You are given a sorted array consisting of only integers where every element appears exactly twice, except for one element which appears exactly once.

arr = [1,1,2,2,3,3,4,4,**5**,6,6]
output = 5

arr = [**1**,2,2,3,3,4,4,6,6]
output = 1

arr = [1,1,4,4,**6**]
output = 6

1, 1, 2, 2, 4, 4, 5, 6, 6, 7, 7

0 1 2 3 4 5 6 7 8 9 10

Fi = 0, last = 10, mid = 5
Fi = mid + 1, fi = 6, last = 10, mid = 8
Fi = 6, last = 7, mid = 6

[1 1 2 2 3 4 4]

\

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

int missingNumber(vector<int> v) {
    int first = 0, last = v.size() -1, mid = 0, temp =0;
    if(v.size() == 1)   return v[0];

    while(first <= last) {
        mid = first + (last-first)/2;
        if(mid % 2 == 0) {
            temp = mid +1;
        }
        else {
            temp = mid - 1;
        }
        if(v[temp] == v[mid]) {
            first = mid +1;
        }
        else {
            last = mid - 1;
        }
    }
    return v[first];
}

int main() {
    vector<int> v = {1,1,4,4,5,6,6,7,7,8,8};
    cout <<missingNumber(v);
     // your code goes here
     return 0;
}
```

—————————————————

There are different types of tours and combos at Headout as given in example:

Combos; A,B,C,D (Name)
Tours: 1,2,3,4,5,6 (Name,Price)

A - 1,2,3
B - 2,3,4
C - 4,5,6
D - 1,2,3,4,5,6
E -

2 ways :

1. Combos, Tours, Combo_id-Tour_id(CT_table) -
   Combo id, tours <>

2. Denormalised schema -
   Combo, Tours <>

Queries 👍

1. Data = Select * from CT_table where combo_id = “huhif”;
2. totalPrice = 0;
3. for (item, 0->size(data){
   Item_price = Select price from tours where tour_id = item.id
   totalPrice += Item_price
   }