

```
/*
```

In this problem you are given two non-empty strings A and B, both contain lower case English alphabets. You have to find the number of times B occurs as a substring of A.

Input

```
1
abcabcabcabc
abc
```

output:

Case 1: 4

```
*/
```

```
LLI failure_table[MX];
```

```
void failure_function(string pat)
```

```
{
    LLI M=pat.size();
    LLI len = 0;
    LLI i;

    failure_table[0]=0;
    i = 1;

    while(i < M)
    {
        if(pat[i] == pat[len])
        {
            len++;
            failure_table[i] = len;
            i++;
        }
        else
        {
            if( len != 0 )
            {
                len = failure_table[len-1];
            }
            else
            {
                failure_table[i] = 0;
                i++;
            }
        }
    }
}
```

```

int KMP(string pat, string txt)
{
    LLI M = pat.size();
    LLI N = txt.size();
    LLI j = 0, c=0;

    // building the failure table
    failure_function(pat);

    LLI i = 0;
    while(i < N)
    {
        if(pat[j] == txt[i])
        {
            j++;
            i++;
        }

        if (j == M)
        {
            c++;
            j = failure_table[j-1];
        }

        else if(pat[j] != txt[i])
        {
            if(j != 0)
                j = failure_table[j-1];
            else
                i = i+1;
        }
    }
    return c;
}

int main()
{
    string txt, pat;
    int T;
    cin >> T;
    getchar();
    for(int t=1; t<=T; t++)
    {
        getline(cin, txt);
        getline(cin, pat);
        int c=KMP(pat, txt);
        txt.clear();
        pat.clear();
        printf("Case %d: %d\n", t, c);
    }
    return 0;
}

```