Chintu’s Mansion

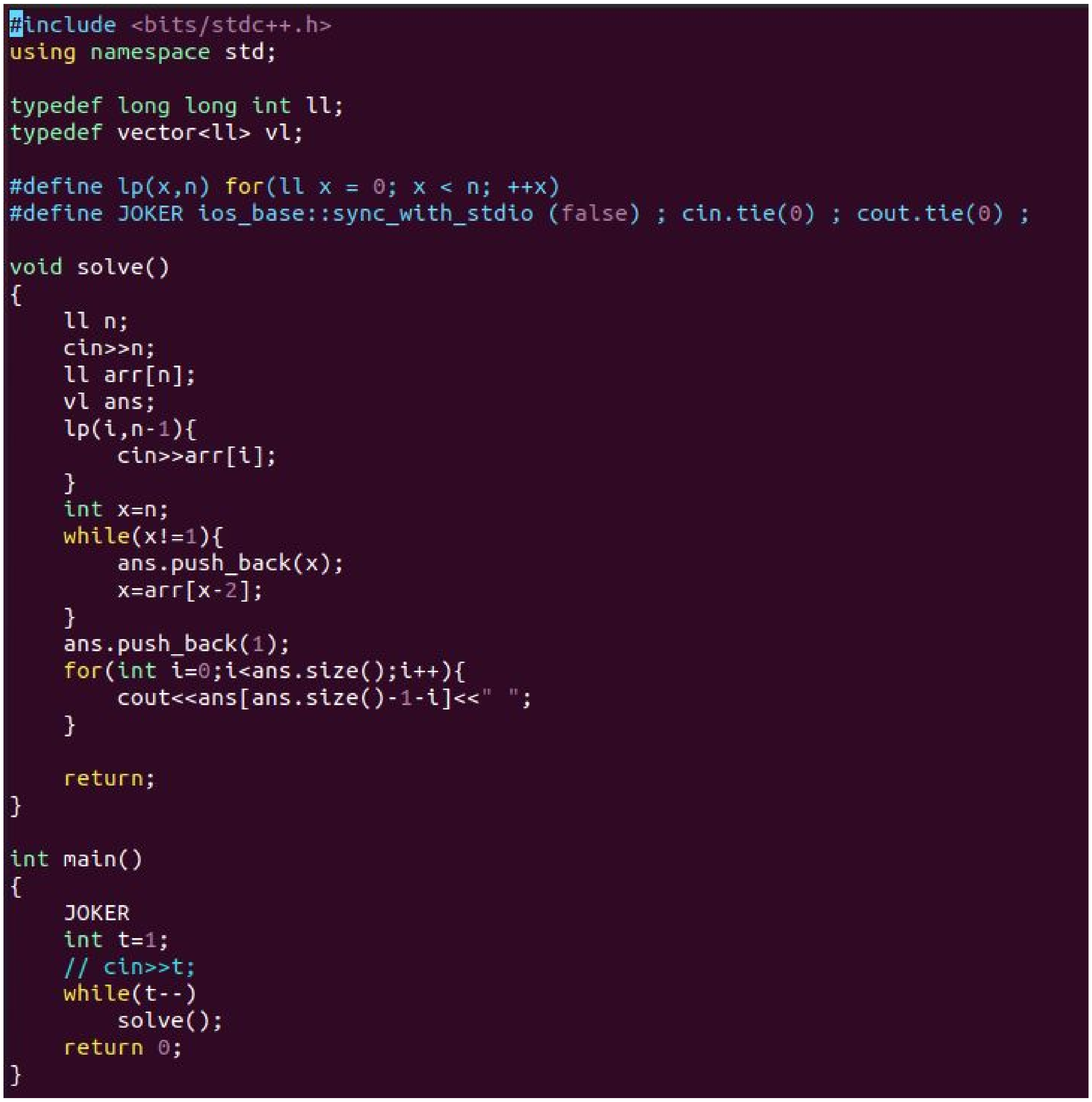
Explanation:

The array consists of room to which the room referred by index ‘i’ is connected and index ‘i’ refers to the room ‘i+2’(considering the index ‘i’ of the array starts from zero) so room ‘i+2’ is connected to arr[i].

We need to find the path from the first to the last room so we will start from the last index of the array as it refers to the last room so arr[n-2] will be the room connected to the last room now we will find the room connected to arr[n-2] which will be arr[arr[n-2]] and so on. We will continue this until we find arr[i]=1 at some point. While traversing we can store the intermediate rooms into a container and that will be the answer.

Worst-case time complexity: O(n)

Worst-case space complexity: O(n)



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

typedef long long int ll;
typedef vector<ll> vl;

#define lp(x,n) for(ll x = 0; x < n; ++x)
#define JOKER ios_base::sync_with_stdio (false) ; cin.tie(0) ; cout.tie(0) ;

void solve()
{
    ll n;
    cin>>n;
    ll arr[n];
    vl ans;
    lp(i,n-1){
        cin>>arr[i];
    }
    int x=n;
    while(x!=1){
        ans.push_back(x);
        x=arr[x-2];
    }
    ans.push_back(1);
    for(int i=0;i<ans.size();i++){
        cout<<ans[ans.size()-1-i]<<" ";
    }

    return;
}

int main()
{
    JOKER
    int t=1;
    // cin>>t;
    while(t--)
        solve();
    return 0;
}
```