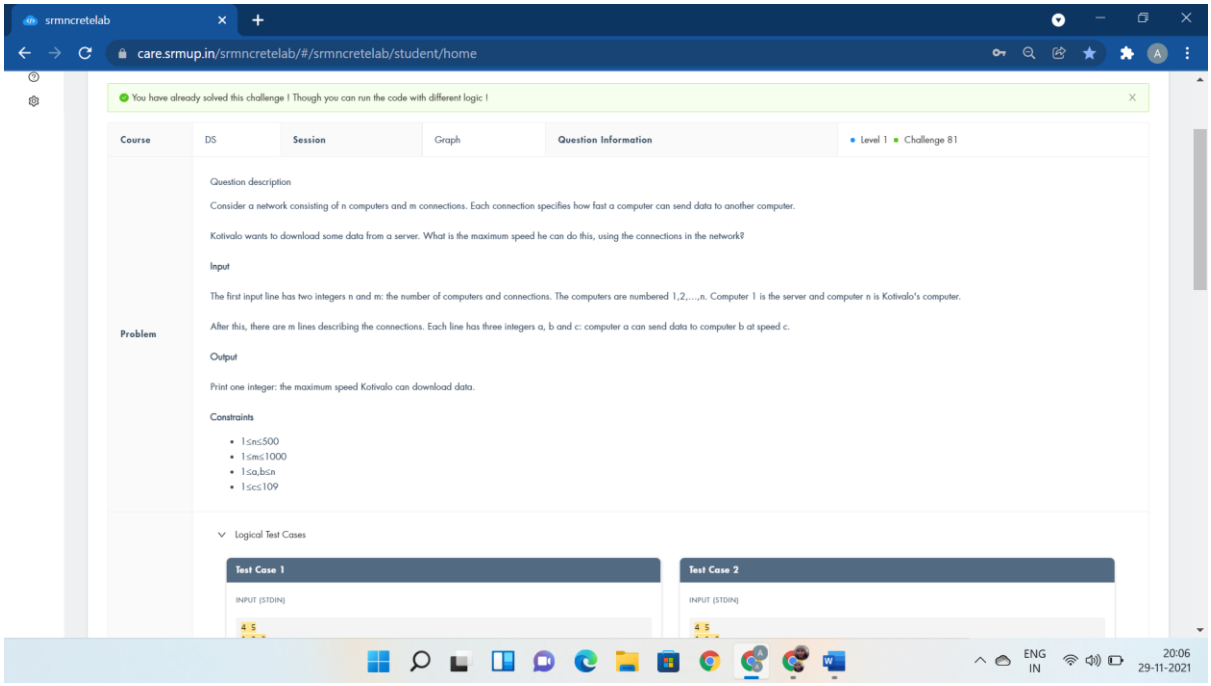
```
  for(i = 0; i < n; i++)
        {
            if (*(ptr + i) <= m)
            {
                continue;
            }

            for (j = i + 1; j < n; j++)
            {
                int temp = *(ptr + i) & *(ptr + j);
                if(temp > m)
                {
                    m = temp;
                }
            }
        }

        printf("%d", m);
}
```

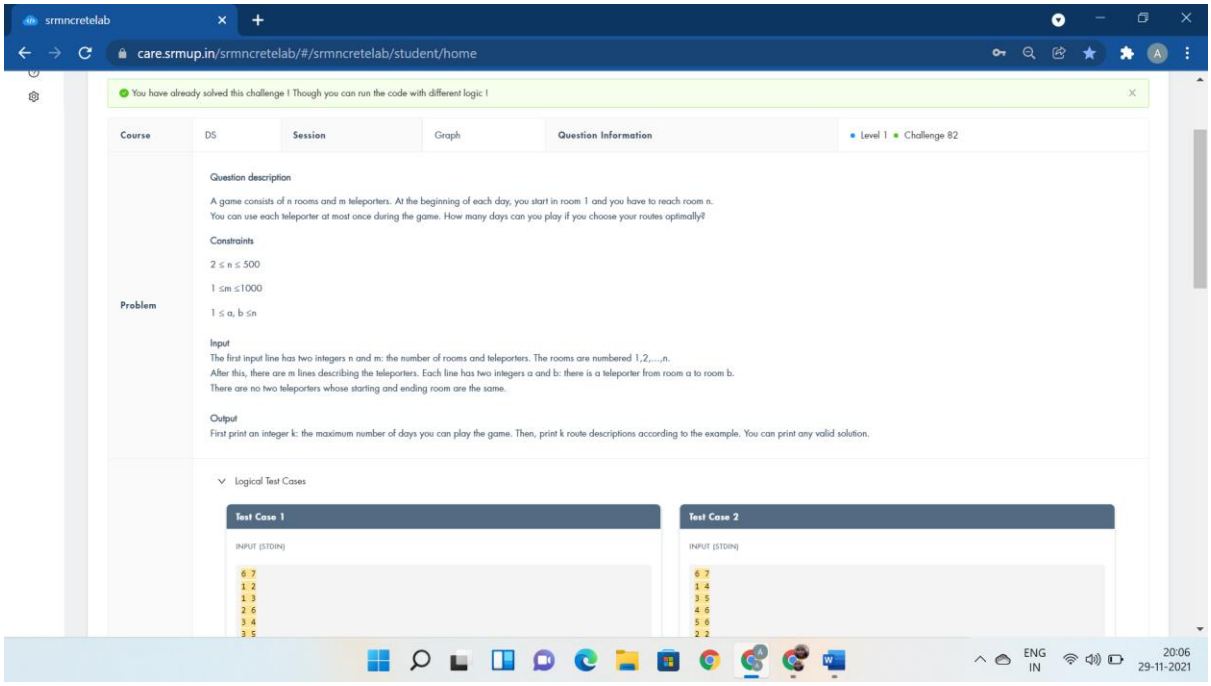
GRAPH:-




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

using namespace std;
```

```
using ll = long long;
#define F0R(i,a) for(int i=0; i<(a); i++)
#define FOR(i,a,b) for(int i=(a); i<=(b); i++)
int n, m;
ll adj[501][501], oadj[501][501];
ll flow[501];
bool V[501];
int pa[501];
void link(int i,int h){}
int bfs(int n,int s,int t){return 1;}
bool reachable() {
	memset(V, false, sizeof(V));
	queue<int> Q; Q.push(1); V[1]=1;
	while(!Q.empty()) {
		int i=Q.front(); Q.pop();
		FOR(j,1,n) if (adj[i][j] && !V[j])
			V[j]=1, pa[j]=i, Q.push(j);
	}
	return V[n];
}

int main() {
  bfs(1,1,1);
  link(1,1);
	cin >> n >> m;
	FOR(i,1,n) FOR(j,1,n) adj[i][j] = 0;
	F0R(i,m) {
		ll a,b,c; cin >> a >> b >> c;
		adj[a][b] += c;
	}
	int v, u;
	ll maxflow = 0;
	while(reachable()) {
		ll flow = 1e18;
		for (v=n; v!=1; v=pa[v]) {
			u = pa[v];
			flow = min(flow, adj[u][v]);
		}
```

```
                maxflow += flow;
                for (v=n; v!=1; v=pa[v]) {
                        u = pa[v];
                        adj[u][v] -= flow;
                        adj[v][u] += flow;
                }
        }
        cout << maxflow << '\n';
}
```

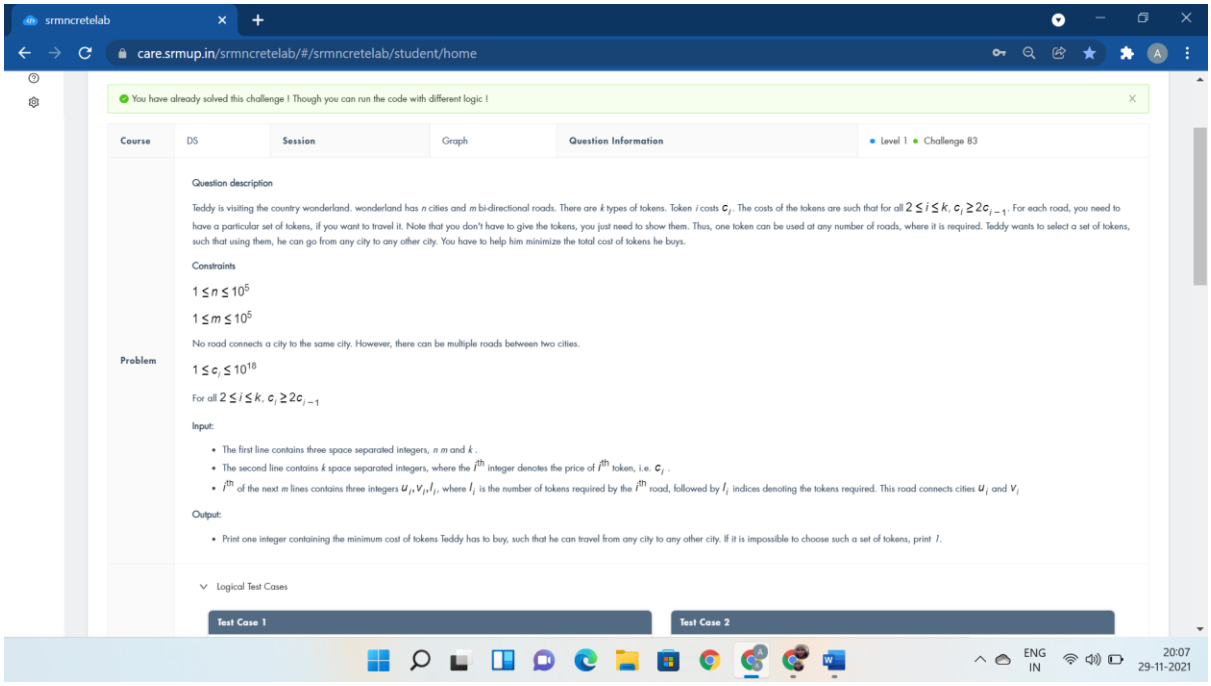



```
#include <stdio.h>

#define N    500
#define M    1000

struct L {
   struct L *next;
   int h;
} aa[N * 2];

int ij[M + N], cc[(M + N) * 2], dd[N * 2];

int bfs(int n,int s,int t) {
```

```
    static int qq[N * 2];
    int head, cnt, h, i, j, d;

    for (i = 0; i < n; i++)
        dd[i] = n;
    dd[s] = 0;
    head = cnt = 0;
    qq[head + cnt++] = s;
    while (cnt) {
        struct L *l;

        i = qq[cnt--, head++];
        d = dd[i] + 1;
        for (l = aa[i].next; l; l = l->next)
            if (cc[h = l->h]) {
                j = i ^ ij[h >> 1];
                if (dd[j] == n) {
                    dd[j] = d;
                    if (j == t)
                        return 1;
                    qq[head + cnt++] = j;
                }
            }
    }
    return 0;
}

int dfs(int n, int i, int t) {
    struct L *l;
    int h, j, d;

    if (i == t)
        return 1;
    d = dd[i] + 1;
    for (l = aa[i].next; l; l = l->next)
        if (cc[h = l->h]) {
            j = i ^ ij[h >> 1];
            if (dd[j] == d && dfs(n, j, t)) {
```

```
            cc[h]--, cc[h ^ 1]++;
            return 1;
          }
        }
    dd[i] = n;
    return 0;
}

int dinic(int n, int s, int t) {
    int f = 0;

    while (bfs(n, s, t))
        while (dfs(n, s, t))
            f++;
    return f;
}

void link(int i, int j, int h, int c) {
    static struct L l91[(M + N) * 2], *l = l91;

    ij[h] = i ^ j;
    cc[h << 1] = c;
    l->h = h << 1;
    l->next = aa[i].next, aa[i].next = l++;
    l->h = h << 1 ^ 1;
    l->next = aa[j].next, aa[j].next = l++;
}

int qq[N];

int path(int i, int t) {
    int cnt = 0;

    while (i != t) {
        struct L *l;
        int h;

        qq[cnt++] = i;
```

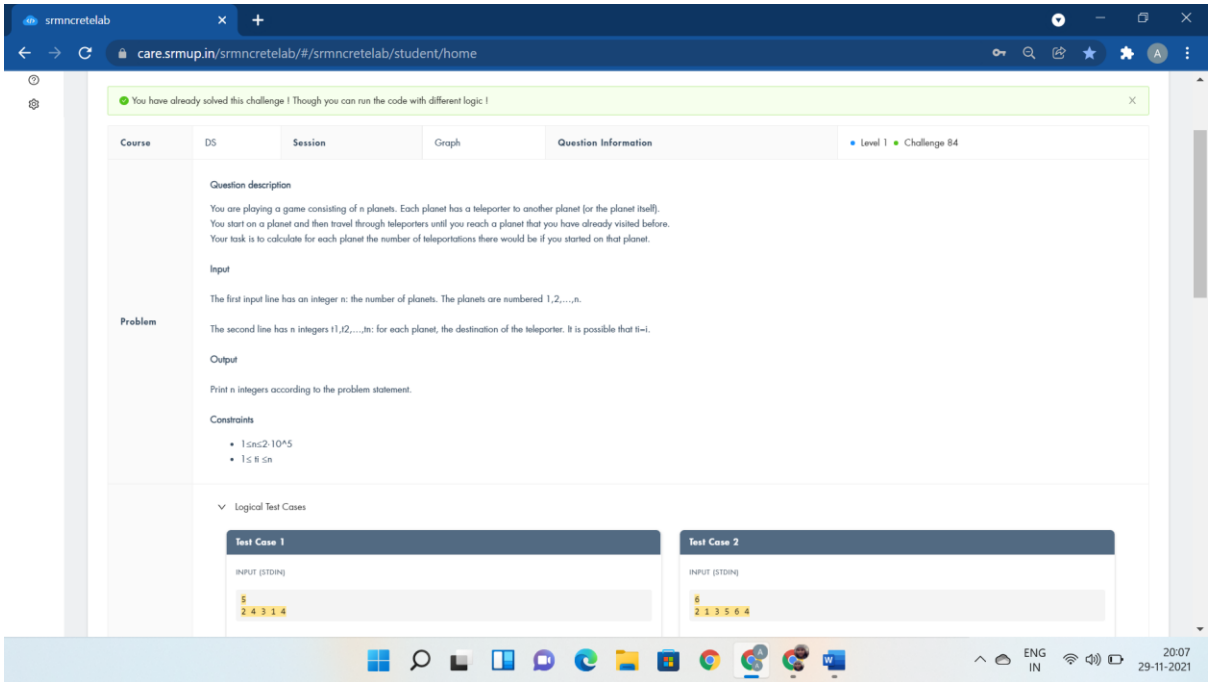
```
        for (l = aa[i].next; l; l = l->next)
            if (((h = l->h) & 1) == 0 && cc[h ^ 1]) {
                cc[h]++, cc[h ^ 1]--;
                i ^= ij[h >> 1];
                break;
            }
    }
    qq[cnt++] = t;
    return cnt;
}

int main() {
    int n, m, h, i, j, k, s, t, cnt;

    scanf("%d%d", &n, &m);
    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i << 1 ^ 1, j << 1, h, 1);
    }
    for (i = 0; i < n; i++)
        link(i << 1, i << 1 ^ 1, m + i, n);
    s = 0, t = (n - 1) << 1 ^ 1;
    k = dinic(n * 2, s, t);
    printf("%d\n", k);
    while (k--) {
        cnt = path(s, t);
        printf("%d\n", cnt / 2);
        for (i = 0; i < cnt; i += 2)
            printf("%d ", (qq[i] >> 1) + 1);
        printf("\n");
    }
    return 0;
}
```




```
#include <stdio.h>

#if defined( _WIN32 )
typedef __int64 az_int64_t;
typedef unsigned __int64 az_uint64_t;
#define I64(x) x ## I64
#define F64 "I64"
#else
typedef long long az_int64_t;
typedef unsigned long long az_uint64_t;
#define I64(x) x ## ll
#define F64 "ll"
#endif

#define MAXN (100*1024)

struct link
{
 az_int64_t t;
 int u, v;
};

struct link links[MAXN];
```

```
int n, m, k;
az_int64_t c[64];

int gr[MAXN];

int getgr( int g )
{
  return (g == gr[g]) ? g : (gr[g] = getgr( gr[g] ));
}

int test( az_int64_t r )
{
  int i, left = n-1, u, v;
  for(i=1;i<=n;++i) gr[i] = i;
  for( i = 0; i < m; ++i)
    if( (links[i].t & r) == 0 &&
        (u = getgr( links[i].u )) != (v = getgr( links[i].v )) )
    {
      gr[v] = u;
      if( --left == 0 ) return 1;
    }
  return 0;
}

int main( void )
{
  az_int64_t rejected = 0, sum = 0;
  int i;

  scanf( "%d %d %d", &n, &m, &k);
  for( i = 0; i < k; ++i) scanf( "%" F64 "d", &c[i]);
  for( i = 0; i < m; ++i)
  {
    int l, id;
    scanf( "%d %d %d", &links[i].u, &links[i].v, &l);
    while( l-- > 0 )
    {
      scanf( "%d", &id);
```

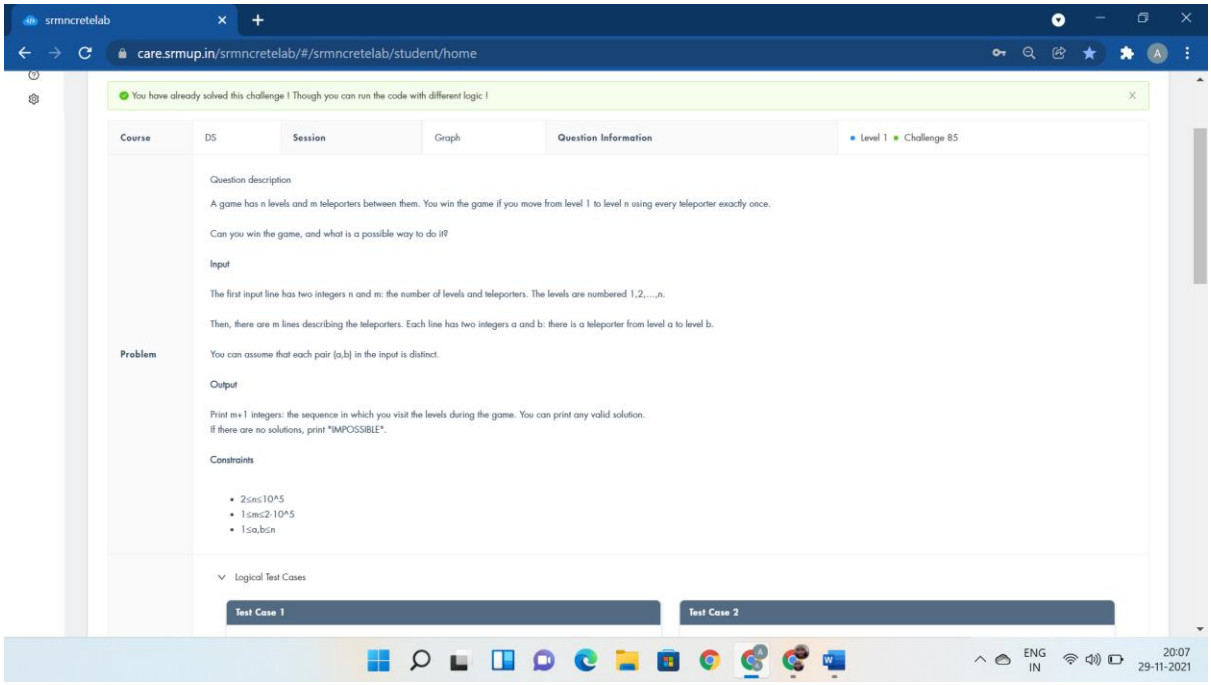
```
    links[i].t |= I64(1) << (id-1);
  }
 }

 if( !test( 0 ) )
 {
  printf( "-1\n" );
  return 0;
 }

 for( i = k-1; i >= 0; --i)
 {
  az_int64_t f = I64(1) << i;
  if( test( rejected | f ) ) rejected |= f; else sum += c[i];
 }

 printf( "%" F64 "d\n", sum);
 return 0;
}
```




```
#include <stdio.h>
#include <string.h>
```

```
#define N    200000


int main() {
   static int aa[N], cc[N], dd[N], qq[N];
   int n, i, j, c, d, q, cnt;


   scanf("%d", &n);
   for (i = 0; i < n; i++)
      scanf("%d", &aa[i]), aa[i]--;
   memset(cc, -1, n * sizeof *cc);
   cnt = 0;
   for(i = 0;i<n;i++) {
      if (cc[i] != -1)
         continue;
      d = 0;
      j = i;
      while (cc[j] == -1) {
         cc[j] = -2;
         d++;
         j = aa[j];
      }
      if (cc[j] == -2) {
         c = cnt++;
         q = 0;
         while (cc[j] == -2) {
            cc[j] = c;
            q++;
            j = aa[j];
         }
         qq[c] = q;
         d -= q;
      } else {
         c = cc[j];
         d += dd[j];
      }
      j = i;
      while (cc[j] == -2) {
```

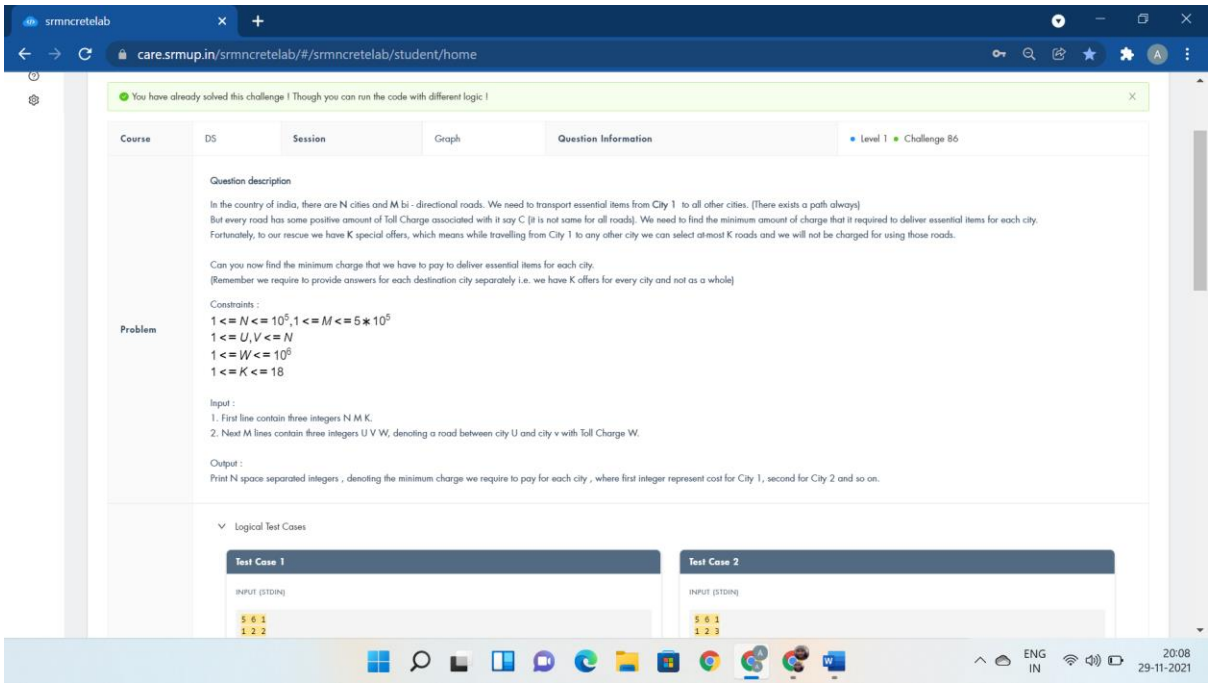
```
            cc[j] = c;
            dd[j] = d--;
            j = aa[j];
        }
    }
    for (i = 0; i < n; i++)
        printf("%d ", dd[i] + qq[cc[i]]);
    printf("\n");
    return 0;
}
```




```
#include <stdio.h>

#define N    100000
#define M    200000

struct L {
    struct L *next;
    int j;
} *aa[N];

struct L *new_L(int j) {
    static struct L l91[M + 1 + M], *l = l91;
```

```
    l->j = j;
    return l++;
}

void link(int i,int j) {
    struct L *l = new_L(j);

    l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e) {
    struct L *f = e->next, *l;
    int i = e->j;

    while ((l = aa[i])) {
        aa[i] = l->next;
        e = e->next = new_L(l->j);
        i = l->j;
    }
    e->next = f;
}

int main() {
    static int din[N], dout[N];
    struct L *e_, *e;
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);
    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
        dout[i]++, din[j]++;
    }
    if (dout[0] - din[0] != 1 || din[n - 1] - dout[n - 1] != 1) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
```

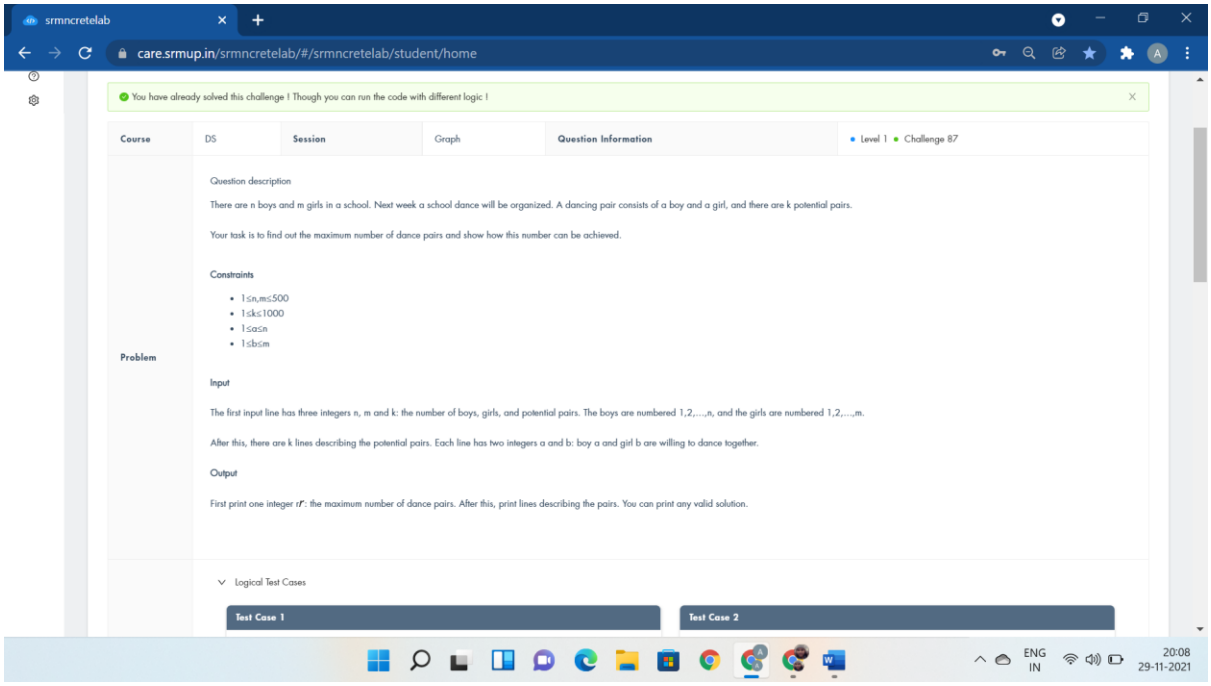
```
  for (i = 1; i < n - 1; i++)
    if (dout[i] != din[i]) {
      printf("IMPOSSIBLE\n");
      return 0;
    }
  e_ = new_L(0);
  m++;
  hierholzer(e_);
  for (e = e_; e; e = e->next) {
    hierholzer(e);
    m--;
  }
  if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
  }
  for (e = e_; e; e = e->next)
    printf("%d ", e->j + 1);
  printf("\n");
  return 0;
}
```




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

```
using namespace std;
void solve(){}
int main(){
   solve();
   long long int n,m;
                int k;
   cin>>n>>m>>k;
   vector<pair<long long int,long long int>> adjList[n+1];
   for(long long int i=0;i<m;++i){
      long long int a,b,c;
      cin>>a>>b>>c;
      /*if((a==1 && b==n) || (a==n && b==1)){
         cout<<"0\n";
         return 0;
      }*/
      adjList[a].push_back(pair<long long int,long long int>{b,c});
      adjList[b].push_back(pair<long long int,long long int>{a,c});
   }
   vector<vector<long long int>> dp(n+1,vector<long long int>(k+1,10000000000000));
   queue<pair<long long int,long long int>> q;
   dp[1][0]=0;
   q.push(pair<long long int,long long int>{0,1});
   while(!q.empty()){
      long long int from=q.front().first;
      long long int now=q.front().second;
      q.pop();
      bool change=false;
      for(auto to:adjList[now]){
         if(to.first==from){
            continue;
         }

                                          for(int i=0;i<=k;++i){
            if(i!=k && dp[to.first][i+1] > dp[now][i]){
               dp[to.first][i+1] = dp[now][i];
               change=true;
            }
         //for(int i=0;i<2;++i){
```

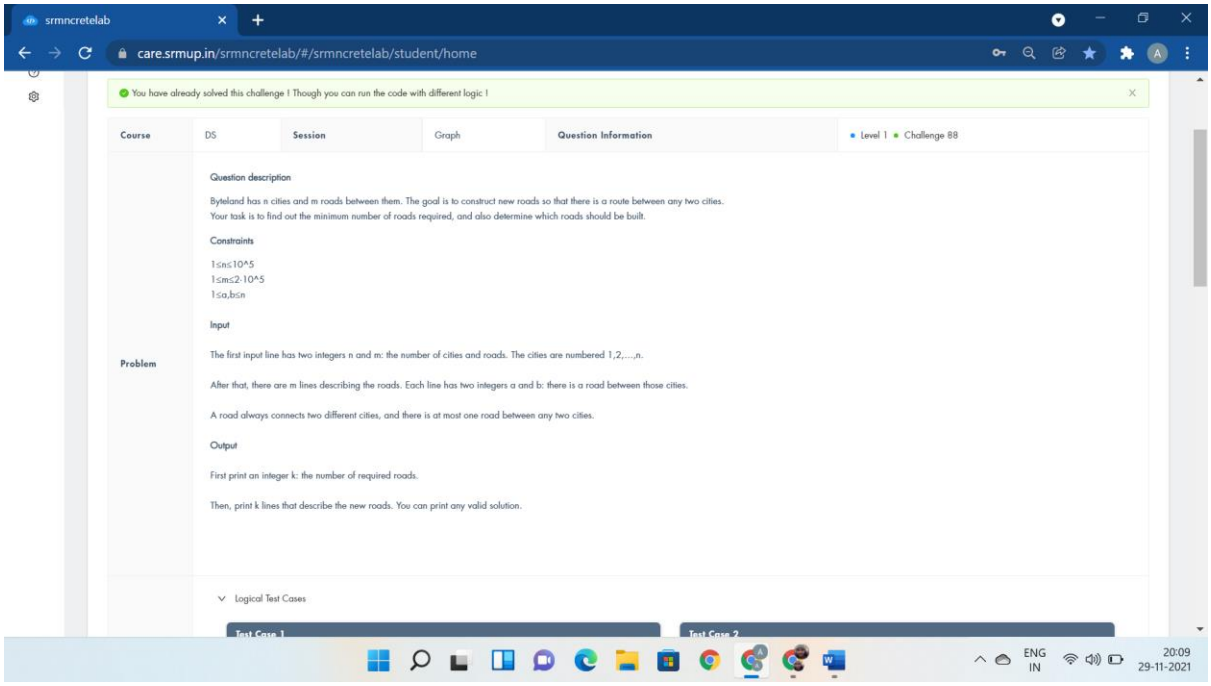
```
            if(dp[to.first][i] > dp[now][i]+to.second){
                dp[to.first][i] = dp[now][i]+to.second;
                change=true;
            }
          //}
                                          }
          if(change){
              q.push(pair<long long int,long long int>{now,to.first});
          }
       }
    }
    for(long long int i=1; i<=n; i++)
     {
        long long int ans = 10000000000000;
        for(long long int j =0; j<=k; j++)
        {
            ans = min(ans,dp[i][j]);

        }
        cout<<ans<<" ";
     }
    return 0;
}
```




```
#include <stdio.h>

#define N    500
#define M    1000

struct L {
   struct L *next;
   int v;
} aa[N + 1];

int vv[N + 1], uu[N + 1], dd[N + 1];

void link(int u,int v) {
   static struct L l91[M], *l = l91;

   l->v = v;
   l->next = aa[u].next, aa[u].next = l++;
}

int bfs(int n) {
   static int qq[N];
   int u, head, cnt, d;
```

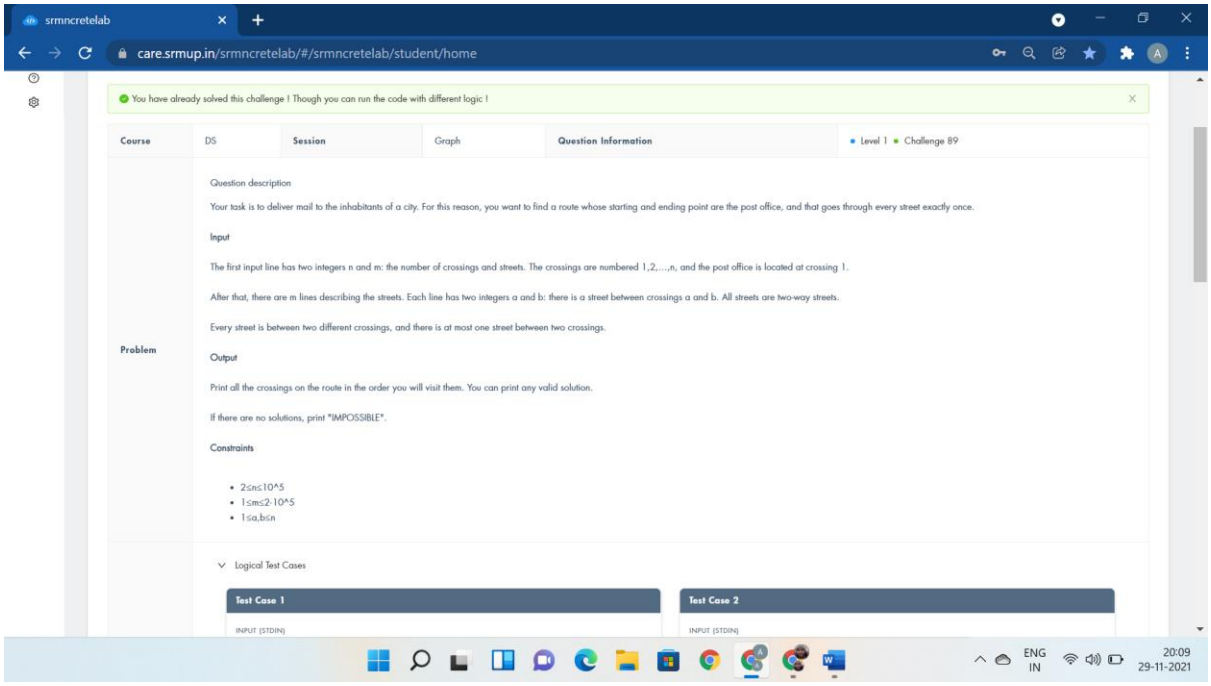
```
    head = cnt = 0;
    dd[0] = n;
    for (u = 1; u <= n; u++)
        if (vv[u] == 0) {
            dd[u] = 0;
            qq[head + cnt++] = u;
        } else
            dd[u] = n;
    while (cnt) {
        struct L *l;

        u = qq[cnt--, head++];
        d = dd[u] + 1;
        for (l = aa[u].next; l; l = l->next) {
            int v = l->v, w = uu[v];

            if (dd[w] == n) {
                dd[w] = d;
                if (w == 0)
                    return 1;
                qq[head + cnt++] = w;
            }
        }
    }
    return 0;
}

int dfs(int n, int u) {
    struct L *l;
    int d;

    if (u == 0)
        return 1;
    d = dd[u] + 1;
    for (l = aa[u].next; l; l = l->next) {
        int v = l->v, w = uu[v];

        if (dd[w] == d && dfs(n, w)) {
```

```
            vv[u] = v;
            uu[v] = u;
            return 1;
        }
    }
    dd[u] = n;
    return 0;
}

int hopcroft_karp(int n) {
    int m = 0;

    while (bfs(n)) {
        int u;

        for (u = 1; u <= n; u++)
            if (vv[u] == 0 && dfs(n, u))
                m++;
    }
    return m;
}

int main() {
    int n, n_, m, u, v;

    scanf("%d%d%d", &n, &n_, &m);
    while (m--) {
        scanf("%d%d", &u, &v);
        link(u, v);
    }
    printf("%d\n", hopcroft_karp(n));
    for (u = 1; u <= n; u++)
        if (vv[u])
            printf("%d %d\n", u, vv[u]);
    return 0;
}
```




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

using namespace std;


#define rep(i, a, b) for(int i = a; i < (b); ++i)

#define trav(a, x) for(auto& a : x)

#define all(x) begin(x), end(x)

#define sz(x) (int)(x).size()

typedef long long ll;

typedef pair<int, int> pii;

typedef vector<int> vi;


vi val, comp, z, cont;

int Time, ncomps;

template<class G, class F> int dfs(int j, G& g, F& f) {

  int low = val[j] = ++Time, x; z.push_back(j);

  trav(e,g[j]) if (comp[e] < 0)

    low = min(low, val[e] ?: dfs(e,g,f));


  if (low == val[j]) {

    do {

      x = z.back(); z.pop_back();

      comp[x] = ncomps;

      cont.push_back(x);
```

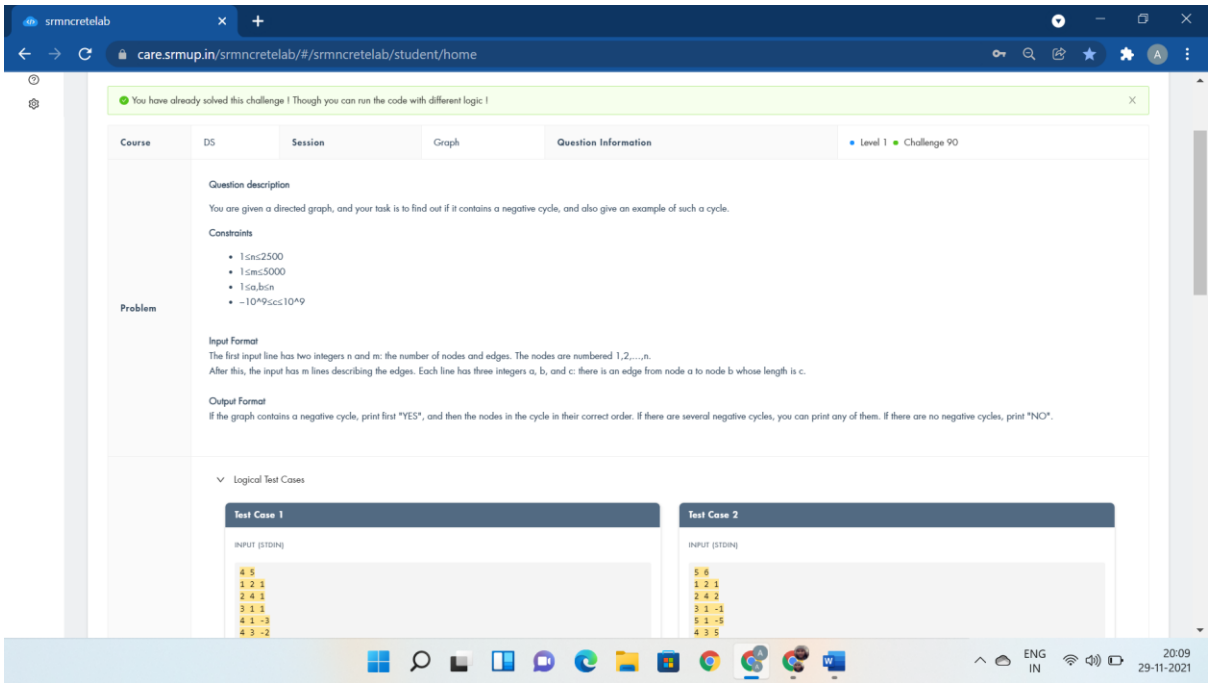
```
        } while (x != j);
        f(cont); cont.clear();
        ncomps++;
    }
    return val[j] = low;
}
template<class G, class F> void scc(G& g, F f) {
    int n = sz(g);
    val.assign(n, 0); comp.assign(n, -1);
    Time = ncomps = 0;
    rep(i,0,n) if (comp[i] < 0) dfs(i, g, f);
}

int main() {
    cin.sync_with_stdio(0); cin.tie(0);
    cin.exceptions(cin.failbit);

    int n, m;
    cin >> n >> m;
    vector<vi> g(n);
    while(m--) {
        int a, b;
        cin >> a >> b;
        a--, b--;
        g[a].push_back(b);
        g[b].push_back(a);
    }
    vi r;
    scc(g, [&](vi &c) { r.push_back(c[0]); });
    cout << sz(r)-1 << '\n';
    rep(i, 1, sz(r))
    cout << r[0]+1 << " " << r[i]+1 << '\n';
    return 0;
}
```




```
#include <stdio.h>

#define N    100000
#define M    200000

struct L {
   struct L *next;
   int h;
} *aa[N];

int ij[M + 1];
char lazy[M + 1];

struct L *new_L(int h) {
   static struct L l91[M * 2 + 1 + M], *l = l91;

   l->h = h;
   return l++;
}

void link(int i,int h) {
   struct L *l = new_L(h);
```

```
    l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e, int i) {
    struct L *f = e->next, *l;

    while ((l = aa[i])) {
        int h = l->h;

        if (lazy[h])
            aa[i] = l->next;
        else {
            lazy[h] = 1;
            e = e->next = new_L(h);
            i ^= ij[h];
        }
    }
    e->next = f;
}

int main() {
    static int dd[N];
    struct L *e_, *e;
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);
    for (h = 1; h <= m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        ij[h] = i ^ j;
        link(i, h), link(j, h);
        dd[i]++, dd[j]++;
    }
    for (i = 0; i < n; i++)
        if (dd[i] % 2) {
            printf("IMPOSSIBLE\n");
            return 0;
        }
    e_ = new_L(0);
```

```
    i = 0;
    m++;
    for (e = e_; e; e = e->next) {
        i ^= ij[e->h];
        hierholzer(e, i);
        m--;
    }
    if (m != 0) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
    i = 0;
    for (e = e_; e; e = e->next) {
        i ^= ij[e->h];
        printf("%d ", i + 1);
    }
    printf("\n");
    return 0;
}
```

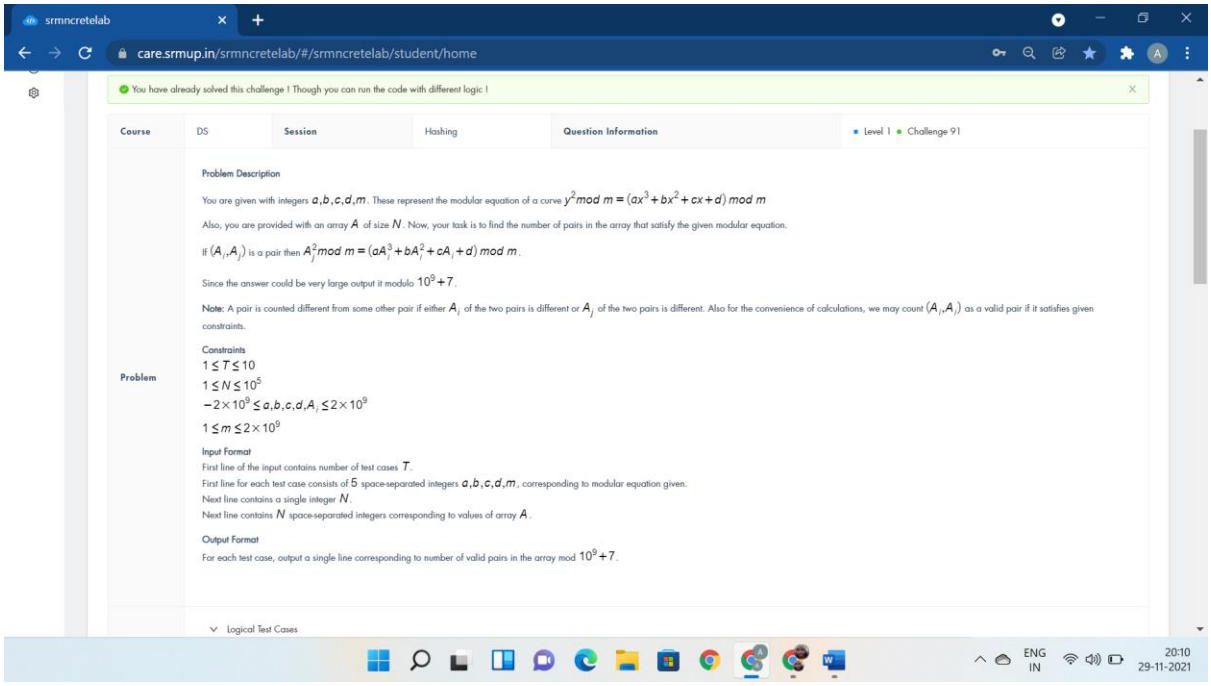



```
#include <stdio.h>

#define N  2500
```

```
#define M    5000


int main() {
   static int aa[M], bb[M], cc[M], pp[N], ii[1 + N];
   static char used[N];
   static long long dd[N];
   int n, m, h, r, a, b, c, k;
   long long d;


   scanf("%d%d", &n, &m);
   for (h = 0; h < m; h++)
      scanf("%d%d%d", &aa[h], &bb[h], &cc[h]), aa[h]--, bb[h]--;
   for (r = 0; r < n; r++)
      for (h = 0; h < m; h++) {
         a = aa[h], b = bb[h], c = cc[h];
         d = dd[a] + c;
         if (dd[b] > d) {
            dd[b] = d;
            pp[b] = a;
            if (r == n - 1) {
               while (!used[b]) {
                  used[b] = 1;
                  b = pp[b];
               }
               k = 0;
               while (used[b]) {
                  used[b] = 0;
                  ii[k++] = b;
                  b = pp[b];
               }
               ii[k++] = b;
               printf("YES\n");
               while(k--)
                  printf("%d ", ii[k] + 1);
               printf("\n");
               return 0;
            }
         }
```

```
    }
  printf("NO\n");
  return 0;
}
```

HASHING :-




```
#include <bits/stdc++.h>
using namespace std;
const int md = 1E9 + 7;
map<long long, int> mp;

int main() {
  int t;
  cin >> t;
  while(t--) {
    long long a, b, c, d, m;
    cin >> a >> b >> c >> d >> m;
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0; i < n; i ++) {
      cin >> arr[i];
      mp[(((arr[i] * arr[i]) % m) + m) % m] ++;
```