PRACTICAL EXAM

U20CS005
BANSI MARAKANA

QUESTION 1:

```
indication(fever).
indication(rash).
indication(headache).
indication(runny_nose).
indication(conjunctivitis).
indication(cough).
indication(body_ache).
indication(chills).
indication(sore_throat).
indication(sneezing).

name('Abc').
name('Def').
name('Ghi').

disease('Flu',[fever, headache, body_ache, conjunctivitis, chills, sore_throat, runny_nose,
cough]).
disease('Common_cold',[headache, sneezing, sore_throat, runny_nose, chills]).
disease('Chicken_pox',[fever, chills, body_ache, rash]).
disease('Measles',[cough, sneezing, runny_nose]).

patient(1, 'Patient 1', 'ABC Apartment, Delhi - 110001', [treatment('Doctor D1', 'Flu'),
treatment('Doctor D2', 'Measles')]).
patient(2, 'Patient 2', 'XYZ Building, Gujarat - 400069', [treatment('Doctor D2', 'Common cold')]).
patient(3, 'Patient 3', 'PQR Society, Delhi - 110029', [treatment('Doctor D1', 'Flu'),
treatment('Doctor D3', 'Chicken pox'), treatment('Doctor D4', 'Measles')]).

symptom(Name, Indication) :- write('Does '), write(Name), write(' have a '),
   write(Indication), write('? (y/n)'), nl,
   response(Reply), Reply='y'.

hypothesis(Name, Disease) :- disease(Disease, Indications),
   forall(member(I, Indications), symptom(Name, I)).

%1
total_diseases(P_Id, Count) :-
   patient(P_Id, _, _, Treatments),
   findall(Disease, member(treatment(_, Disease), Treatments), Diseases),
   length(Diseases, Count).
```

```
%2
name_and_zipcode(P_Id, Name, Zipcode) :-
    patient(P_Id, Name, Address, _),
    atom_codes(Address, Codes),
    reverse(Codes, RCodes),
    phrase(postal(Zip), RCodes),
    reverse(Zip, RZip),
    atom_codes(Zipcode, RZip).

%3
delhi_patients(P_Id, Name):- patient(P_Id, Name, Address, _),sub_atom(Address, _, _, _,
'Delhi').

%4.
doctor_patients('Doctor D1', Patient) :-
    patient(_, Patient, _, Treatments),
    member(treatment('Doctor D1', _), Treatments).

%5
common_cold_patients(P_Id) :-
    patient(P_Id, _, _, Treatments),
    member(treatment(_, 'Common cold'), Treatments).

%6
patient_details(P_Id, Details) :-
    patient(P_Id, _, Address, _),
    sub_atom(Address, B, _, A, ','),
    sub_atom(Address, _, A, 0, CityZip),
    atom_number(CityZip, Zip),
    atom_codes(Building, Address, 0, B),
    tuple(Building, Zip, Details).

%7
patient_doctors(P_Id, Doctors) :-
    patient(P_Id, _, _, Treatments),
    findall(Doc, member(treatment(Doc, _), Treatments), Doctors).
```

total_diseases(P_Id,Count)

P_Id	Count
1	2
2	1
3	3

patient_doctors(P_Id, Doctors)

P_Id	Doctors
1	['Doctor D1', 'Doctor D2']
2	['Doctor D2']
3	['Doctor D1', 'Doctor D3', 'Doctor D4']

doctor_patients('Doctor D1', Patient)

Patient
'Patient 1'
'Patient 3'

common_cold_patients(P_Id)

P_Id
2

false

delhi_patients(P_Id, Name)

P_Id	Name
1	'Patient 1'
3	'Patient 3'

QUESTION 2:

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

const int N = 20;
int n, tsp[N][N], seen[1 << N][N], ans = INT_MAX;
struct node
{
    int removed, current, cost;
};

void bfs()
{
    queue<node> q;
    q.push({1, 0, 0});
    seen[1][0] = 1;
    while (!q.empty())
    {
        node nd = q.front();
        q.pop();
        if (nd.removed == (1 << n) - 1 && tsp[nd.current][0])
            ans = min(ans, nd.cost + tsp[nd.current][0]);
        for (int i = 0; i < n; i++)
        {
            if (!seen[nd.removed | (1 << i)][i] && tsp[nd.current][i])
            {
                seen[nd.removed | (1 << i)][i] = 1;
                q.push({nd.removed | (1 << i), i, nd.cost +
tsp[nd.current][i]});
            }
        }
    }
}

int main()
{
    cout << "\nTSP using BFS\n";
    cout << "Enter the number of nodes: ";
    cin >> n;
```

```
    cout << "Enter adjacency matrix: \n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            cin >> tsp[i][j];
    bfs();
    if (ans == INT_MAX)
        cout << "Cycle doesn't exist!!\n";
    else
        cout << "Minimum cost is: " << ans << "\n";
    return 0;
}
```

```
TSP using BFS
Enter the number of nodes: 3
Enter adjacency matrix:
1 0 1
1 1 1
1 1 1
Minimum cost is: 3
```