PSEUDOCODE :-

1. creating an array of pairs (or a data structure to store pairs of elements)
2. inserting the pairs => (arrive[i] , 2) into the array
3. inserting the pairs => (depart[i] , 1) into the array
4. sorting the array of pairs
5. if at each instant of time, the total number of rooms available are greater than or equal to the current number of bookings, then the answer is True
6. Else, it is false

```
// Inputting the arrival array and departure array and K (no. of rooms).
// initializing an array of pairs data structure.

 vector<pair<int,int>> ans;
 int sz=arrive.size();

 for(int i=0;i<sz;i++)
 {
    //inserting the pairs for arrival and departure

    ans.push_back(make_pair(arrive[i],2));
    ans.push_back(make_pair(depart[i],1));
 }

 sort(ans.begin(), ans.end());
 int curroom=0;
 int roommax=0;

 for(int i=0;i<2*sz;i++)
 {
    // incrementing the current room bookings if there is an arrival
    if(ans[i].second==2)
    {
       curroom++;
       roommax=max(curroom,roommax);
    }
```

```
        //otherwise decrementing the current room bookings if there is a
// departure
        else{
            curroom--;
        }
    }

    if(K>=roommax)
     return true; =>1
else
     return false; => 0
```

TIME COMPLEXITY => O(N)
SPACE COMPLEXITY => O(N)