```
#include<bits/stdc++.h>
using namespace std;
using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)

template<class T>
class segtree {
private:
 ll n=1;
 T unit;
 std::vector<T> dat;
 std::function<T(T,T)> func;
public:
 segtree(){}
 segtree(std::vector<T>& a,T v,std::function<T(T,T
)> f){
  build(a,v,f);
 }
 void build(std::vector<T>& a,T v,std::function<T(T
,T)> f){
  ll sz=a.size();
  unit=v;
  func=f;
  while(n<sz)n<<=1LL;
  dat.resize(2*n-1,unit);
  for(int i=0;i<sz;++i)dat[i+n-1]=a[i];
  for(int i=n-2;i>=0;--i){
   dat[i]=func(dat[i*2+1],dat[i*2+2]);
  }
 }
 void update(ll idx,T val){
  idx+=n-1;
  dat[idx]=val;
  while(idx){
   idx--;idx>>=1LL;
   dat[idx]=func(dat[idx*2+1],dat[idx*2+2]);
  }
 }
 void add(ll idx, T val){
  idx+=n-1;
  dat[idx]+=val;
  while(idx){
   idx--;idx>>=1LL;
   dat[idx]=func(dat[idx*2+1],dat[idx*2+2]);
  }
 }
 T query(ll a,ll b,ll k=0,ll l=0,ll r=-1){
  if(r<0)r=n;
  if( b<=l || r<=a )return unit;
  if( a<=l && r<=b )return dat[k];
  else{
   auto left=query(a,b,2*k+1,l,(l+r)/2);
   auto right=query(a,b,2*k+2,(l+r)/2,r);
   return func(left,right);
  }
 }
};

void solve_RMQ(){
 int n,q;
 cin>>n>>q;
 ll base = (1LL<<31)-1;
 vector<ll> a(n,base);
 auto f = [](auto a,auto b){return min(a,b);};
 segtree<ll> rmq(a,base,f);
 rep(i,q){
  ll c,x,y;
  cin>>c>>x>>y;
  if(c){
   auto tmp = rmq.query(x,y+1);
   cout<<(tmp)<<endl;
  }
  else{
   rmq.update(x,y);
  }
 }
}
void solve_RSQ(){
 ll n,q;
 cin>>n>>q;
 ll base=0;
 auto f = [](auto a,auto b){return a+b;};
 vector<ll> a(n);
 segtree<ll> rsq(a,base,f);
```

```
  rep(i,q){
    ll c,x,y;
    cin>>c>>x>>y;
    if(c){
      x--;y--;
      auto tmp=rsq.query(x,y+1);
      cout<<(tmp)<<endl;
    }
    else{
      x--;
      rsq.add(x,y);
    }
  }
}

main(){
  solve_RMQ();
  // https://onlinejudge.u-
aizu.ac.jp/courses/library/3/DSL/2/DSL_2_A
  // solve_RSQ();
  // https://onlinejudge.u-
aizu.ac.jp/courses/library/3/DSL/2/DSL_2_B
}
```

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

// https://onlinejudge.u-aizu.ac.jp/courses/library/5/GRL/1/GRL_1_B

using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)
constexpr long long INF = (1LL<<60);
template<class T>inline bool chmin(T &a,const T &b){if(a>b){a=b;return true;}return false;}
using edge = struct edge{ll from,to,cost;edge(ll from,ll to,ll cost):from(from),to(to),cost(cost){}};
using weighted_edge = vector<edge>;

auto Bellman_Ford(weighted_edge& edges,ll n,ll s,ll k=1){
  vector<ll> dist(n,INF);
  dist[s]=0LL;
  rep(i,n*k){
    for(const auto& e:edges){
      if(dist[e.from]==INF)continue;
      chmin(dist[e.to],dist[e.from]+e.cost);
    }
  }
  bool negative_cycle=false;
  rep(i,n*k){
    for(const auto& e:edges){
      if(dist[e.from]==INF)continue;
      if( chmin(dist[e.to],dist[e.from]+e.cost) ){
        negative_cycle=true;
      }
    }
  }
  return make_pair(dist,negative_cycle);
}

void solve(){

  ll v,e,r;
  cin>>v>>e>>r;
  weighted_edge edges;
  rep(i,e){
    ll u,v,w;
    cin>>u>>v>>w;
    edges.emplace_back(u,v,w);
  }
  bool negative_cycle;
  vector<ll> dist;
  tie(dist,negative_cycle) = Bellman_Ford(edges,v,r,2);


  if(negative_cycle)puts("NEGATIVE CYCLE");
  else{
    for(auto d:dist){
      if(d==INF)puts("INF");
      else cout<<d<<endl;
    }
  }


}

main(){
  solve();
}
```

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

// https://onlinejudge.u-aizu.ac.jp/courses/library/5/GRL/1/GRL_1_A

#define rep(i,n) for(ll i=0;i<(n);++i)
template<class T>inline bool chmin(T &a,const T &b){if(a>b){a=b;return true;}return false;}
constexpr long long INF = (1LL<<60);
using ll = long long;
using vertex = struct vertex{ll to,cost;vertex(ll to,ll cost):to(to),cost(cost){}};
using weighted_adjlist = vector<vector<vertex>>;

auto Dijkstra(weighted_adjlist& adj,ll S){
  ll N=(ll)adj.size();
  vector<ll> dist(N,(1LL<<60));
  vector<ll> prev_v(N,-1LL);
  using pll = pair<ll,ll>;
  priority_queue<pll,vector<pll>,greater<>> que;
  dist[S]=0LL;
  que.emplace(dist[S],S);
  while(!que.empty()){
    ll cost,from;
    tie(cost,from) = que.top();
    que.pop();
    if(dist[from]<cost)continue;
    for(const auto& v:adj[from]){
      if( chmin(dist[v.to],dist[from]+v.cost) ){
        prev_v[v.to]=from;
        que.emplace(dist[v.to],v.to);
      }
    }
  }
  return dist;
  //return prev_v;
  //return make_pair(dist,prev_v);
}

void solve(){
  ll v,e,r;
  cin>>v>>e>>r;
  weighted_adjlist adj(v);
  rep(i,e){
    int u,v,w;
    cin>>u>>v>>w;
    adj[u].emplace_back(v,w);
  }
  auto dist = Dijkstra(adj,r);

  for(auto d:dist){
    if(d==INF)puts("INF");
    else cout<<d<<endl;
  }

}

main(){
  solve();
}
```

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

// https://onlinejudge.u-aizu.ac.jp/courses/library/5/GRL/all/GRL_2_A

using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)
using edge = struct edge{ll from,to,cost;edge(ll from,ll to,ll cost):from(from),to(to),cost(cost){}};
using weighted_edge = vector<edge>;

class Union_Find {
private:
 ll n;
 vector<ll> parent;

public:
 Union_Find(ll v){
  n=v;
  parent.resize(n,-1);
 }

 ll root(ll v){
  if(parent[v]<0)return v;
  return parent[v]=root(parent[v]);
 }
 ll size(ll v){
  return -parent[root(v)];
 }
 bool connect(ll a,ll b){
  a=root(a);
  b=root(b);
  if(a==b)return false;
  if(size(a)<size(b)) swap(a,b);
  parent[a]+=parent[b];
  parent[b]=a;
  return true;
 }
 bool same(ll a,ll b){
  return root(a)==root(b);
 }
};

void solve(){

 ll v,e;
 cin>>v>>e;
 weighted_edge edges;
 rep(i,e){
  ll u,v,w;
  cin>>u>>v>>w;
  edges.emplace_back(u,v,w);
 }
 auto cmp = [](auto a,auto b){return a.cost<b.cost;};
 sort(edges.begin(),edges.end(),cmp);
 Union_Find uni(v);
 ll ans=0;
 for(const auto& e:edges){
  if( uni.same(e.from,e.to) )continue;
  uni.connect(e.from,e.to);
  ans+=e.cost;
 }
 cout<<(ans)<<endl;

}

main(){
 solve();
}
```

```
#include<bits/stdc++.h>
using ll = long long;

template<class T>
class segtree {
private:
 ll n=1;
 T unit;
 std::vector<T> dat;
 std::function<T(T,T)> func;
public:
 segtree(){}
 segtree(std::vector<T>& a,T v,std::function<T(T,T)> f){
  build(a,v,f);
 }
 void build(std::vector<T>& a,T v,std::function<T(T,T)> f){
  ll sz=a.size();
  unit=v;
  func=f;
  while(n<sz)n<<=1LL;
  dat.resize(2*n-1,unit);
  for(int i=0;i<sz;++i)dat[i+n-1]=a[i];
  for(int i=n-2;i>=0;--i){
   dat[i]=func(dat[i*2+1],dat[i*2+2]);
  }
 }
 void update(ll idx,ll val){
  idx+=n-1;
  dat[idx]=val;
  while(idx){
   idx--;idx>>=1LL;
   dat[idx]=func(dat[idx*2+1],dat[idx*2+2]);
  }
 }
 T query(ll a,ll b,ll k=0,ll l=0,ll r=-1){
  if(r<0)r=n;
  if( b<=l || r<=a )return unit;
  if( a<=l && r<=b )return dat[k];
  else{
   auto left=query(a,b,2*k+1,l,(l+r)/2);
   auto right=query(a,b,2*k+2,(l+r)/2,r);
   return func(left,right);
  }
 }
};

// lowest common ancestor
class LCA{
private:
 using pll = std::pair<ll,ll>;
 ll n,root;
 std::vector<std::vector<ll>> adj;
 std::vector<ll> vs,depth,id;
 std::function<pll(pll,pll)> mindex = [](auto const& a,auto const& b){return min(a,b);};
 segtree<pll> RMQ;
public:
 LCA(){}
 LCA(std::vector<std::vector<ll>> const& a):adj(a),root(0){
  init();
 }
 LCA(std::vector<std::vector<ll>> const& a,ll r):adj(a),root(r){
  init();
 }
 void dfs(ll v,ll p,ll d,ll& k){
  id[v]=k;
  vs[k]=v;
  depth[k++]=d;
  for(auto const& nv:adj[v]){
   if(nv!=p){
    dfs(nv,v,d+1,k);
    vs[k]=v;
    depth[k++]=d;
   }
  }
 }
 void init(){
  n=adj.size();
  vs.resize(2*n-1);
  depth.resize(2*n-1);
```

```
    id.resize(n);
    ll k=0;
    dfs(root,-1,0,k);
    constexpr ll INF = (1LL<<60);
    pll base(INF,INF);
    std::vector<std::pair<ll,ll>>dat(2*n-1);
    for(ll i=0;i<2*n-1;++i)dat[i]=std::make_pair(depth[i],i);
    RMQ.build(dat,base,mindex);
  }
  ll lca(ll u,ll v){
    auto t = RMQ.query(std::min(id[u],id[v]),std::max(id[u],id[v])+1LL);
    return vs[t.second];
  }
  ll operator[](ll i)const{
    return depth[id[i]];
  }
};

void solve1(){
  // https://atcoder.jp/contests/abc014/tasks/abc014_4
  ll n;
  std::cin>>n;
  std::vector<std::vector<ll>> adj(n);
  for(ll i=0;i<n-1;++i){
    ll x,y;
    std::cin>>x>>y;
    x--;y--;
    adj[x].emplace_back(y);
    adj[y].emplace_back(x);
  }
  LCA g(adj);
  ll q;
  std::cin>>q;
  std::vector<ll> a(q),b(q);
  for(ll i=0;i<q;++i){
    std::cin>>a[i]>>b[i];
    a[i]--;b[i]--;
  }
  for(ll i=0;i<q;++i){
    ll c=g.lca(a[i],b[i]);
    ll ans = g[a[i]]+g[b[i]]-g[c]*2+1;
    std::cout<<(ans)<<std::endl;
  }
}

main(){
  solve1();
}
```

```
#include<bits/stdc++.h>
using namespace std;
using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)

constexpr ll COM_MAX = (ll)1e6;
vector<ll> fac(COM_MAX),finv(COM_MAX),inv(COM_MAX);
//前処理
void comb_init(){
 fac[0]=fac[1]=1;
 finv[0]=finv[1]=1;
 inv[1]=1;
 for(int i=2;i<COM_MAX;++i){
  fac[i]=fac[i-1]*i%MOD;
  inv[i]=MOD-inv[MOD%i]*(MOD/i)%MOD;
  finv[i]=finv[i-1]*inv[i]%MOD;
 }
}
//二項係数計算
ll comb(ll n,ll k){
 if(n<k)return 0;
 if(n<0||k<0)return 0;
 return fac[n]*(finv[k]*finv[n-k]%MOD)%MOD;
}
//重複組み合わせ
ll nHk(ll n,ll k){
 return comb(n+k-1,n-1);
}

std::vector<ll> divisor(ll n){
 std::vector<ll> div;
 for(ll p=1;p*p<=n;++p){
  if(n%p==0){
   div.push_back(p);
   if(p*p!=n)div.push_back(n/p);
  }
 }
 std::sort(div.begin(),div.end());
 return div;
}
```

```
std::vector<std::pair<ll,ll>> prime_factorization(ll n)
{
 std::vector<std::pair<ll,ll>> factor;
 for(ll p=2;p*p<=n;++p){
  if(n%p!=0)continue;
  ll num=0;
  while(n%p==0){
   num++;
   n/=p;
  }
  factor.emplace_back(p,num);
  if(n==1)return factor;
 }
 if(n!=1)factor.emplace_back(n,1);
 return factor;
}

using Real = long double;
using Complex = complex<Real>;
constexpr Real PI = (Real)acos(-1);

vector<Complex> DFT(vector<Complex> a){
 ll n = a.size();
 if(n==1)return a;
 vector<Complex> a0(n/2);
 vector<Complex> a1(n/2);
 rep(i,n/2)a0[i]=a[i*2];
 rep(i,n/2)a1[i]=a[i*2+1];
 vector<Complex> inversed_a0 = DFT(a0);
 vector<Complex> inversed_a1 = DFT(a1);
 vector<Complex> inversed_a(n);
 rep(i,n){
  Complex zeta_n_i = Complex( cos(2*PI*i/n),sin(2*PI*i/n) );
  inversed_a[i] = inversed_a0[i%(n/2)]+zeta_n_i*inversed_a1[i%(n/2)];
 }
 return inversed_a;
}
vector<Complex> IDFT(vector<Complex> inversed_a){
 ll n = inversed_a.size();
 vector<Complex> arranged = DFT(inversed_a);
 vector<Complex>swaped(n);
 rep(i,n)swaped[i]=arranged[(n-i)%n];
 vector<Complex> a(n);
 rep(i,n)a[i]=swaped[i]/Complex(n,0);
 return a;
}
vector<Complex> conv(vector<Complex> a,vector<
Complex> b){
 ll deg = a.size()+b.size()-1;
 ll n = 1;
 while(n<deg)n<<=1;
 a.resize(n);b.resize(n);
 vector<Complex> inversed_c(n);
 vector<Complex> inversed_a = DFT(a);
 vector<Complex> inversed_b = DFT(b);
 rep(i,n){
  inversed_c[i] = inversed_a[i]*inversed_b[i];
 }
 vector<Complex> c = IDFT(inversed_c);
 return c;
}

main(){

}
```

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

// https://ei1333.github.io/luzhiled/snippets/math/mod-int.html
// http://noshi91.hatenablog.com/entry/2019/03/31/174006

template<long long MOD>
struct mod_int{
  using ll = long long;
  ll val;
  constexpr mod_int():val(0){}
  constexpr mod_int(ll x)noexcept:val(x>=0?x%MOD:(MOD-(-x)%MOD)%MOD){}
  constexpr ll value()const noexcept{return val;}
  constexpr mod_int operator+(mod_int const& rhs){
    return mod_int(*this)+=rhs;
  }
  constexpr mod_int operator-(mod_int const& rhs){
    return mod_int(*this)-=rhs;
  }
  constexpr mod_int operator*(mod_int const& rhs){
    return mod_int(*this)*=rhs;
  }
  constexpr mod_int operator/(mod_int const& rhs){
    return mod_int(*this)/=rhs;
  }
  constexpr mod_int &operator+=(mod_int const& rhs)noexcept{
    val += rhs.val;
    if(val>=MOD)val-=MOD;
    return *this;
  }
  constexpr mod_int &operator-=(mod_int const& rhs)noexcept{
    if(val<rhs.val)val+=MOD;
    val-=rhs.val;
    return *this;
  }
  constexpr mod_int &operator*=(mod_int const& rhs)noexcept{
    (val*=rhs.val)%=MOD;
    return *this;
  }
  constexpr mod_int &operator/=(mod_int rhs)noexcept{
    *this*=rhs.inverse();
    return *this;
  }
  constexpr bool operator==(mod_int const& rhs){
    return val==rhs.val;
  }
  constexpr bool operator!=(mod_int const& rhs){
    return !(val==rhs.val);
  }
  constexpr bool operator<(mod_int const& rhs){
    return val<rhs.val;
  }
  constexpr bool operator>(mod_int const& rhs){
    return val>rhs.val;
  }
  constexpr bool operator<=(mod_int const& rhs){
    return !(val>rhs.val);
  }
  constexpr bool operator>=(mod_int const& rhs){
    return !(val<rhs.val);
  }
  constexpr friend std::ostream &operator<<(std::ostream& os, mod_int const& mi){
    return os<<mi.val;
  }
  friend std::istream &operator>>(std::istream& is, mod_int & mi){
    ll t;is>>t;
    mi = mod_int<MOD>(t);
    return is;
  }
  constexpr mod_int inverse(){
    ll a=val,b=MOD,u=1,v=0,t;
    while(b>0){
      t=a/b;
      std::swap(a-=t*b,b);
      std::swap(u-=t*v,v);
    }
  return mod_int(u);
 }
 constexpr mod_int mpow(ll n){
  mod_int res(1),mul(val);
  while(n>0){
   if(n&1)res*=mul;
   mul*=mul;
   n>>=1;
  }
  return res;
 }
 constexpr friend  mod_int mpow(ll x,ll n){
  mod_int res(1),mul(x);
  while(n>0){
   if(n&1)res*=mul;
   mul*=mul;
   n>>=1;
  }
  return res;
 }
};
using mint = mod_int<1000000007>;

void solve1(){
 // https://atcoder.jp/contests/abc037/tasks/abc037_d
 int h,w;
 std::cin>>h>>w;
 std::vector<std::vector<mint>> a(h,std::vector<mint>(w));
 for(int i=0;i<h;++i)for(int j=0;j<w;++j){
  std::cin>>a[i][j];
 }
 std::vector<std::vector<mint>> dp(h,std::vector<mint>(w,-1));
 constexpr int dx[]={1,0,-1,0},dy[]={0,1,0,-1};
 auto dfs = [&](auto&& dfs,int y,int x)->mint{
  if(dp[y][x]!=-1)return dp[y][x];
  mint res=1;
  for(int i=0;i<4;++i){
   int ny=y+dy[i],nx=x+dx[i];
   if(ny==-1||nx==-1||ny==h||nx==w)continue;
   if(a[ny][nx]<=a[y][x])continue;
   res += dfs(dfs,ny,nx);
  }
  return dp[y][x]=res;
 };
 mint ans=0;
 for(int i=0;i<h;++i)for(int j=0;j<w;++j){
  ans += dfs(dfs,i,j);
 }
 std::cout<<(ans)<<std::endl;
}

void solve2(){
 //https://atcoder.jp/contests/abc135/tasks/abc135_d
 std::string s;
 std::cin>>s;
 size_t n=s.size();
 std::vector<int> coef(n+1);
 coef[0]=1;
 for(size_t i=1;i<n;++i)coef[i]=coef[i-1]*10%13;
 std::vector<std::vector<mint>> dp(n+1,std::vector<mint>(13));
 dp[0][0]=1;
 for(size_t i=0;i<n;++i){
  if(s[i]=='?'){
   for(int j=0;j<10;++j)for(int k=0;k<13;++k){
    dp[i+1][(j*coef[n-i-1]+k)%13] += dp[i][k];
   }
  }else{
   int num = (s[i]-'0')*coef[n-i-1];
   for(int j=0;j<13;++j){
    dp[i+1][(num+j)%13] += dp[i][j];
   }
  }
 }
 std::cout<<(dp[n][5])<<std::endl;
}

main(){}
```

```
#include<bits/stdc++.h>
using namespace std;
using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)

class Rolling_Hash{
private:
  ll n;
  static constexpr ll base1=1007,base2=2009;
  static constexpr ll mod1=(1e9+7),mod2=(1e9+9);
  vector<ll> hash1,hash2,mpow1,mpow2;
public:
  Rolling_Hash(const string& s){
    n=s.size();
    hash1.resize(n+1,0);
    hash2.resize(n+1,0);
    mpow1.resize(n+1,1);
    mpow2.resize(n+1,1);
    for(ll i=0;i<n;++i){
      hash1[i+1] = (hash1[i]*base1+s[i])%mod1;
      hash2[i+1] = (hash2[i]*base2+s[i])%mod2;
      mpow1[i+1] = (mpow1[i]*base1)%mod1;
      mpow2[i+1] = (mpow2[i]*base2)%mod2;
    }
  }
  auto get(ll l,ll r){
    ll lefval = hash1[r]-hash1[l]*mpow1[r-l]%mod1;
    ll rigval = hash2[r]-hash2[l]*mpow2[r-l]%mod2;
    if(lefval<0)lefval+=mod1;
    if(rigval<0)rigval+=mod2;
    return make_pair(lefval,rigval);
  }
  auto get_LCP(ll a,ll b){
    ll len = min( n-a+1,n-b+1 );
    ll low=0,high=len;
    while( high-low>1 ){
      ll mid=(high+low)>>1;
      if(get(a,a+mid)!=get(b,b+mid))high=mid;
      else low=mid;
    }
    return low;
  }
};

void solve1(){
  // http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ALDS1_14_B&lang=ja
  string s,t;
  cin>>s>>t;
  ll n=s.size(),m=t.size();
  Rolling_Hash SR(s),TR(t);
  rep(i,n-m+1){
    if(SR.get(i,i+m)==TR.get(0,m)){
      cout<<i<<endl;
    }
  }
}
void solve2(){
  // https://atcoder.jp/contests/abc141/tasks/abc141_e
  ll n;
  string s;
  cin>>n>>s;
  Rolling_Hash rori(s);
  ll ans=0;
  for(ll i=0;i<n;++i){
    for(ll j=i+1;j<n;++j){
      ll lcp = rori.get_LCP(i,j);
      ans=max(ans,min(lcp,j-i));
    }
  }
  cout<<(ans)<<endl;
}

main(){
  //solve1();
  solve2();
}
```

```
#include<bits/stdc++.h>
using namespace std;
using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)
// http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=GRL_3_C&lang=jp

class StronglyConnectedComponents {
private:
  ll n;
  vector<vector<ll>> dag,rdag,st;
  vector<ll> comp,order;
  vector<bool> used;
public:
  StronglyConnectedComponents(const vector<vector<ll>>& adj){
    n = adj.size();
    comp.resize(n);
    used.resize(n);
    dag.resize(n);
    rdag.resize(n);
    for(ll i=0;i<n;++i){
      for(const auto& x:adj[i]){
        dag[i].emplace_back(x);
        rdag[x].emplace_back(i);
      }
    }
  }
  inline auto operator[](ll idx)const{
    return comp[idx];
  }
  void dfs(ll now_v){
    if(used[now_v])return;
    used[now_v]=true;
    for(const auto& next_v:dag[now_v]) dfs(next_v);
    order.push_back(now_v);
  }
  auto rdfs(ll now_v,ll num){
    if(comp[now_v])return false;
    comp[now_v]=num;
    for(const auto& next_v:rdag[now_v]) rdfs(next_v,num);
    return true;
  }
  auto build(){
    for(ll i=0;i<n;++i) dfs(i);
    reverse(order.begin(),order.end());
    ll num=1;
    for(const auto& v:order) num+=rdfs(v,num);
    st.resize(num);
    for(ll i=0;i<n;++i){
      st[comp[i]].emplace_back(i);
    }
    return st;
  }
  void dump(){
    rep(i,n)cout<<i<<" ";cout<<endl;
    rep(i,n)cout<<comp[i]<<" ";cout<<endl;
  }
};

main(){


  ll n,m;
  cin>>n>>m;
  vector<vector<ll>> adj(n);
  rep(i,m){
    ll s,t;
    cin>>s>>t;
    adj[s].emplace_back(t);
  }
  StronglyConnectedComponents scc(adj);
  scc.build();
  ll q;
  cin>>q;
  rep(i,q){
    ll u,v;
    cin>>u>>v;
    if(scc[u]==scc[v])puts("1");
    else puts("0");
  }
}
```

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
using namespace std;

// Sparse Table
template<class MeetSemiLattice> struct SparseTable {
    vector<vector<MeetSemiLattice> > dat;
    vector<int> height;

    SparseTable() { }
    SparseTable(const vector<MeetSemiLattice> &vec) { init(vec); }
    void init(const vector<MeetSemiLattice> &vec) {
        int n = (int)vec.size(), h = 0;
        while ((1<<h) < n) ++h;
        dat.assign(h, vector<MeetSemiLattice>(1<<h));
        height.assign(n+1, 0);
        for (int i = 2; i <= n; i++) height[i] = height[i>>1]+1;
        for (int i = 0; i < n; ++i) dat[0][i] = vec[i];
        for (int i = 1; i < h; ++i)
            for (int j = 0; j < n; ++j)
                dat[i][j] = min(dat[i-1][j], dat[i-1][min(j+(1<<(i-1)),n-1)]);
    }

    MeetSemiLattice get(int a, int b) {
        return min(dat[height[b-a]][a], dat[height[b-a]][b-(1<<height[b-a])]);
    }
};

// Suffix Array ( Manber&Myers: O(n (logn)^2) )
struct SuffixArray {
    string str;
    vector<int> sa;        // sa[i] : the starting index of the i-th smallest suffix (i = 0, 1, ..., n)
    vector<int> lcp;       // lcp[i]: the lcp of sa[i] and sa[i+1] (i = 0, 1, ..., n-1)
    inline int& operator [] (int i) {return sa[i];}

    SuffixArray(const string& str_) : str(str_) { buildSA(); calcLCP(); }
    void init(const string& str_) { str = str_; buildSA(); calcLCP(); }

    // build SA
    vector<int> rank_sa, tmp_rank_sa;
    struct CompareSA {
        int n, k;
        const vector<int> &rank;
        CompareSA(int n, int k, const vector<int> &rank_sa) : n(n), k(k), rank(rank_sa) {}
        bool operator()(int i, int j) {
            if (rank[i] != rank[j]) return (rank[i] < rank[j]);
            else {
                int rank_ik = (i + k <= n ? rank[i + k] : -1);
                int rank_jk = (j + k <= n ? rank[j + k] : -1);
                return (rank_ik < rank_jk);
            }
        }
    };
    void buildSA() {
        int n = (int)str.size();
        sa.resize(n+1), lcp.resize(n+1), rank_sa.resize(n+1), tmp_rank_sa.resize(n+1);
        for (int i = 0; i < n; ++i) sa[i] = i, rank_sa[i] = (int)str[i];
        sa[n] = n, rank_sa[n] = -1;
        for (int k = 1; k <= n; k *= 2) {
            CompareSA csa(n, k, rank_sa);
            sort(sa.begin(), sa.end(), csa);
            tmp_rank_sa[sa[0]] = 0;
            for (int i = 1; i <= n; ++i) {
                tmp_rank_sa[sa[i]] = tmp_rank_sa[sa[i - 1]];
                if (csa(sa[i - 1], sa[i])) ++tmp_rank_sa[sa[i]];
            }
            for (int i = 0; i <= n; ++i) rank_sa[i] = tmp_ra
nk_sa[i];
    }
  }
  vector<int> rsa;
  SparseTable<int> st;
  void calcLCP() {
    int n = (int)str.size();
    rsa.resize(n+1);
    for (int i = 0; i <= n; ++i) rsa[sa[i]] = i;
    lcp.resize(n+1);
    lcp[0] = 0;
    int cur = 0;
    for (int i = 0; i < n; ++i) {
      int pi = sa[rsa[i] - 1];
      if (cur > 0) --cur;
      for (; pi + cur < n && i + cur < n; ++cur) {
        if (str[pi + cur] != str[i + cur]) break;
      }
      lcp[rsa[i] - 1] = cur;
    }
    st.init(lcp);
  }

  // calc lcp
  int getLCP(int a, int b) {          // lcp of str.sutstr(a) and str.substr(b)
    int x=min(rsa[a], rsa[b]);
    int y=max(rsa[a], rsa[b]);
    cout<<sa[x]<<" "<<sa[y]<<" "<<st.get(x,y)<<endl;
    return st.get(x,y);
  }

  void dump_rsa(){
   for(int i=0;i<rsa.size();++i){
    cout<<i<<" "<<rsa[i]<<endl;
   }
  }
};


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

  int N;
  string S;
  cin >> N >> S;

  // Suffix Array 構築
  SuffixArray SA(S);

  // 全探索
  int res = 0;
  for (int i = 0; i < N; ++i) {
    for (int j = i+1; j < N; ++j) {
      int lcp = SA.getLCP(i, j);
      lcp = min(lcp, j-i);
      res = max(res, lcp);
    }
  }
  cout << res << endl;
}
```

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

#define rep(i,n) for(ll i=0;i<(n);++i)
using ll = long long;
constexpr ll INF = (1LL<<60);
constexpr ll MOD = (1e9+7);
constexpr double PI = acos(-1);
constexpr double EPS = (1e-7);
template<typename T>T gcd(T a,T b){while(1){swap(a,b);if(a==0)return b;if(b==0)return a;a%=b;}}
template<class T>inline T lcm(T a,T b){return a/gcd(a,b)*b;}
template<class T>inline bool chmax(T &a,const T &b){if(a<b){a=b;return 1;}return 0;}
template<class T>inline bool chmin(T &a,const T &b){if(a>b){a=b;return 1;}return 0;}
inline void dump(){cout<<endl;}
template<class Head,class... Tail>inline void dump(Head&& head, Tail&&... tail){cout<<head<<", ";dump(forward<Tail>(tail)...);}
template<typename T>inline istream &operator>>(istream&input,vector<T>&v){for(auto &elemnt:v)input>>elemnt;return input;}
template<class T>vector<T> make_vector(size_t a){return vector<T>(a);}
template<class T, class... Ts>auto make_vector(size_t a, Ts... ts){return vector<decltype(make_vector<T>(ts...))>(a, make_vector<T>(ts...));}
using vertex = struct vertex{ll to,cost;vertex(ll to,ll cost):to(to),cost(cost){}};
using edge = struct edge{ll from,to,cost;edge(ll from,ll to,ll cost):from(from),to(to),cost(cost){}};
using weighted_adjlist = vector<vector<vertex>>;
using weighted_edge = vector<edge>;
struct pos{
 int y,x;
 pos(){}
 pos(int _y,int _x):y(_y),x(_x){}
 pos operator+(pos const& a){
  return pos(y+a.y,x+a.x);
 }
 pos operator-(pos const& a){
  int _y=y-a.y;
  int _x=x-a.x;
  return pos(_y,_x);
 }
 pos& operator+=(pos const& a){
  x+=a.x;y+=a.y;
  return (*this);
 }
 pos& operator-=(pos const& a){
  y-=a.y;
  x-=a.x;
  return *this;
 }
 pos& operator=(pos const& a){x=a.x;y=a.y;return (*this);}
 bool operator==(pos const& rhs)const{
  return y==rhs.y&&x==rhs.x;
 }
 bool operator!=(pos const& rhs)const{
  return !(y==rhs.y&&x==rhs.x);
 }
 bool operator<(pos const& rhs)const{
  return y<rhs.y;
 }
 bool operator>(pos const& rhs)const{
  return y>rhs.y;
 }
};
template<ll MOD=ll(1e9+7)>
ll mod_pow(ll n,ll p){
 ll ans=1LL;
 while(p){
  if(p&1)ans*=n;
  n*=n;
  p>>=1;
  ans%=MOD;
  n%=MOD;
 }
 return ans;
}
double rad_to_deg(double n){
  return n*180.0/PI;
}
double deg_to_rad(double n){
  return n*PI/180.0;
}

main(){
  cin.tie(0);
  ios::sync_with_stdio(false);
  cout<<fixed<<setprecision(10);


}
```

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

//http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=DSL_1_A&lang=jp

using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)

class Union_Find {
private:
  ll n;
  vector<ll> parent;

public:
  Union_Find(ll v){
    n=v;
    parent.resize(n,-1);
  }


  ll root(ll v){
    if(parent[v]<0)return v;
    return parent[v]=root(parent[v]);
  }
  ll size(ll v){
    return -parent[root(v)];
  }
  bool connect(ll a,ll b){
    a=root(a);
    b=root(b);
    if(a==b)return false;
    if(size(a)<size(b)) swap(a,b);
    parent[a]+=parent[b];
    parent[b]=a;
    return true;
  }
  bool same(ll a,ll b){
    return root(a)==root(b);
  }

};
```

```
void solve(){
  ll n,q;
  cin>>n>>q;
  Union_Find uni(n);
  rep(i,q){
    ll c,x,y;
    cin>>c>>x>>y;
    if(c){
      auto tmp = uni.same(x,y);
      cout<<(tmp)<<endl;
    }
    else{
      uni.connect(x,y);
    }
  }
}

main(){
  solve();
}
```

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

// https://onlinejudge.u-aizu.ac.jp/courses/library/5/GRL/1/GRL_1_C

using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)
constexpr long long INF = (1LL<<60);
template<class T>inline bool chmin(T &a,const T &b){if(a>b){a=b;return 1;}return 0;}

auto Warshall_Floyd(vector<vector<ll>>& dist){
 ll v = dist.size();
 rep(i,v)dist[i][i]=0LL;
 rep(i,v)rep(j,v)rep(k,v){
  if(dist[j][i]==INF||dist[i][k]==INF)continue;
  chmin(dist[j][k],dist[j][i]+dist[i][k]);
 }
 bool negative_cycle=false;
 rep(i,v)rep(j,v)rep(k,v){
  if(dist[j][i]==INF||dist[i][k]==INF)continue;
  if( chmin(dist[j][k],dist[j][i]+dist[i][k]) ){
   negative_cycle=true;
  }
 }
 return negative_cycle;
}

void solve(){


 ll v,e;
 cin>>v>>e;

 vector<vector<ll>> dist (v,vector<ll>(v,INF));

 rep(i,e){
  ll u,v,w;
  cin>>u>>v>>w;
  chmin(dist[u][v],w);
 }

 bool negative_cycle=Warshall_Floyd(dist);

 if(negative_cycle)puts("NEGATIVE CYCLE");
 else{
  rep(i,v){
   rep(j,v){
    if(dist[i][j]==INF)cout<<"INF"<<(j+1==v?"¥n":" ");
    else cout<<dist[i][j]<<(j+1==v?"¥n":" ");
   }
  }
 }

}

main(){
 solve();
}
```

```
#include<bits/stdc++.h>
using namespace std;
using ll = long long;
#define rep(i,n) for(ll i=0;i<(n);++i)

auto Z_algorithm(const string& s){
  ll n=s.size();
  vector<ll> z(n);
  z[0]=n;
  ll L=0,R=0;
  for(ll i=1;i<n;++i){
    if( i>R ){
      L=R=i;
      while( R<n && s[R-L]==s[R] )R++;
      z[i]=R-L;
      R--;
    }
    else{
      ll k=i-L;
      if( z[k]<R-i+1 )z[i]=z[k];
      else{
        L=i;
        while( R<n && s[R-L]==s[R] )R++;
        z[i]=R-L;
        R--;
      }
    }
  }
  return z;
}

main(){

  for(auto x:Z_algorythm("aababaaabaabaabbba"))cout<<x<<" ";
  cout<<endl;
  for(auto x:Z_algorythm("abcdabcdabcd"))cout<<x<<" ";
  cout<<endl;
  for(auto x:Z_algorythm("aaaaaaaaaaaaaa"))cout<<x<<" ";
  cout<<endl;
  for(auto x:Z_algorythm("abracadabra"))cout<<x<<" ";
  cout<<endl;
  for(auto x:Z_algorythm("ababa"))cout<<x<<" ";
  cout<<endl;
  for(auto x:Z_algorythm("abcabc"))cout<<x<<" ";
  cout<<endl;

}
```