Aizu University (team: mo3zimanju)

```cpp
##################################################
################## 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));
      }
    }
  }
}


signed main(){
  int v,e,r;
  cin>>v>>e>>r;
  rep(i,e){
    int a,b,c;
    cin>>a>>b>>c;
    mp[a].edges.push_back(make_pair(c,b));
```

```
  }
  dijkstra(r,v);
  rep(i,v){
    if(mp[i].cost==-1) cout<<"INF"<<endl;
    else cout<<mp[i].cost<<endl;
  }
  return 0;
}


##################################################
################ 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;
}


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;
}


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


  return 0;
}


##################################################
################# 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;
}

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


  return 0;
}


##################################################
################ segment_tree.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"

class segtree{
  int n;


signed main(){
  cin.tie(0);
```

```
  ios::sync_with_stdio(false);


  return 0;
}


##################################################
################ combination.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

int aa[200000],bb[200000],cc[2000000];

void t(){
  aa[0]=aa[1]=1;
  bb[0]=bb[1]=1;
  cc[1]=1;
  repi(i,2,200000){
    aa[i]=aa[i-1]*i%MOD;
    cc[i]=MOD-cc[MOD%i]*(MOD/i)%MOD;
    bb[i]=bb[i-1]*cc[i]%MOD;
  }
}

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



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

  return 0;
}


##################################################
#################### bfs.cpp #####################
##################################################

int a[N][M];

void bfs(){



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

```
281:
282: #include<bits/stdc++.h>
283: using namespace std;
284: #define MOD 1000000007
285: #define BIG 1000000010
286: #define EPS 1e-9
287: #define fst first
288: #define scd second
289:
290: #define debug(x) cout<<x<<endl;
291: #define repi(i,x,n) for(int i=x;i<n;i++)
292: #define rep(i,n) repi(i,0,n)
293: #define repn(i,n) for(int i=n;i>=0;i--)
294: #define int long long
295:
296: const int MAX=500000;
297:
298: int fac[MAX],finv[MAX],inv[MAX];
299:
300: void t(){
301:   fac[0]=fac[1]=1;
302:   finv[0]=finv[1]=1;
303:   inv[1]=1;
304:   repi(i,2,MAX){
305:     fac[i]=fac[i-1]*i%MOD;
306:     inv[i]=MOD-inv[MOD%i]*(MOD/i)%MOD;
307:     finv[i]=finv[i-1]*inv[i]%MOD;
308:   }
309: }
310:
311: int calc(int n,int k){
312:   if(n<k) return 0;
313:   if(n<0 || k<0) return 0;
314:   return fac[n]*(finv[k]*finv[n-k]%MOD)%MOD;
315: }
316:
317:
318:
319: signed main(){
320:   int n,m;
321:   cin>>n>>m;
322:   t();
323:   cout<<calc(n,m)<<endl;
324:
325:   return 0;
326: }
327:
328:
329: ##################################################
330: ################## zalgorithm.cpp ################
331: ##################################################
332:
333: #include<bits/stdc++.h>
334: using namespace std;
335: #define MOD 1000000007
336: #define BIG 1000000010
337: #define EPS 1e-9
338: #define fst first
339: #define scd second
340:
341: #define debug(x) cout<<x<<endl;
342: #define repi(i,x,n) for(int i=x;i<n;i++)
343: #define rep(i,n) repi(i,0,n)
344: #define repn(i,n) for(int i=n;i>=0;i--)
345: #define int long long
346: #define endl "\n"
347:
348: int A[100000];
349:
350: void zalgorithm(int a,string S){
```

```
351:   A[0] = S.size();
352:   int i = 1, j = 0;
353:   while (i < S.size()) {
354:     while (i+j < S.size() && S[j] == S[i+j]) ++j;
355:     A[i] = j;
356:     if (j == 0) { ++i; continue;}
357:     int k = 1;
358:     while (i+k < S.size() && k+A[k] < j) A[i+k] = A[k], ++k;
359:     i += k; j -= k;
360:   }
361: }
362:
363: signed main(){
364:   cin.tie(0);
365:   ios::sync_with_stdio(false);
366:   string s;
367:   cin>>s;
368:   zalgorithm(s);
369:   rep(i,s.size() ){
370:     cout<<A[i];
371:   }
372:   cout<<endl;
373:
374:   return 0;
375: }
376:
377:
378: ////////////////////////////////////////////////////
379: ////////////////// prime.cpp ///////////////////////
380: ////////////////////////////////////////////////////
381:
382: #include<bits/stdc++.h>
383: using namespace std;
384: #define MOD 1000000007
385: #define BIG 1000000010
386: #define repi(i,x,n) for(int i=x;i<n;i++)
387: #define rep(i,n) repi(i,0,n)
388: #define repn(i,n) for(int i=n;i>=0;i--)
389: typedef long long int ll;
390:
391: #define N 100000
392:
393: int prime[N];
394:
395: void eratosthenes(){
396:   rep(i,N) prime[i]=1;
397:   prime[0]=prime[1]=0;
398:   rep(i,n){
399:     if(prime[i]){
400:       for(int j=i+i;j<N;j+=i) prime[j]=0;
401:     }
402:   }
403: }
404:
405: bool primeNumber(int n){
406:   if(n < 2) return false;
407:   else{
408:     for(int i = 2; i * i <= n; i++){
409:       if(n % i == 0) return false;
410:     }
411:     return true;
412:   }
413: }
414:
415:
416:
417: int main(){
418:
419:
420:   return 0;
```

```
421: }
422:
423:
424: #include<bits/stdc++.h>
425: using namespace std;
426: #define MOD 1000000007
427: #define BIG 1000000010
428: #define repi(i,x,n) for(int i=x;i<n;i++)
429: #define rep(i,n) repi(i,0,n)
430: #define repn(i,n) for(int i=n;i>=0;i--)
431: typedef long long int ll;
432:
433: #define N 100000
434:
435: int prime[N];
436:
437: void eratosthenes(){
438:   rep(i,N) prime[i]=1;
439:   prime[0]=prime[1]=0;
440:   rep(i,n){
441:     if(prime[i]){
442:       for(int j=i+i;j<N;j+=i) prime[j]=0;
443:     }
444:   }
445: }
446:
447: bool primeNumber(int n){
448:   if(n < 2) return false;
449:   else{
450:     for(int i = 2; i * i <= n; i++){
451:       if(n % i == 0) return false;
452:     }
453:     return true;
454:   }
455: }
456:
457:
458:
459: int main(){
460:
461:
462:   return 0;
463: }
464:
465:
466:
467:
468: ////////////////////////////////////////////////////
469: ////////////////// power.cpp ///////////////////////
470: ////////////////////////////////////////////////////
471:
472: #include<bits/stdc++.h>
473: using namespace std;
474: #define MOD 1000000007
475: //#define MOD 998244353
476: #define INF 1000000010
477: #define EPS 1e-9
478: #define F first
479: #define S second
480:
481: #define debug(x) cout<<x<<endl;
482: #define repi(i,x,n) for(int i=x;i<n;i++)
483: #define rep(i,n) repi(i,0,n)
484: #define lp(i,n) repi(i,0,n)
485: #define repn(i,n) for(int i=n;i>=0;i--)
486: #define int long long
487: #define endl "\n"
488:
489: typedef pair<int,int> PII;
490: typedef pair<int,string> PIS;
```

```
491: typedef pair<string,int> PSI;
492:
493: int power(int n,int m){
494:   int now=2;
495:   int num=1;
496:   while(n!=0){
497:     //cout<<n<<" "<<m<<endl;
498:     if(n%2!=0){
499:       num=(num*m)%MOD;
500:
501:     }
502:     n/=2;
503:     m=(m*m)%MOD;
504:   }
505:   return num;
506: }
507:
508: signed main(){
509:   cin.tie(0);
510:   ios::sync_with_stdio(false);
511:   int a,b;
512:   cin>>a>>b;
513:   cout<<power(b,a)<<endl;
514:
515:   return 0;
516: }
517:
518:
519: ##################################################
520: ################ rollinghash.cpp #################
521: ##################################################
522:
523: #include<bits/stdc++.h>
524: using namespace std;
525: using Int = long long;
526: template<typename T1,typename T2> inline void chmin(T1 &a,T2 b){if(a>b) a=b;}
527: template<typename T1,typename T2> inline void chmax(T1 &a,T2 b){if(a<b) a=b;}
528: //BEGIN CUT HERE
529: template<typename T,T MOD,T B>
530: struct RollingHash{
531:   vector<T> hash,p;
532:   RollingHash(){}
533:   RollingHash(const string &s){
534:     int n=s.size();
535:     hash.assign(n+1,0);
536:     p.assign(n+1,1);
537:     for(int i=0;i<n;i++){
538:       hash[i+1]=(hash[i]*B+s[i])%MOD;
539:       p[i+1]=p[i]*B%MOD;
540:     }
541:   }
542:   //S[l, r)
543:   T find(int l,int r){
544:     T res=hash[r]+MOD-hash[l]*p[r-l]%MOD;
545:     return res>=MOD?res-MOD:res;
546:   }
547: };
548: //END CUT HERE
549: //INSERT ABOVE HERE
550: signed main(){
551:   cin.tie(0);
552:   ios::sync_with_stdio(0);
553:   string t,p;
554:   cin>>t>>p;
555:   using ll = long long;
556:   const ll MOD = 1e9+7;
557:   const ll B = 1777771;
558:   using RH = RollingHash<ll, MOD, B>;
559:   RH rt(t),rp(p);
560:   for(int i=0;i<(int)t.size()-(int)p.size()+1;i++){
```

```
561:     if(rt.find(i,i+p.size())==rp.find(0,p.size())) cout<<i<<"\n";
562:   }
563:   cout<<flush;
564:   return 0;
565: }
566:
567:
568: ##################################################
569: ############### unionfindtree.cpp ################
570: ##################################################
571:
572: #include<bits/stdc++.h>
573: using namespace std;
574: #define MOD 1000000007
575: //#define MOD 998244353
576: #define INF 1000000010
577: #define EPS 1e-9
578: #define F first
579: #define S second
580:
581: #define debug(x) cout<<x<<endl;
582: #define repi(i,x,n) for(int i=x;i<n;i++)
583: #define rep(i,n) repi(i,0,n)
584: #define lp(i,n) repi(i,0,n)
585: #define repn(i,n) for(int i=n;i>=0;i--)
586: #define int long long
587: #define endl "\n"
588:
589: typedef pair<int,int> PII;
590: typedef pair<int,string> PIS;
591: typedef pair<string,int> PSI;
592:
593: struct UnionFind{
594:   vector<int> data;
595:
596:   UnionFind(int N){
597:     data.assign(N,-1);
598:   }
599:
600:  bool unite(int x, int y) {
601:     x = find(x), y = find(y);
602:     if(x == y) return (false);
603:     if(data[x] > data[y]) swap(x, y);
604:     data[x] += data[y];
605:     data[y] = x;
606:     return (true);
607:   }
608:
609:   int find(int k) {
610:     if(data[k] < 0) return (k);
611:     return (data[k] = find(data[k]));
612:   }
613:
614:   int size(int k) {
615:     return (-data[find(k)]);
616:   }
617: };
618:
619: signed main(){
620:   cin.tie(0);
621:   ios::sync_with_stdio(false);
622:
623:
624:   return 0;
625: }
626:
627:
628: ##################################################
629: ############### suffix_array.cpp ################
630: ##################################################
```

```
  631:
  632: #include<bits/stdc++.h>
  633: using namespace std;
  634: #define MOD 1000000007
  635: #define INF 1000000010
  636: #define EPS 1e-9
  637: #define fst first
  638: #define scd second
  639:
  640: #define debug(x) cout<<x<<endl;
  641: #define repi(i,x,n) for(int i=x;i<n;i++)
  642: #define rep(i,n) repi(i,0,n)
  643: #define lp(i,n) repi(i,0,n)
  644: #define repn(i,n) for(int i=n;i>=0;i--)
  645: #define int long long
  646: #define endl "\n"
  647:
  648:
  649: struct SuffixArray {
  650:   vector< int > SA;
  651:   const string s;
  652:
  653:   SuffixArray(const string &str) : s(str) {
  654:     SA.resize(s.size());
  655:     iota(begin(SA), end(SA), 0);
  656:     sort(begin(SA), end(SA), [&](int a, int b) {
  657:       return s[a] == s[b] ? a > b : s[a] < s[b];
  658:     });
  659:     vector< int > classes(s.size()), c(s.begin(), s.end()), cnt(s.size());
  660:     for(int len = 1; len < s.size(); len <<= 1) {
  661:       for(int i = 0; i < s.size(); i++) {
  662:         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]) {
  663:           classes[SA[i]] = classes[SA[i - 1]];
  664:         } else {
  665:           classes[SA[i]] = i;
  666:         }
  667:       }
  668:       iota(begin(cnt), end(cnt), 0);
  669:       copy(begin(SA), end(SA), begin(c));
  670:       for(int i = 0; i < s.size(); i++) {
  671:         int s1 = c[i] - len;
  672:         if(s1 >= 0) SA[cnt[classes[s1]]++] = s1;
  673:       }
  674:       classes.swap(c);
  675:     }
  676:   }
  677:
  678:   int operator[](int k) const {
  679:     return SA[k];
  680:   }
  681:
  682:   size_t size() const {
  683:     return s.size();
  684:   }
  685:
  686:   bool lt_substr(const string &t, int si = 0, int ti = 0) {
  687:     int sn = (int) s.size(), tn = (int) t.size();
  688:     while(si < sn && ti < tn) {
  689:       if(s[si] < t[ti]) return true;
  690:       if(s[si] > t[ti]) return false;
  691:       ++si, ++ti;
  692:     }
  693:     return si >= sn && ti < tn;
  694:   }
  695:
  696:   int lower_bound(const string &t) {
  697:     int low = -1, high = (int) SA.size();
  698:     while(high - low > 1) {
  699:       int mid = (low + high) / 2;
```

```
700:       if(lt_substr(t, SA[mid])) low = mid;
701:       else high = mid;
702:     }
703:     return high;
704:   }
705:
706:   pair< int, int > lower_upper_bound(string &t) {
707:     int idx = lower_bound(t);
708:     int low = idx - 1, high = (int) SA.size();
709:     t.back()++;
710:     while(high - low > 1) {
711:       int mid = (low + high) / 2;
712:       if(lt_substr(t, SA[mid])) low = mid;
713:       else high = mid;
714:     }
715:     t.back()--;
716:     return {idx, high};
717:   }
718:
719:   void output() {
720:     for(int i = 0; i < size(); i++) {
721:       cout << i << ": " << s.substr(SA[i]) << endl;
722:     }
723:   }
724: };
725:
726: struct LongestCommonPrefixArray {
727:   const SuffixArray &SA;
728:   vector< int > LCP, rank;
729:
730:   LongestCommonPrefixArray(const SuffixArray &SA) : SA(SA), LCP(SA.size()) {
731:     rank.resize(SA.size());
732:     for(int i = 0; i < SA.size(); i++) {
733:       rank[SA[i]] = i;
734:     }
735:     for(int i = 0, h = 0; i < SA.size(); i++) {
736:       if(rank[i] + 1 < SA.size()) {
737:         for(int j = SA[rank[i] + 1]; max(i, j) + h < SA.size() && SA.s[i + h] == S
A.s[j + h]; ++h);
738:         LCP[rank[i] + 1] = h;
739:         if(h > 0) --h;
740:       }
741:     }
742:   }
743:
744:   int operator[](int k) const {
745:     return LCP[k];
746:   }
747:
748:   size_t size() const {
749:     return LCP.size();
750:   }
751:
752:   void output() {
753:     for(int i = 0; i < size(); i++) {
754:       cout << i << ": " << LCP[i] << " " << SA.s.substr(SA[i]) << endl;
755:     }
756:   }
757: };
758:
759:
760:
761:
762: signed main(){
763:   cin.tie(0);
764:   ios::sync_with_stdio(false);
765:   string s;
766:   cin>>s;
767:   SuffixArray sa(s);
768:   int n;
```

```
  769:    cin>>n;
  770:    sa.output();
  771:    return 0;
  772: }
  773:
  774:
```