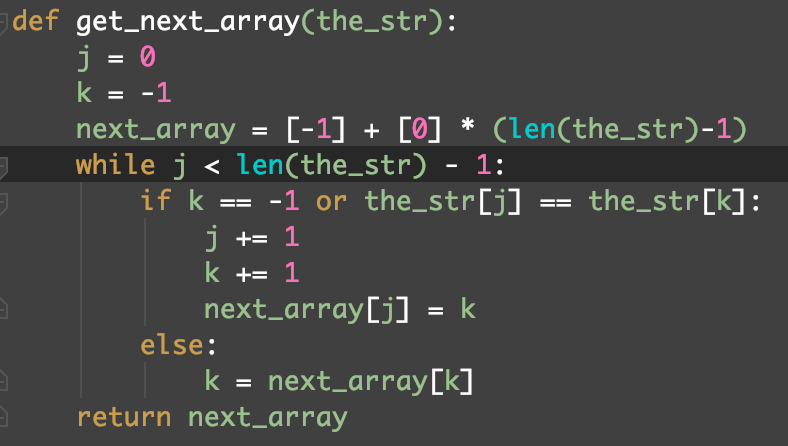
Lab 6 Report

1. **Given a string, calculate next function, print the result out.**
   1. Firstly, for a next\_array, the first two elements should be -1 and 0. So I set a list likes: [-1,0,…..,0,0]. I also set 2 int : k and j.

The number j means the location of current element waiting for get a value in the next\_array, while the k is used to calculate the that value. While string[j] != string[k], the k = next\_array[k] and join the judge again until string[j] == string[k], when next\_array[i]=k.

At the end we get a complete next\_array.

* 1. Code in Python:

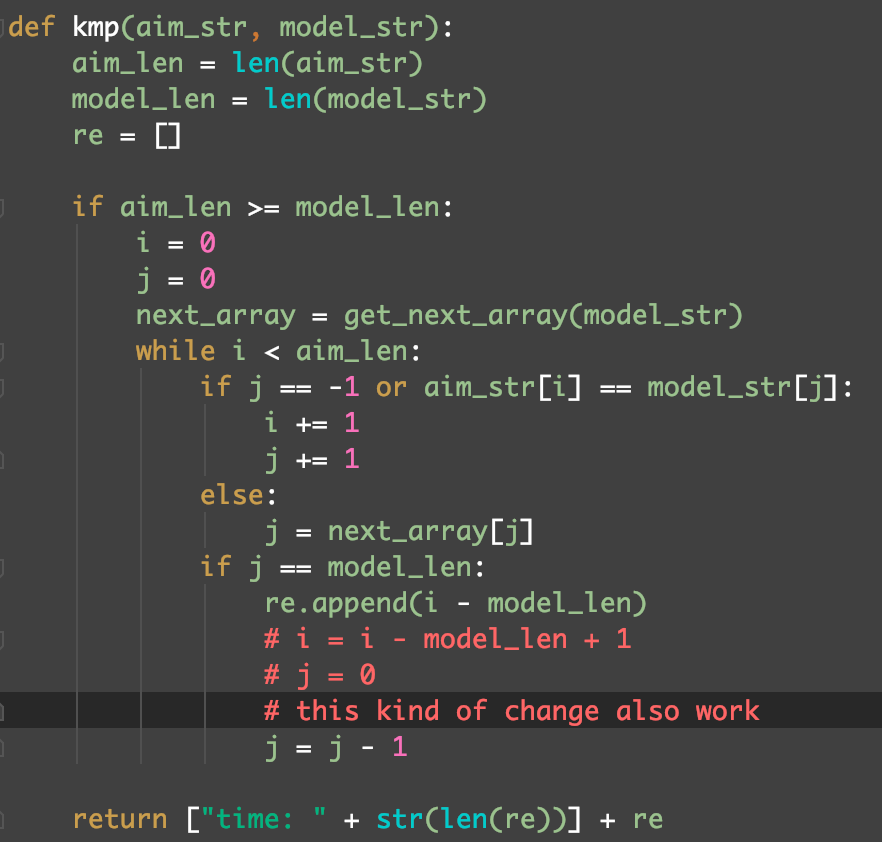


* 1. Test:





1. **Implement KMP algorithm**
   1. Since we have gotten the next\_array, while aim\_str[i] != model\_str[j], I just change the j to next\_array[j] until aim\_str[i] == model\_str[j]. While the j == length of model\_str, a copy of model\_str in aim\_str is found and change the j to (j-1).
   2. Code in Python:



* 1. Test:





1. **Given two strings S1, S2, find out how many time S2 appears in S1. Print out corresponding location when a search is successful.**

S1: ATCGCCGCGATCGTATTTAATCGATTGCATCG

S2: ATCG

Result: ATCGCCGCGATCGTATTTAATCGATTGCATCG

Test:

Output:

