U18CO018
Shubham Shekhaliya

1->

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

int main() {
    cout << "Enter the number of model in tables: \n";
    map<string, pair<int, int> > mp;
    int n;
    cin >> n;
    while (n--) {
        string name;
        cout << "Enter the model name: ";
        cin >> name;
        int total_unit, cost;
        cout << "Enter the total number of units sold: ";
        cin >> total_unit;
        cout << "Enter the cost of model: ";
        cin >> cost;
        mp[name] = {total_unit, cost};
    }
    while (true) {
        cout << "1 Enter a model name\n2 Exit\n";
        int choice;
        cin >> choice;
        if (choice == 1) {
            cout << "Enter model name: ";
            string name;
            cin >> name;
            if (mp.find(name) == mp.end()) {
                cout << "Model not found !!\n";
            } else {
                pair<int, int> p = mp[name];
                cout << "total units sold: " << p.first << endl;
                cout << "cost of model: " << p.second << endl;
                cout << "total cost: " << p.first * p.second << endl;
            }
        } else {
            break;
        }
    }
```

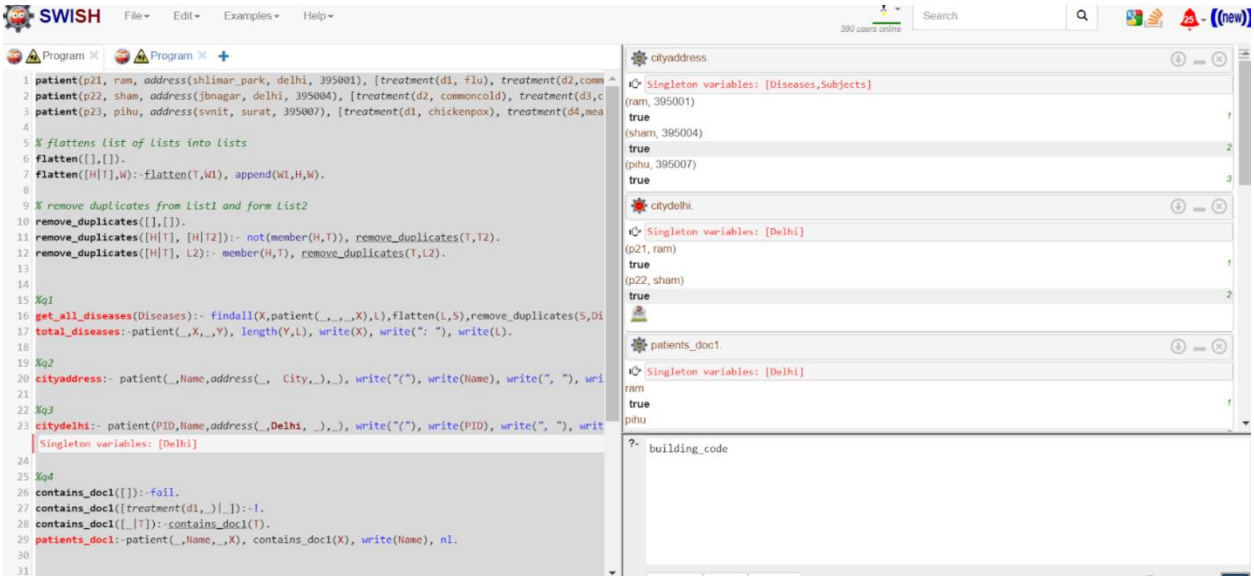
```
    return 0;
}
```

```
PS D:\Course-Work\7th SEM> cd "d:\Course-Work\7th SEM\PPL\Practical\" ; if ($?) { g++ code.cpp -o code } ; if ($?) { .\c
Enter the number of model in tables:
2
Enter the model name: m1
Enter the total number of units sold: 5
Enter the cost of model: 20
Enter the model name: m2
Enter the total number of units sold: 5
Enter the cost of model: 10
1 Enter a model name
2 Exit
1
Enter model name: m3
Model not found !!
1 Enter a model name
2 Exit
1
Enter model name: m2
total units sold: 5
cost of model: 10
total cost: 50
1 Enter a model name
2 Exit
2
PS D:\Course-Work\7th SEM\PPL\Practical>
```

2->

```
patient(p21, shubham, address(shlimar_park, delhi, 395001), [sub(d1, flu),
sub(d2,commoncold)] ).
patient(p22, darshan, address(jbnagar, delhi, 395004), [sub(d2, commoncold),
sub(d3,chickenpox)] ).
patient(p23, sagar, address(svnit, surat, 395007), [sub(d1, chickenpox),
sub(d4,measles)] ).

% flattens list of lists into lists
flatten([],[]).
flatten([H|T],W):-flatten(T,W1), append(W1,H,W).

% remove duplicates from List1 and form List2
remove_duplicates([],[]).
remove_duplicates([H|T], [H|T2]):- not(member(H,T)), remove_duplicates(T,T2).
remove_duplicates([H|T], L2):- member(H,T), remove_duplicates(T,L2).

%q1
total_diseases:-patient(_,X,_,Y), length(Y,L), write(X), write(": "), write(L).

%q2
name_and_zip:-patient(_,X,address(_,_,Zip),_),write(X), write(" has zip code: "),
write(Zip).
```

```
%q3
citydelhi:- patient(PID,Name,address(_,delhi, _),_), write(PID), write(" has a
name "), write(Name), nl.

%q4
contains_doc1([]):-fail.
contains_doc1([treatment(d1,_)|_]):-!.
contains_doc1([_|T]):-contains_doc1(T).

patients_doc1:-patient(_,Name,_,X), contains_doc1(X), write(Name), nl.

%q5
contains_cold([]):-fail.
contains_cold([sub(_,commoncold)|_]):-!.
contains_cold([_|T]):-contains_cold(T).

common_cold_patient:- patient(id,_,_,X), contains_cold(X), write(id), nl.

%q6
cityaddress:- patient(_,_,address(Building, _, Code),_), write("("),
write(Building),
write(", "), write(Code), write(")"), nl.

%q7
extract_doctor([],[]).
extract_doctor([treatment(T,_)|Rest], [T|Tail]):-extract_doctor(Rest,Tail).
doctor_for_patients:- patient(_,Name,_,Y),extract_doctor(Y,Z), write(Name),
write(": "), write(Z), nl.
```


SWISH
File
Edit
Examples
Help
390 users online
Search
Program
Program
patient(p21, ram, address(shlimar_park, delhi, 395001), [treatment(d1, flu), treatment(d2,comm
patient(p22, sham, address(jbnagar, delhi, 395004), [treatment(d2, commoncold), treatment(d3,c
patient(p23, pihu, address(svnit, surat, 395007), [treatment(d1, chickenpox), treatment(d4,mea
% flattens list of lists into lists
flatten([],[]).
flatten([H|T],W):-flatten(T,W1), append(W1,H,W).
% remove duplicates from list1 and form list2
remove_duplicates([],[]).
remove_duplicates([H|T], [H|T2]):- not(member(H,T)), remove_duplicates(T,T2).
remove_duplicates([H|T], L2):- member(H,T), remove_duplicates(T,L2).
%q1
get_all_diseases(Diseases):- findall(X,patient(_,_,_,X),L),flatten(L,S),remove_duplicates(S,Di
total_diseases:-patient(_,X,_,Y), length(Y,L), write(X), write(": "), write(L).
%q2
cityaddress:- patient(_,Name,address(_, City,_),_), write("("), write(Name), write(", "), wri
%q3
citydelhi:- patient(PID,Name,address(_,Delhi, _),_), write("("), write(PID), write(", "), writ
Singleton variables: [Delhi]
%q4
contains_doc1([]):-fail.
contains_doc1([treatment(d1,_)|_]):-!.
contains_doc1([_|T]):-contains_doc1(T).
patients_doc1:-patient(_,Name,_,X), contains_doc1(X), write(Name), nl.
cityaddress.
Singleton variables: [Diseases,Subjects]
(ram, 395001)
true
(sham, 395004)
true
(pihu, 395007)
true
citydelhi.
Singleton variables: [Delhi]
(p21, ram)
true
(p22, sham)
true
patients_doc1.
Singleton variables: [Delhi]
ram
true
pihu
?-
building_code