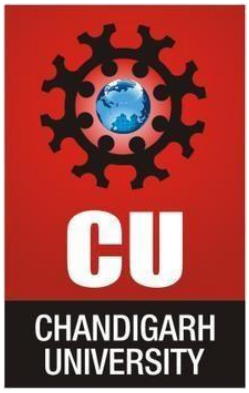
CHANDIGARH UNIVERSITY
UNIVERSITY INSTITUTE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Submitted By: Satyam **Submitted To: Navneet Chaudhry**	
Subject Name	Competitive Coding
Subject Code	21 CSP-314
Branch	BE-CSE
Semester	5th

LAB INDEX

NAME: Satyam
SUBJECT NAME: Competitive Coding Lab
UID: 20BCS9393
SUBJECT CODE: 21CSP-314
SECTION: 607-A

Sr. No	Program	Date	Evaluation				Sign
			LW (12)	VV (8)	FW (10)	Total (30)	
1.	**ARRAYS:** https://www.hackerrank.com/challenges/30-arrays/problem https://www.hackerrank.com/challenges/simple-arraysum/problem?isFullScreen=true https://www.hackerrank.com/challenges/compare-thetriplets/problem?isFullScreen=true https://www.hackerrank.com/challenges/diagonaldifference/problem?isFullScreen=true	04-Aug-2022					
2.	**STACK & QUEUES:** https://www.hackerrank.com/challenges/equalstacks/problem?isFullScreen=true https://www.hackerrank.com/challenges/game-of-twostacks/problem?isFullScreen=true https://www.hackerrank.com/challenges/balancedbrackets/problem?isFullScreen=true https://www.hackerrank.com/challenges/down-to-zeroii/problem?isFullScreen=true https://www.hackerrank.com/challenges/trucktour/problem?isFullScreen=true	18-Aug-2022					

3.	**Linked List:** https://www.hackerrank.com/challenges/compare-two-linkedlists/problem?isFullScreen=true https://www.hackerrank.com/challenges/insert-a-node-into-asorted-doubly-linked-list/problem?isFullScreen=true	25-Aug-2022					
4.	**Searching and Sorting:** https://www.hackerrank.com/challenges/missingnumbers/problem?isFullScreen=true https://www.hackerrank.com/challenges/closestnumbers/problem?isFullScreen=true	01-Aug-2022					

EXPERIMENT-1.4(a)

1. Aim/Overview of the practical:

To demonstrate the concept of Searching and Sorting.

2. Task to be done/ Which logistics used:

https://www.hackerrank.com/challenges/compare-two-linked-lists/problem?isFullScreen=true

Given two arrays of integers, find which elements in the second array are missing from the first array.

Notes

- If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.
- Return the missing numbers sorted ascending.
- Only include a missing number once, even if it is missing multiple times.
- The difference between the maximum and minimum numbers in the original list is less than or equal to 100.

Function Description

Complete the *missingNumbers* function in the editor below. It should return a sorted array of missing numbers.

missingNumbers has the following parameter(s):

- *int arr[n]:* the array with missing numbers
- *int brr[m]:* the original array of numbers

Returns

- *int[]:* an array of integers

3. Steps for experiment/practical/Code:

```
 #include<iostream> using
namespace std; const int maxn =
10000; int A[maxn*2 + 5]; int
main() {   int n, m;   int xmin
= maxn, xmax = -maxn;   cin >>
n;   for( int i = 0; i<n; i++ )
{     int tmp;     cin >> tmp;
    A[tmp] --;   }   cin >> m;
for( int i = 0; i<m; i++ ) {
int tmp;     cin >> tmp;     A[tmp]
++;     if (xmax < tmp) { xmax =
tmp; }     if (xmin > tmp) { xmin =
tmp; }
  }
  for( int i=xmin; i<=xmax; i++ ) {
if( A[i] > 0 ) {       cout << i <<
" ";
    }   }
return 0;
}
```

4. Result/Output/Writing Summary:

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Input (stdin) Download

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

Your Output (stdout)

```
204 205 206
```

Expected Output Download

```
204 205 206
```

EXPERIMENT-1.4(b)

1.Aim/Overview of the practical:

To demonstrate the concept of Linked List.

2. Task to be done/ Which logistics used:

https://www.hackerrank.com/challenges/closest-numbers/problem?isFullScreen=true
Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well. In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

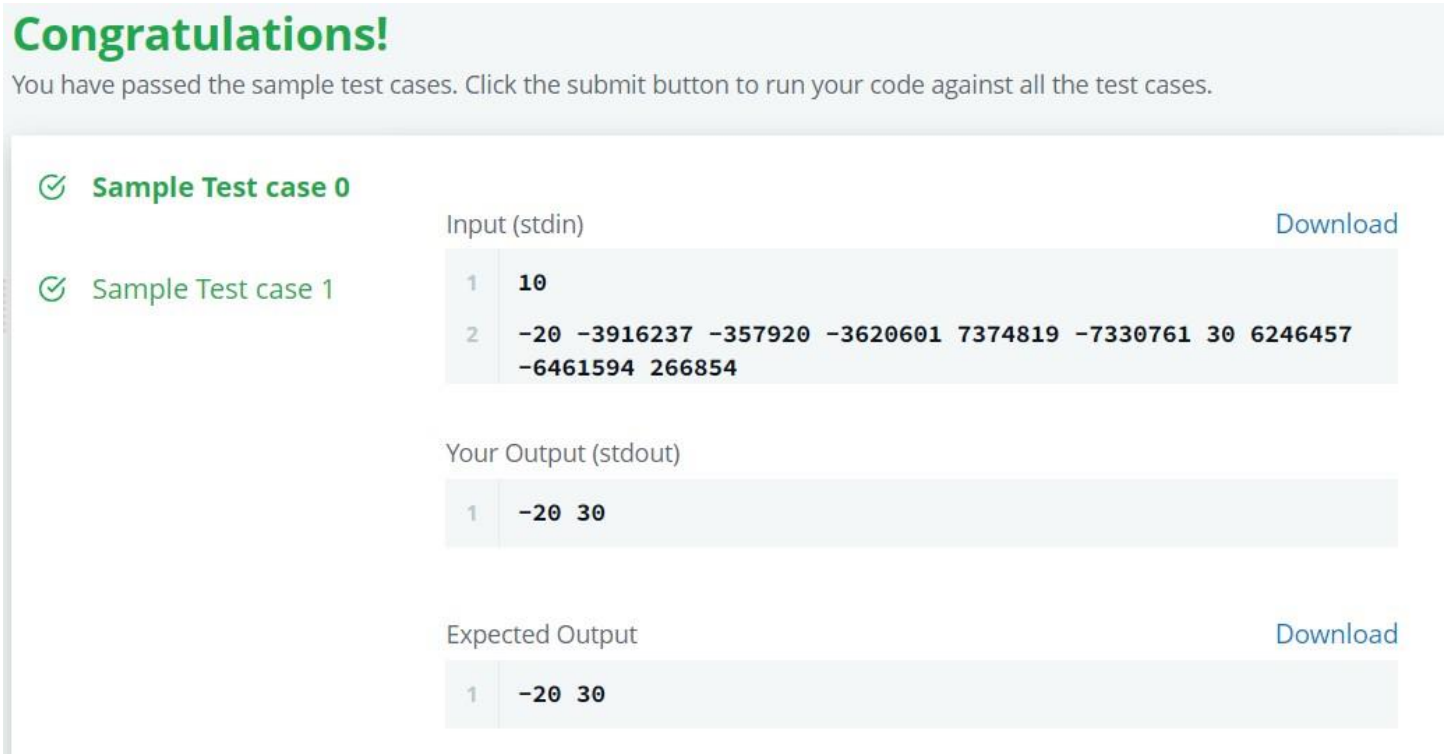
3. Steps for experiment/practical/Code:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std; int
cmp(int x,int y)
{
    return x<y;
} int main() {
vector<int> arr,temp;
int i,n,diff,x;
cin>>n;
for(i=0;i<n;i++)
    {
cin>>x;
        arr.push_back(x);
    }
sort(arr.begin(),arr.end(),cmp);
diff=arr[1]-arr[0];
```

```
for(i=2;i<n;i++)
if(diff>(arr[i]-arr[i-1]))
        {
            diff=arr[i]-arr[i-1];
temp.clear();
temp.push_back(arr[i-1]);
temp.push_back(arr[i]);
        }         else  if(diff==(arr[i]-
arr[i-1]))
        {
temp.push_back(arr[i-1]);
temp.push_back(arr[i]);
        }
for(i=0;i<temp.size();i++)
cout<<temp[i]<<" ";

    return 0;
}
```

6. Result/Output/Writing Summary:



Learning outcomes (What I have learnt):

1. Through this experiment I learn concepts of searching and sorting.

2. different operations on searching and sorting.

3.Learned about different algorithms of searching and sorting.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			