Name Zarak khan

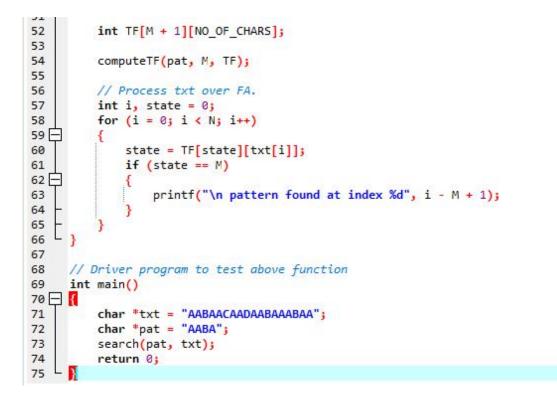
ld 13794

Dept BS(CS)

C++ Program to Perform Finite State Automaton based Search

```
Naik.cpp
 1 #include<stdio.h>
2 #include<string.h>
3 #define NO_OF_CHARS 256
 4
       int getNextState(char *pat, int M, int state, int x)
  5
 6₽{
           // If the character c is same as next character in pattern,
// then simply increment state
if (state < M && x == pat[state])</pre>
 7
 9
10
11
              return state + 1;
12
13
           int ns, i; // ns stores the result which is next state
14
15
          // ns finally contains the longest prefix which is also suffix
// in "pat[0..state-1]c"
16
           // Start from the largest possible value and stop when you find
// a prefix which is also suffix
for (ns = state; ns > 0; ns--)
17
18
19
20 🖵
           {
21 T
22 E
               if (pat[ns - 1] == x)
23
24 E
                    for (i = 0; i < ns - 1; i++)</pre>
                    {
25
                        if (pat[i] != pat[state - ns + 1 + i])
26
                            break;
                               ......
                   }
                  if (i == ns - 1)
                         return ns;
      return 0;
}
/* This function builds the TF table which represents Finite Automata for a
 given pattern */
void computeTF(char *pat, int M, int TF[][NO_OF_CHARS])
{
      int state, x;
      for (state = 0; state <= M; ++state)</pre>
            for (x = 0; x < NO_OF_CHARS; ++x)</pre>
                 TF[state][x] = getNextState(pat, M, state, x);
}
/* Prints all occurrences of pat in txt */
void search(char *pat, char *txt)
{
      int M = strlen(pat);
      int N = strlen(txt);
```

1 Compile Log 📈 Debug 🗖 Find Results 🏙 Close



Output:

