**Pattern Matching Program**

This is a C++ program designed to search for patterns within text files and identify their positions. It reads multiple pairs of text and pattern files, searches for the specified pattern within the text, and outputs the findings to separate output files.

**Usage**

1. Compile the program using a C++ compiler, for example:

Copy code

g++ code.cpp -o pattern\_matcher

1. Run the compiled executable:

bashCopy code

./pattern\_matcher

1. Follow the on-screen prompts to enter the number of test cases and provide corresponding text and pattern files.
2. The program will analyze each text and pattern pair and generate an output file for each test case, indicating the positions where the pattern is found within the text.

**Program Structure**

**main() Function**

The main function prompts the user for the number of test cases and processes each case by attempting to open the corresponding text and pattern files. If either file is not found, an error message is displayed, and the program moves on to the next test case. Otherwise, it calls the **find\_Pattern()** function to search for the pattern within the text.

**find\_Pattern() Function**

This function is responsible for reading the contents of the text file, searching for the specified pattern within the text, and writing the results to an output file. It reads the pattern from the pattern file and then iterates through the text while invoking the **match\_Pattern()** function to check for pattern matches.

**match\_Pattern() Function**

The pattern matching logic is implemented in this function. It uses recursive backtracking to compare the characters of the text and pattern. The function considers various cases, including matching individual characters, handling special characters like '.', '^', '$', and '?', and tracking line and position information.

**Input Files**

For each test case, the program expects two files: a text file (named as "textX.txt", where X is the test case number) and a pattern file (named as "patternX.txt"). Both files should be placed in the same directory as the program executable. The text file contains the content in which the pattern is to be searched, while the pattern file contains the pattern to be matched.

**Output Files**

For each test case, the program generates an output file (named as "output X.txt", where X is the test case number) that records the positions where the pattern is found within the text. Each line of the output file corresponds to a match and includes the line number and position where the pattern is located.

**Notes**

* This program assumes that the text and pattern files are plain text files without any formatting or special encoding.
* The pattern matching algorithm supports basic regular expression-like patterns, including '.' (matches any character), '^' (matches the beginning of a line), '$' (matches the end of a line), and '?' (makes the preceding character optional).