Answer Script

Question No. 01

Problem - 1

➔ Write C++ program to solve The Lakes by using dfs. **10**

Answer No. 01

Code:

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

vector<int> a[1005];
bool visited[1005][1005];

int dx[4] = {1, 0, 0, -1};
int dy[4] = {0, -1, 1, 0};

bool isValid(int x, int y, int n, int m)
{
   return (x>-1 and x<n and y>-1 and y<m and !visited[x][y] and a[x][y]!=0);
}

int DFS(int x, int y, int n, int m)
{
   int res = a[x][y];
   visited[x][y] = true;

   for(int i=0; i<4; i++)
   {
      int tx = x + dx[i];
      int ty = y + dy[i];

      if(isValid(tx, ty, n, m))
      {
         res += DFS(tx, ty, n, m);
      }
   }

   return res;
}
```

```
int main()
{
   int t;
   cin>>t;
   while(t--)
   {
      int n, m;
      cin>>n>>m;

      for(int i=0; i<n; i++)
      {
         a[i].assign(m, 0);
         for(int j=0; j<m; j++)
         {
            cin>>a[i][j];
            visited[i][j] = false;
         }
      }

      int ans = 0;

      for(int i=0; i<n; i++)
      {
         for(int j=0; j<m; j++)
         {
            if(visited[i][j] or a[i][j]==0)
               continue;
            ans = max(ans, DFS(i, j, n, m));
         }
      }

      cout<<ans<<"\n";
   }

   return 0;
}
```

Question No. 02

Problem - 2

➔ Write C++ program to solve Network Delay Time by using dijkstra.

10

Answer No. 02

Code:

```
class Solution {
public:
    int networkDelayTime(vector<vector<int>>& times, int n, int k) {
        vector<int> v(n+1, -1);
        unordered_map<int, vector<pair<int, int>>> mp;

        for(auto t: times)
        {
            mp[t[0]].push_back({t[1], t[2]});
        }

        int lng = 0;
        int cnt = 1;
        v[k] = 0;

        queue<int> q;
        q.push(k);

        while(!q.empty())
        {
            int src = q.front();
            q.pop();

            for(auto p: mp[src])
            {
                if(v[p.first] == -1 || v[p.first] > p.second + v[src])
                {
                    if(v[p.first] == -1)
                    {
                        cnt++;
                    }
                    v[p.first] = p.second + v[src];
                    q.push(p.first);
                }
            }
        }

        if(cnt != n)
        {
            return -1;
        }

        for(int i=1; i<v.size(); i++){
            lng = max(lng, v[i]);
        }

        return lng;
    }
};
```

Question No. 03

Problem - 3

➔ Write C++ program to solve Coin Combinations I by using the tabulation method. **10**

Answer No. 03

Code:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
const ll MOD = 1e9 + 7;

int main()
{
    int n, x;
    cin>>n>>x;

    vector<ll> coins(n, 0);
    for(int i=0; i<n; i++)
        cin>>coins[i];

    vector<ll> nums(x+1, 0);
    nums[0] = 1;
    ll sum;

    for(int i=1; i<=x; i++)
    {
        sum = 0;
        for(int j=0; j<n; j++)
        {
            if(i-coins[j] >= 0)
            {
                sum += nums[i-coins[j]];
            }
        }
        nums[i] = sum % MOD;
    }
    cout<<nums[x]<<"\n";
    return 0;
}
```

Question No. 04

Problem - 4

➔ Write C++ program to solve Sum of Two Values **10**

Answer No. 04

Code:

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

typedef pair<int,int> pii;
const int M = 2e5 + 1;
pii p[M];
int n, x;

int findTarget(int val)
{
    int l = 1;
    int r = n;

    while(l<=r)
    {
        int m = l + (r-l) / 2;

        if(p[m].first == val)
            return p[m].second;
        else if(p[m].first<val)
            l = m + 1;
        else
            r = m - 1;
    }
    return -1;
}

int main()
{
    cin>>n>>x;

    for(int i=1; i<=n; i++)
    {
        int a;
        cin>>a;
        p[i] = {a, i};
    }

    sort(p+1, p+n+1);

    for(int i=1; i<=n; i++)
    {
        int y = findTarget(x - p[i].first);

        if(y != -1 && y != p[i].second)
        {
            cout<<y<<" "<<p[i].second<<"\n";
            return 0;
        }
    }

    cout<<"IMPOSSIBLE\n";

    return 0;
}
```

Question No. 05

Problem - 5

➔ Write C++ program to solve Money Sums by using the tabulation method. **15**

Answer No. 05

Code:

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

const int N = 1e5;
bool dp[101][N+1];

int main()
{
    int n;
    cin>>n;

    vector<int> coins(n);
    for(int i=0; i<n; i++)
    {
        cin>>coins[i];
    }

    dp[0][0] = true;

    for(int i=1; i<=n; i++)
    {
        for(int j=0; j<=N; j++)
        {
            dp[i][j] = dp[i-1][j];
            int sum = j - coins[i-1];

            if(sum >= 0 && dp[i-1][sum])
            {
                dp[i][j] = true;
            }
        }
    }
    vector<int> v;

    for(int i=1; i<=N; i++)
    {
        if(dp[n][i])
        {
            v.push_back(i);
        }
    }

    cout<<v.size()<<"\n";
    for(int x: v)
    {
        cout<<x<<" ";
    }
    cout<<"\n";

    return 0;
}
```

Question No. 06

Problem - 6

➔ Write C++ program to solve Back to Underworld **15**

Answer No. 06

Code:

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

const int N = 20009;
vector<int> adj_list[N];
int num[N];
int vamp = 0, lyk = 0;

void BFS(int src)
{
   queue<int> q;
   q.push(src);

   num[src] = 1;
   vamp++;

   while(!q.empty())
   {
      int head = q.front();
      q.pop();

      for(int i=0; i<adj_list[head].size(); i++)
      {
         int x = adj_list[head][i];

         if(num[x] == 0)
         {
            q.push(x);

            if(num[head] == 1)
            {
               num[x] = 2;
               lyk++;
            }
            else
```

```
                {
                    num[x] = 1;
                    vamp++;
                }
            }
        }
    }

}

int main()
{
    int t, ans;
    cin>>t;

    for(int i=1; i<=t; i++)
    {
        for(int j=0; j<N; j++)
        {
            adj_list[j].clear();
        }

        memset(num, 0, sizeof num);

        int n;
        cin>>n;

        vamp = 0;
        lyk = 0;
        ans = 0;

        for(int j=0; j<n; j++)
        {
            int u, v;
            cin>>u>>v;

            adj_list[u].push_back(v);
            adj_list[v].push_back(u);
        }

        for(int k=0; k<N; k++)
        {
            if(!adj_list[k].empty() && num[k] == 0)
            {
                vamp = 0;
        lyk = 0;

        BFS(k);

        ans += max(vamp, lyk);
      }
    }

    cout<<"Case "<<i<<": "<<ans<<"\n";
  }
  return 0;
}
```

Question No. 07

Problem - 7

- ➔ Write C++ program to solve Required Length. **15**

Answer No. 07

Code:

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

#define ll long long
const int N = 1e9;
map<ll, ll> mp;

ll solve(ll n, ll x)
{
    string a = to_string(x);
    if(a.size() == n)
    {
        return 0;
    }

    if(mp.count(x)>0)
    {
        return mp[x];
    }

    mp[x] = N;
    ll p = x;
    while(p>0)
    {
        if(p%10 > 1)
        {
            mp[x] = min(mp[x], solve(n, x*(p%10))+1);
        }
        p /= 10;
    }

    return mp[x];
}

int main()
```

```
{
    ll n, x;
    cin>>n>>x;

    ll ans = solve(n, x);

    if(ans == N)
        cout<<-1<<"\n";
    else
        cout<<ans<<"\n";

    return 0;
}
```

Question No. 08

Problem - 8

- ➔ Write C++ program to solve LCS by using the memoization method. **15**

Answer No. 08

Code:

```
#include <bits/stdc++.h>
using namespace std;
const int N = 3003;
#define ll long long

ll dp[N][N];
ll n, m;
string s, t;

ll LCS(ll i, ll j)
{
   if(i == 0 || j == 0)
   {
      return 0;
   }

   if(dp[i][j] != -1)
   {
      return dp[i][j];
   }

   if(s[i-1]==t[j-1])
   {
      dp[i][j] = 1 + LCS(i-1, j-1);
   }
   else
   {
      dp[i][j] = max(LCS(i-1, j), LCS(i, j-1));
   }

   return dp[i][j];
}
```

```
int main()
{
    cin>>s>>t;

    n = s.size();
    m = t.size();

    for(ll i=0; i<=n; i++)
    {
        for(ll j=0; j<=m; j++)
        {
            dp[i][j] = -1;
        }
    }

//    cout<<LCS(0, 0)<<"\n";
    ll res = LCS(n, m);

    string ans;
    ll x = n;
    ll y = m;
    while(x && y)
    {
        if(s[x-1] == t[y-1])
        {
            ans += s[x-1];
            x--;
            y--;
        }
        else if(dp[x][y-1] >= dp[x-1][y])
        {
            y--;
        }
        else
        {
            x--;
        }
    }
    reverse(ans.begin(), ans.end());
    cout<<ans<<"\n";

    return 0;
}
```