Name: Ashmit Thawait

Roll No: 102203790

Group: 2CO17

Lab Assignment 6

Q1. Rabin-Karp string matching algorithm

```cpp
#include <bits/stdc++.h>
using namespace std;
#define d 256

void search(char pat[], char txt[], int q)
{
    int M = strlen(pat);
    int N = strlen(txt);
    int i, j;
    int p = 0;
    int t = 0;
    int h = 1;

    for (i = 0; i < M - 1; i++)
        h = (h * d) % q;

    for (i = 0; i < M; i++) {
        p = (d * p + pat[i]) % q;
        t = (d * t + txt[i]) % q;
    }

    for (i = 0; i <= N - M; i++) {

        if (p == t) {
            for (j = 0; j < M; j++) {
                if (txt[i + j] != pat[j]) {
                    break;
                }
            }

            if (j == M)
                cout << "Pattern found at index " << i
                    << endl;
        }
```

```
36              if (i < N - M) {
37                  t = (d * (t - txt[i] * h) + txt[i + M]) % q;
38                  if (t < 0)
39                      t = (t + q);
40              }
41          }
42      }
43
44      int main()
45      {
46          char txt[] = "DESIGN AND ANALYSIS OF ALGORITHMS";
47          char pat[] = "ALGO";
48          int q = INT_MAX;
49
50          search(pat, txt, q);
51          return 0;
52      }
```

```
PS D:\DAA Assignments\Assignment 6> g++ .\Rabin-Karp.cpp
PS D:\DAA Assignments\Assignment 6> ./a.exe
Pattern found at index 23
PS D:\DAA Assignments\Assignment 6>
```

Q2. Knuth-Morris-Prath (KMP) algorithm

```cpp
#include <bits/stdc++.h>

void computeLPSArray(char* pat, int M, int* lps);

void KMPSearch(char* pat, char* txt)
{
    int M = strlen(pat);
    int N = strlen(txt);
    int lps[M];

    computeLPSArray(pat, M, lps);

    int i = 0;
    int j = 0;
    while ((N - i) >= (M - j)) {
        if (pat[j] == txt[i]) {
            j++;
            i++;
        }

        if (j == M) {
            printf("Found pattern at index %d ", i - j);
            j = lps[j - 1];
        }

        else if (i < N && pat[j] != txt[i]) {
            if (j != 0)
                j = lps[j - 1];
            else
                i = i + 1;
        }
    }
}
```

```
35  void computeLPSArray(char* pat, int M, int* lps)
36  {
37      int len = 0;
38      lps[0] = 0;
39
40      int i = 1;
41      while (i < M) {
42          if (pat[i] == pat[len]) {
43              len++;
44              lps[i] = len;
45              i++;
46          }
47          else
48          {
49              if (len != 0) {
50                  len = lps[len - 1];
51              }
52              else
53              {
54                  lps[i] = 0;
55                  i++;
56              }
57          }
58      }
59  }
60
61  int main()
62  {
63      char txt[] = "ABABDABACDABABCABAB";
64      char pat[] = "ABABCABAB";
65      KMPSearch(pat, txt);
66      return 0;
67  }
68
```

```
PS D:\DAA Assignments\Assignment 6> g++ kmp.cpp
PS D:\DAA Assignments\Assignment 6> ./a.exe
Found pattern at index 10
PS D:\DAA Assignments\Assignment 6>
```