//String Matching

//Knuth Morris Pratt

//Complexity : O(String + Token)

char P[2000010], T[1000010];

int P\_Size, T\_Size, Table[1000010];

//T is the string that we need to find

//S is the string in which we have to find

//Table contains the prefix table

//builds the prefix table

void PrefixTable()

{

register int i = 0, j = -1;

Table[0] = -1;

while(i < T\_Size) {

while(j >= 0 && T[i] != T[j])

j = Table[j];

i++, j++;

Table[i] = j;

}

}

int KmpSearch()

{

register int i = 0, j = 0, cnt = 0;

while(i < P\_Size) {

while(j >= 0 && P[i] != T[j])

j = Table[j];

i++, j++;

if(j == T\_Size) {

cnt++;

//the match found in i-j, if i-j = 0, then the string is fully matched

//this happens when the string is equal in length of the token

//printf("%d'th Match found at %d\n", cnt, i-j); //the leftmost index

j = Table[j];

}

}

//j contains the first segment index that is matched in token

return cnt;

}