```
##################################################
################## dijkstra.cpp ##################
##################################################

#include<bits/stdc++.h>
using namespace std;

#define MOD 1000000007
//#define MOD 998244353
#define INF 1000000010
#define EPS 1e-9

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define lp(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"
#define N 100000
#define F first
#define S second

typedef pair<int,int> PII;
typedef pair<int,string> PIS;
typedef pair<string,int> PSI;

struct Node{
  vector<PII> edges;//first=cost,second=index

  bool visit;
  int cost;
};

Node mp[N];

void dijkstra(int st,int n){
  rep(i,n){
    mp[i].visit=false;
    mp[i].cost= -1;
  }
  priority_queue<PII>  q;
  mp[st].cost=0;
  rep(i,mp[st].edges.size() ){
    q.push({-mp[st].edges[i].F,mp[st].edges[i].S});
  }
  mp[st].visit=true;
  while(!q.empty() ){
    PII p=q.top();
    q.pop();
    if(! mp[p.S].visit){
      mp[p.S].cost= -p.F;
      mp[p.S].visit=true;
      rep(i,mp[p.S].edges.size() ){
        q.push(make_pair(-mp[p.S].edges[i].F-mp[p.S].cost,mp[p.S].edges[i].S));
      }
    }
  }
}




##################################################
################ prime_factor.cpp ################
##################################################

#include<bits/stdc++.h>
using namespace std;

```

```
#define int long long
#define endl "\n"

map<int,int> prime_factor(int n){
  map<int,int> mp;
  for(int i=2;i*i<=n;i+=2){
    while(n%i==0){
      mp[i]++;
      n/=i;
    }
    if(i==2) i--;
  }
  if(n!=1) mp[n]=1;
  return mp;
}




##################################################
################## template.cpp ##################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
//#define MOD 998244353
#define INF 1000000010
#define EPS 1e-9
#define F first
#define S second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define lp(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"

typedef pair<int,int> PII;
typedef pair<int,string> PIS;
typedef pair<string,int> PSI;


signed main(){
  cin.tie(0);
  ios::sync_with_stdio(false);


  return 0;
}

##################################################
################## divisor.cpp ###################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
#define INF 1000000010
#define EPS 1e-9
#define fst first
#define scd second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define lp(i,n) repi(i,0,n)
```

```
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"

vector<int> divisor(int n){
  vector<int> v;
  for(int i=1;i*i<=n;i++){
    if(n%i==0){
      v.push_back(i);
      if(i*i!=n) v.push_back(n/i);
    }
  }
  sort(v.begin(),v.end() );
  return v;
}




##################################################
################# draftcode.cpp ##################
##################################################

#include<bits/stdc++.h>
using namespace std;


typedef string::const_iterator State;
class ParseError {};

int number(State &begin){
  int num=0;
  while(isdigit(*begin)){
    num*=10;
    num+= *begin - '0';
    begin++;
  }
  return num;
}




##################################################
################# combinaton.cpp #################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
#define BIG 1000000010
#define EPS 1e-9
#define fst first
#define scd second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long

const int MAX=500000;

int fac[MAX],finv[MAX],inv[MAX];

void t(){
  fac[0]=fac[1]=1;
  finv[0]=finv[1]=1;
  inv[1]=1;
  repi(i,2,MAX){
    fac[i]=fac[i-1]*i%MOD;
    inv[i]=MOD-inv[MOD%i]*(MOD/i)%MOD;
    finv[i]=finv[i-1]*inv[i]%MOD;
  }
}

int calc(int n,int k){
  if(n<k) return 0;
  if(n<0 || k<0) return 0;
  return fac[n]*(finv[k]*finv[n-k]%MOD)%MOD;
}



signed main(){
  int n,m;
  cin>>n>>m;
  t();
  cout<<calc(n,m)<<endl;
  return 0;
}


##################################################
################# zalgorithm.cpp #################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
#define BIG 1000000010
#define EPS 1e-9
#define fst first
#define scd second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"

int A[300000],B[300000];


//Aã\201«å\205¥ã\202\213ã\201®ã\201¯ã\200\201Sã\201®iæ\226\207å-\227ç\233®ã\201\213ã\202\211ä½\225æ\226\207å-\227ã\201\214tã\201®æ\216¥é -è¾\236ã\201¨ä¸\200è\207´ã\201\231ã\202\213ã\201\213

void zalgorithm(string S,string t){
  int i = 0, j = 0;
  while (i+j < S.size() && j < t.size() && t[j] == S[i+j]) ++j;
  A[i] = j;
  if (j != 0){
    int k = 1;
    while (i+k < S.size() && k+A[k] < j) A[i+k] = A[k], ++k;
  }
  i=1;j=0;
  while (i < S.size()) {
    while (i+j < S.size() && j < t.size() && t[j] == S[i+j]) ++j;
    A[i] = j;
    if (j == 0) { ++i; continue;}
    int k = 1;
    while (i+k < S.size() && k+A[k] < j) A[i+k] = A[k], ++k;
    i += k; j -= k;
  }
}

void rzalgorithm(string S,string t){
  int i = 0, j = 0;
  reverse(t.begin(),t.end() );
  reverse(S.begin(),S.end() );
  // cout<<t<<endl;
  while (i+j < S.size() && j < t.size() && t[j] == S[i+j]) ++j;
  B[i] = j;
  if (j != 0){
    int k = 1;
    while (i+k < S.size() && k+B[k] < j) B[i+k] = B[k], ++k;
  }
  i=1;j=0;
  while (i < S.size()) {
    while (i+j < S.size() && j < t.size() && t[j] == S[i+j]) ++j;
    B[i] = j;
    if (j == 0) { ++i; continue;}
    int k = 1;
    while (i+k < S.size() && k+B[k] < j) B[i+k] = B[k], ++k;
    i += k; j -= k;
  }
}


##################################################
################### prime.cpp ####################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
#define BIG 1000000010
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
typedef long long int ll;

#define N 100000

int prime[N];

void eratosthenes(){
  rep(i,N) prime[i]=1;
  prime[0]=prime[1]=0;
  rep(i,n){
    if(prime[i]){
      for(int j=i+i;j<N;j+=i) prime[j]=0;
    }
  }
}

bool primeNumber(int n){
  if(n < 2) return false;
  else{
    for(int i = 2; i * i <= n; i++){
      if(n % i == 0) return false;
    }
    return true;
  }
}


##################################################
################### power.cpp ####################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
//#define MOD 998244353
#define INF 1000000010
#define EPS 1e-9
```

```
#define F first
#define S second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define lp(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"

typedef pair<int,int> PII;
typedef pair<int,string> PIS;
typedef pair<string,int> PSI;

int power(int n,int m){
  int now=2;
  int num=1;
  while(n!=0){
    //cout<<n<<" "<<m<<endl;
    if(n%2!=0){
      num=(num*m)%MOD;

    }
    n/=2;
    m=(m*m)%MOD;
  }
  return num;
}


##################################################
################ rollinghash.cpp #################
##################################################

#include<bits/stdc++.h>
using namespace std;
using Int = long long;
template<typename T1,typename T2> inline void chmin(T1 &a,T2 b){if(a>b) a=b;}
template<typename T1,typename T2> inline void chmax(T1 &a,T2 b){if(a<b) a=b;}
//BEGIN CUT HERE
template<typename T,T MOD,T B>
struct RollingHash{
  vector<T> hash,p;
  RollingHash(){}
  RollingHash(const string &s){
    int n=s.size();
    hash.assign(n+1,0);
    p.assign(n+1,1);
    for(int i=0;i<n;i++){
      hash[i+1]=(hash[i]*B+s[i])%MOD;
      p[i+1]=p[i]*B%MOD;
    }
  }
  //S[l, r)
  T find(int l,int r){
    T res=hash[r]+MOD-hash[l]*p[r-l]%MOD;
    return res>=MOD?res-MOD:res;
  }
};
//END CUT HERE
//INSERT ABOVE HERE
signed main(){
  cin.tie(0);
  ios::sync_with_stdio(0);
  string t,p;
  cin>>t>>p;
  using ll = long long;
  const ll MOD = 1e9+7;
  const ll B = 1777771;
  using RH = RollingHash<ll, MOD, B>;
  RH rt(t),rp(p);
  for(int i=0;i<(int)t.size()-(int)p.size()+1;i++){
    if(rt.find(i,i+p.size())==rp.find(0,p.size())) cout<<i<<"\n";
  }
  cout<<flush;
  return 0;
}


##################################################
############### unionfindtree.cpp ################
##################################################



struct UnionFind{
  vector<int> data;

  UnionFind(int N){
    data.assign(N,-1);
  }

 bool unite(int x, int y) {
    x = find(x), y = find(y);
    if(x == y) return (false);
    if(data[x] > data[y]) swap(x, y);
    data[x] += data[y];
    data[y] = x;
    return (true);
  }

  int find(int k) {
    if(data[k] < 0) return (k);
    return (data[k] = find(data[k]));
  }

  int size(int k) {
    return (-data[find(k)]);
  }
};


##################################################
############## ivu_Bellmanford.cpp ###############
##################################################

#include <bits/stdc++.h>
using namespace std;
#define FOR(i,l,r) for(int i=(l); i<(r); i++)
#define REP(i,n) FOR(i,0,n)
#define endl "\n"
#define debug(x) cout<<x<<endl;
typedef long long ll;
static const int INF = 1e9+7;

struct Edge {
  int to,cost;  //to æ\216¥ç¶\232å\205\210ã\200\200, cost é\207\215ã\201¿
  Edge(int to, int cost) : to(to), cost(cost) {}  // ã\202³ã\203³ã\202¹ã\203\210ã\203©ã\202¯ã\202¿
};

typedef vector<vector<Edge> > Edge_List;
Edge_List graph;

vector<int> dist; //æ\234\200ç\237-è•\235é\233¢

// æ\210»ã\202\212å\200¤ã\201\214trueã\201ªã\202\211è² ã\201®é\226\211è•¯ã\202\222å\220«ã\202\200
bool bellman_ford(int n, int s) {
    //nã\201¯é \202ç\202¹æ\225°,sã\201¯å§\213ã\201¾ã\202\212ã\201®é \202ç\202¹

    dist = vector<int>(n,INF);
    dist[s] = 0; // é\226\213å§\213ç\202¹ã\201®è•\235é\233¢ã\201¯0
    for (int i=0; i<n; i++) {
        for (int v=0; v<n; v++) {
            for (int k=0; k<graph[v].size(); k++) {
                Edge e = graph[v][k];
                if (dist[v] != INF && dist[e.to] > dist[v] + e.cost) {
                    dist[e.to] = dist[v] + e.cost;
                    if (i == n - 1) return true; // nå\233\236ç\233®ã\201«ã\202\202æ\233´æ\226°ã\201\214ã\201\202ã\202\213ã\201ªã\202\211ã\201°è² ã\201®é\226\211è•¯ã\201\214å-\230å\234¨ã\201\227ã\201¦ã\201\204ã\202\213
                }
            }
        }
    }
  return false;
}


/*int main() {
  int n, m;
  cin >> n >> m;
  graph = Edge_List(n);

  for (int i=0; i<m; i++) {
    int from,to,cost;
    cin >>from>>to>>cost;
    graph[from].push_back(Edge(to, cost));
  }

  bellman_ford(n,0);

  for (int i=1; i<n; i++) {
    if (dist[i] != INF)
      cout<<"0ã\201\213ã\202\211"<< i <<"ã\201¸ã\201®ã\202³ã\202¹ã\203\210ã\201¯: "<<dist[i]<<endl;
  }

}
*/


##################################################
################ suffix_array.cpp ################
##################################################

#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000007
#define INF 1000000010
#define EPS 1e-9
#define fst first
#define scd second

#define debug(x) cout<<x<<endl;
#define repi(i,x,n) for(int i=x;i<n;i++)
#define rep(i,n) repi(i,0,n)
#define lp(i,n) repi(i,0,n)
#define repn(i,n) for(int i=n;i>=0;i--)
#define int long long
#define endl "\n"


struct SuffixArray {
  vector< int > SA;
  const string s;

  SuffixArray(const string &str) : s(str) {
    SA.resize(s.size());
    iota(begin(SA), end(SA), 0);
    sort(begin(SA), end(SA), [&](int a, int b) {
      return s[a] == s[b] ? a > b : s[a] < s[b];
    });
    vector< int > classes(s.size()), c(s.begin(), s.end()), cnt(s.size());
    for(int len = 1; len < s.size(); len <<= 1) {
      for(int i = 0; i < s.size(); i++) {
        if(i > 0 && c[SA[i - 1]] == c[SA[i]] && SA[i - 1] + len < s.size() && c[SA[i - 1] + len / 2] == c[SA[i] + len / 2]) {
          classes[SA[i]] = classes[SA[i - 1]];
        } else {
          classes[SA[i]] = i;
        }
      }
      iota(begin(cnt), end(cnt), 0);
      copy(begin(SA), end(SA), begin(c));
      for(int i = 0; i < s.size(); i++) {
        int s1 = c[i] - len;
        if(s1 >= 0) SA[cnt[classes[s1]]++] = s1;
      }
      classes.swap(c);
    }
  }

  int operator[](int k) const {
    return SA[k];
  }

  size_t size() const {
    return s.size();
  }

  bool lt_substr(const string &t, int si = 0, int ti = 0) {
    int sn = (int) s.size(), tn = (int) t.size();
    while(si < sn && ti < tn) {
      if(s[si] < t[ti]) return true;
      if(s[si] > t[ti]) return false;
      ++si, ++ti;
    }
    return si >= sn && ti < tn;
  }

  int lower_bound(const string &t) {
    int low = -1, high = (int) SA.size();
    while(high - low > 1) {
      int mid = (low + high) / 2;
      if(lt_substr(t, SA[mid])) low = mid;
      else high = mid;
    }
    return high;
  }

  pair< int, int > lower_upper_bound(string &t) {
    int idx = lower_bound(t);
    int low = idx - 1, high = (int) SA.size();
    t.back()++;
    while(high - low > 1) {
      int mid = (low + high) / 2;
      if(lt_substr(t, SA[mid])) low = mid;
      else high = mid;
    }
    t.back()--;
    return {idx, high};
  }

  void output() {
    for(int i = 0; i < size(); i++) {
      cout << i << ": " << s.substr(SA[i]) << endl;
    }
  }
};

struct LongestCommonPrefixArray {
  const SuffixArray &SA;
  vector< int > LCP, rank;

  LongestCommonPrefixArray(const SuffixArray &SA) : SA(SA), LCP(SA.size()) {
    rank.resize(SA.size());
    for(int i = 0; i < SA.size(); i++) {
      rank[SA[i]] = i;
    }
    for(int i = 0, h = 0; i < SA.size(); i++) {
      if(rank[i] + 1 < SA.size()) {
        for(int j = SA[rank[i] + 1]; max(i, j) + h < SA.size() && SA.s[i + h] == SA.s[j + h]; ++h);
        LCP[rank[i] + 1] = h;
        if(h > 0) --h;
      }
    }
  }

  int operator[](int k) const {
    return LCP[k];
  }

  size_t size() const {
    return LCP.size();
  }

  void output() {
    for(int i = 0; i < size(); i++) {
      cout << i << ": " << LCP[i] << " " << SA.s.substr(SA[i]) << endl;
    }
  }
};




signed main(){
  cin.tie(0);
  ios::sync_with_stdio(false);
  string s;
  cin>>s;
  SuffixArray sa(s);
  int n;
  cin>>n;
  sa.output();
  return 0;
}

```