Project Assignment 1: String Matching using Intel Assembly

Description of Assignment

The Knuth–Morris–Pratt string searching algorithm (or KMP algorithm) searches for occurrences of a string S within a text T. Unlike the brute-force method where every character of the string is compared with every character of the text, KMP makes use of a prefix table which can be constructed before running the actual search.

You will be given a text file of arbitrary length, containing the text T. Your C code should take the file name from the command line, should read the file, and store the text into the computer memory. Then your code should get the the search string S from the standard input.

The KMP prefix table construction and the KMP search algorithm should be written in Intel assembly as a subroutine. The input parameters will be the addresses of the text and the string, whereas the output will be the location index of the first character of the string at its first occurence in the text. (The first character of the text has an index value of 0, the second one has 1, etc.)
The following is the prototype of the kmp function:

```
int kmp(char *text, char *search_string);
```

The function(s) are not required to perform error checking on the parameters, but must follow C calling conventions and must fully conform to the given prototypes.

Your main program in C, will end by outputting:
1. the given text,
2. search string,
3. prefix table,
4. the result of your search
in a proper, readeable format.

To test your program, you can create text files using any text editor. However, during the lab session, you may be given specific files to test your code.

For further information about the KMP algorithm:
http://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm/
https://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm
https://www.youtube.com/watch?v=kBW6oPaVjq0

Submission Details

You are required to implement the KMP algorithm in Intel assembly. The function implementations must fully conform to the provided prototypes since they are expected to be linked to the main program implemented in C.

You are required to submit the C and assembly language source code file(s) through the Ninova system as a zip file. Each member of the group must make a submission, even though the submitted files may be the same for all group members.

Group members will be graded individually based on their performance in the lab session and the submitted group project. The students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system. Any form of cheating or plagiarism will not be tolerated. The submitted work should be the product

of the group itself; collaboration or code sharing between different groups will be accepted as plagiarism. This also includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.