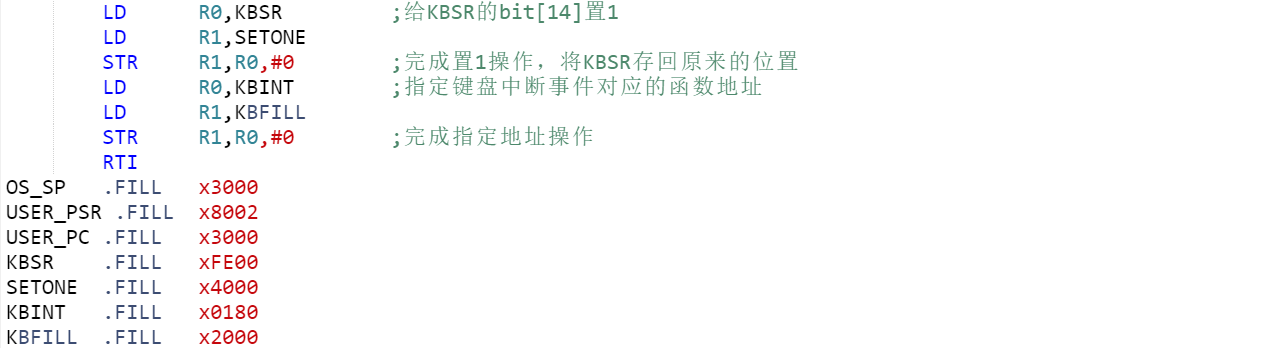
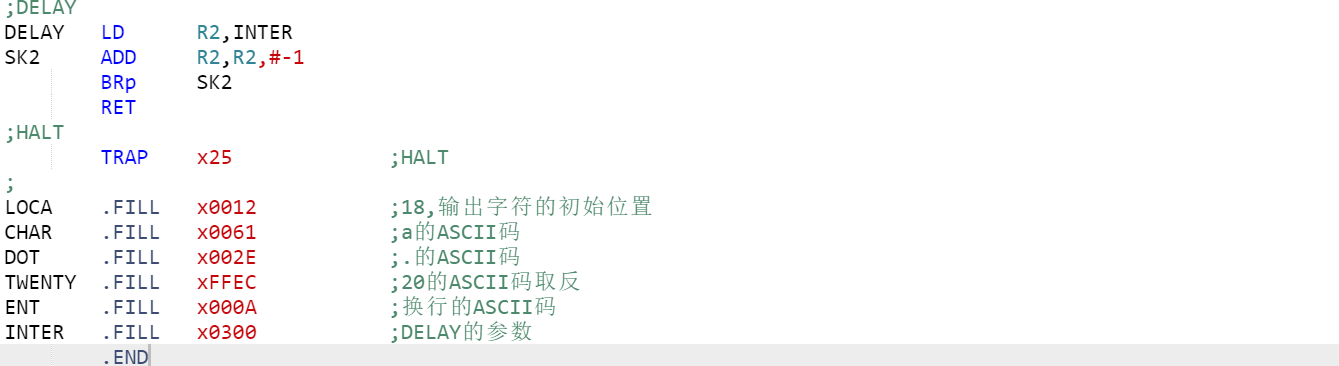
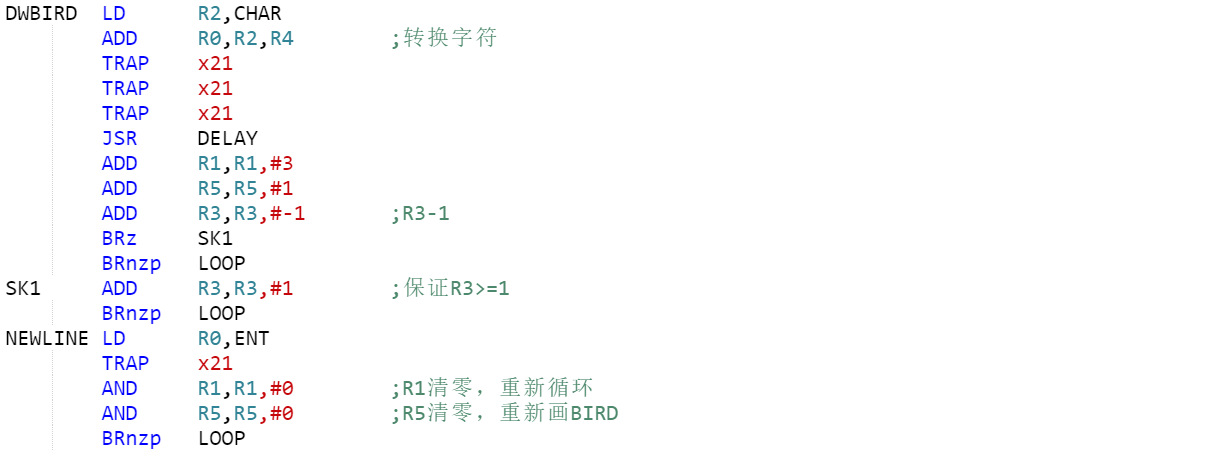
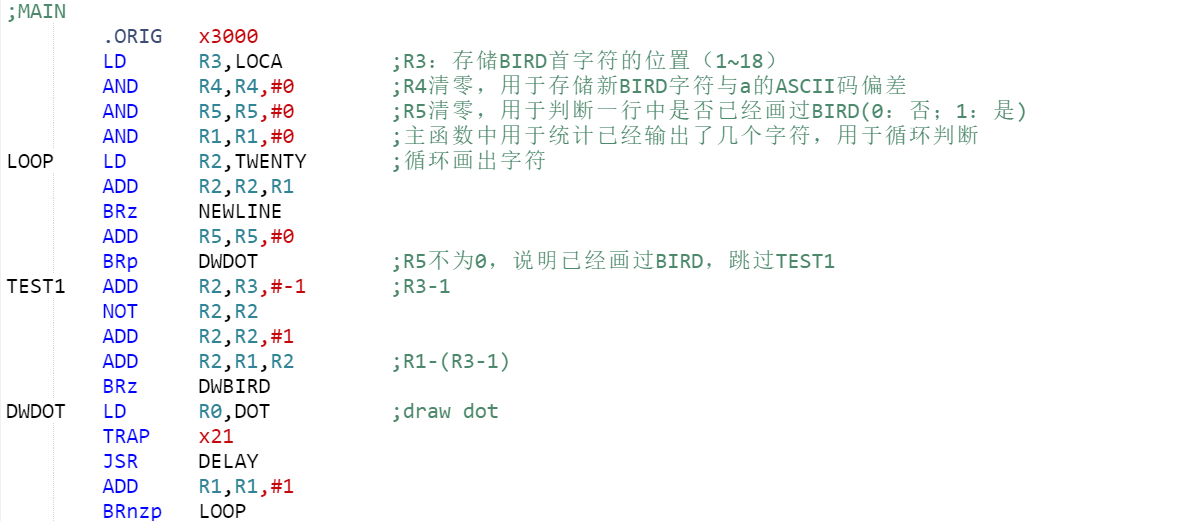
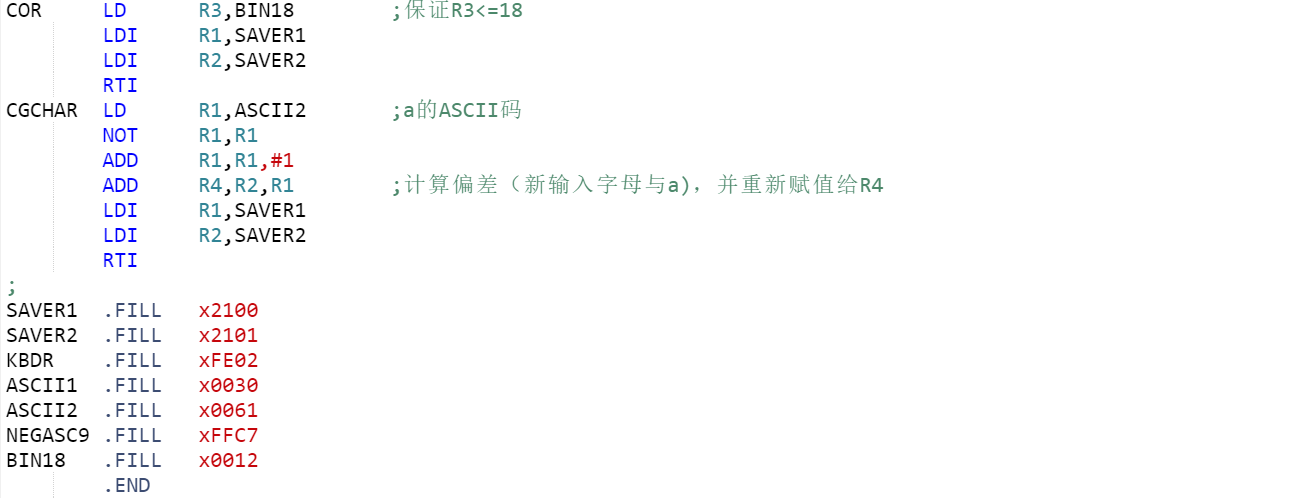
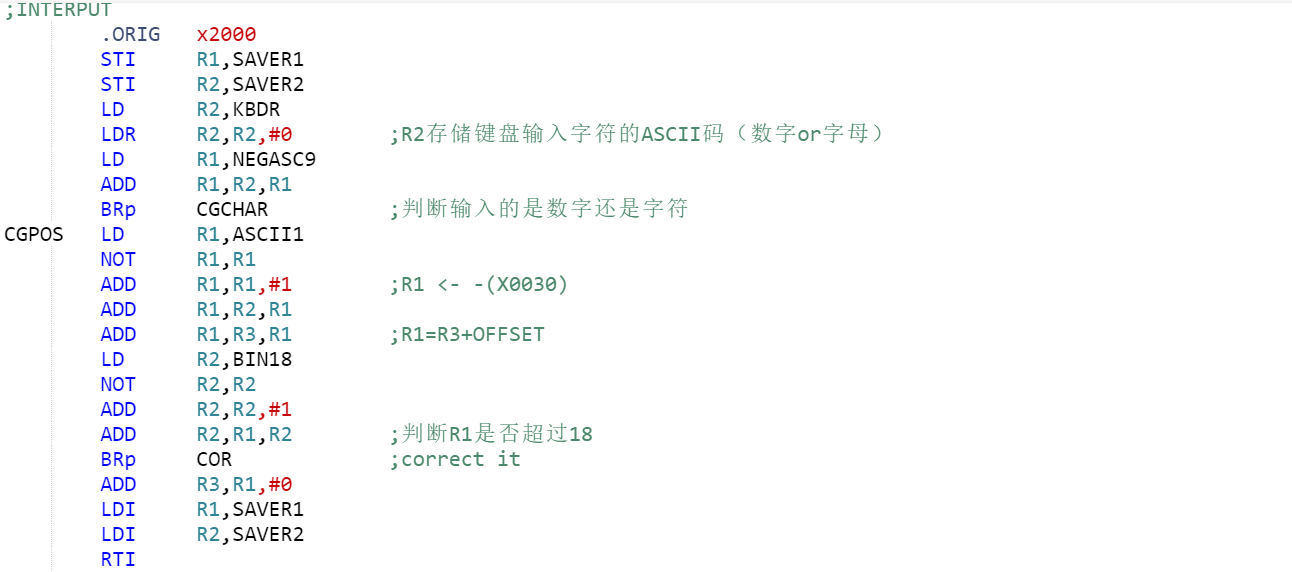
**Algorithm explanation**

First, at the system memory area(begin at x0200), I change the KBSR(set x4000 to it) and let x0180 store the starting address of the interrupt service routine(x2000) . Next, I start to write my main program. I create a loop and print 20 characters(including the dot character and the special character representing a bird) each line. In each loop, I decrement R3 to mimic the bird flying from the right to the left. I use R3 to store the position of the bird(the first representing character) and let R4 to store the offset of the ASCII code between the input letter(a~z) and the lowercase letter ‘a’. Every time I input a digit or a letter, the interrupt service routine is invoked. If I input a digit, the R3 will plus this number. If I input a letter, I will change the representing character by adjusting the value of R4.

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





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

1. **TA:**What’s the function of CGPOS and CGCHAR?

**ME:**Thefunction of CGPOS is changing the position of the bird(the first representing character) by adjusting the value of R3(during this process I will judge the value of R3 to guarantee that it will not surpass 18) .And the function of CGCHAR is changing the representing character by changing the value of R4(calculate the ASCII difference between the input character and the lowercase letter a) .

1. **TA:**Why do you use STI to store the value of R1 and R2 instead of ST when you invoke the interrupt service routine?

**ME:**Because I use the pseudo-ops (.FILL) rather than the pseudo-ops (.BLKW) to create a temporary storage area for R1 and R2. In that case, SAVER1 and SAVER2 represents the address of a memory location not a specific memory location.