# VISVESVARAYA TECHNOLOGICAL UNIVERSITY
# BELAGAVI - 590 018, KARNATAKA.



**ASSIGNMENT – 1**
FOR

## OBJECT ORIENTED CONCEPTS (18CS45)

*Submitted by*

**NAME: SYED FARHAN**                    **USN:4BD20CS104**

**COURSE CO-ORDINATOR**              **HEAD OF DEPARTMENT**

_____                          _____

**P**rof.Naveen H M B.E,M.Tech.,             Dr.Nirmala C.R , Ph.D.

Asst.Professor,                                Prof. & Head,

Department of CS&E                         Department of CS&E,

B.I.E.T, Davanagere                          B.I.E.T, Davangere



**2022**

**Department of Computer Science and Engineering.**

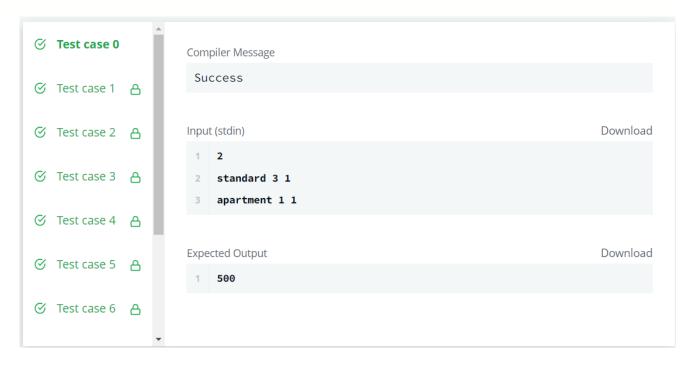**Bapuji Institute of Engineering & Technology**

**Davangere- 577004**

1. **In this challenge, the task is to debug the existing code to successfully execute all provided test files.The given code defines two classes Hotel Room and Hotel Apartment denoting respectively a standard hotel room and a hotel apartment. An instance of any of these classes has two parameters: bedrooms and bathrooms denoting respectively the number of bedrooms and the number of bathrooms in the room.**

```cpp
#include <iostream>
#include <vector>

using namespace std;

class HotelRoom
{
public:
HotelRoom(int bedrooms, int bathrooms)
: bedrooms_(bedrooms), bathrooms_(bathrooms) {}

int get_price()
{
return 50*bedrooms_ + 100*bathrooms_;
}
private:
int bedrooms_;
int bathrooms_;
};

class HotelApartment : public HotelRoom
{
public:
HotelApartment(int bedrooms, int bathrooms)
: HotelRoom(bedrooms + 2, bathrooms) {}
int get_price()
{
return HotelRoom::get_price() + 100;
}
};
int main() {
int n;
cin >> n;
vector<HotelRoom*> rooms;
for (int i = 0; i < n; ++i) {
string room_type;
int bedrooms;
int bathrooms;
cin >> room_type >> bedrooms >> bathrooms;
if (room_type == "standard") {
rooms.push_back(new HotelRoom(bedrooms, bathrooms));
} else {
rooms.push_back(new HotelApartment(bedrooms, bathrooms));
}
}

int total_profit = 0;
for (auto room : rooms) {
```

```cpp
                total_profit += room->get_price();
            }
            cout << total_profit << endl;

            for (auto room : rooms) {
                delete room;
            }
            rooms.clear();

            return 0;
        }
```
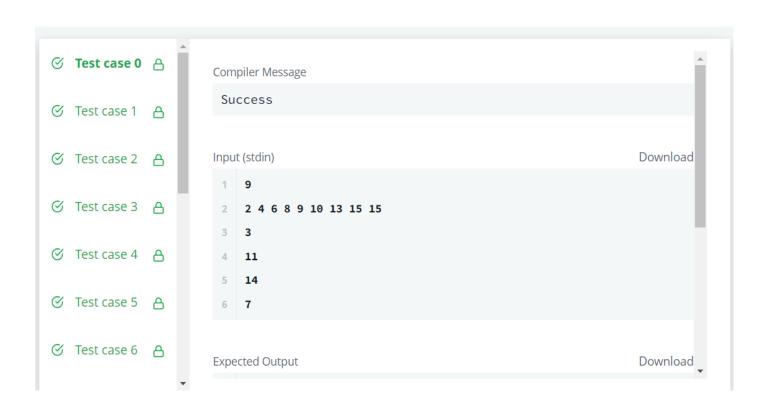
**OUTPUT :**

2. **You are given integers in sorted order. Also, you are given queries. In each query, you will be given an integer and you have to tell whether that integer is present in the array. If so, you have to tell at which index it is present and if it is not present, you have to tell the index at which the smallest integer that is just greater than the given number is present.**

```cpp
//Lower Bound-STL in C++ - Hacker Rank Solution
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main()
{
    int m, n, num, i, val;
```
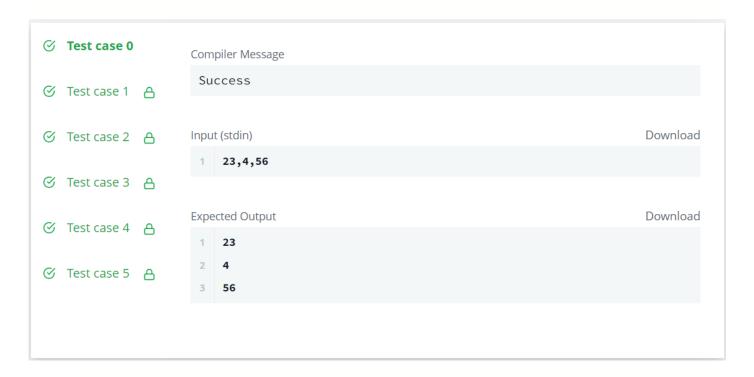
```cpp
        cin >>m;
        vector <int> v(m);
        for(i = 0; i < m; i++)
            cin >> v[i];
        cin>>n;
        for(i = 0; i < n; i++)
        {
            cin >> val;
            vector <int>::iterator low = lower_bound(v.begin(), v.end(), val);
            if (v[low - v.begin()] == val)
            {
                cout << "Yes " << (low - v.begin() + 1) << endl;
            }
            else
            {
                cout << "No " << (low - v.begin() + 1) << endl;
            }
        }
    return 0;
}
```

**OUTPUT :**

3. **In this challenge, we work with** *string streams*.
   *stringstream* **is a stream class to operate on strings. It implements input/output operations on memory (string) based streams.** *stringstream* **can be helpful in different type of parsing. The following operators/functions are commonly used here**

- *Operator >>* **Extracts formatted data.**
- *Operator <<* **Inserts formatted data.**
- *Method str()* **Gets the contents of underlying string device object.**
- *Method str(string)* **Sets the contents of underlying string device object.**

```cpp
//StringStream in C++ - Hacker Rank Solution
#include <sstream>
#include <vector>
#include <iostream>
using namespace std;

vector<int> parseInts(string str)
{
    // Complete this function
    stringstream ss(str);
    vector<int> result;
    char ch;
    int tmp;

    while (ss >> tmp)
    {
        result.push_back(tmp);
        ss >> ch;
    }

    return result;
}

int main()
{
    string str;
    cin >> str;
    vector<int> integers = parseInts(str);
    for(int i = 0; i < integers.size(); i++)
    {
        cout << integers[i] << "\n";
    }

    return 0;
}
```

**OUTPUT :**

4. **An interface Number is defined in the following program. You have to declare a class A, which will implement the interface Number. Note that the method findSqr(n) will return the square of the number n.**

```
1  import java.util.Scanner;
2
3  interface Number {
4      int findSqr(int i);   // Returns the square of n
5  }
6
7  class A implements Number {
8      //Define a method to find the square of a number
9      int i, square;
10     public int findSqr(int i)
11     {
12         square=i*i;
13         return square;
14     }
15 }
16
17
```

```
0  public class Question5_1{
1      public static void main (String[] args){
2          A a = new A();    //Create an object of class A
3          // Read a number from the keyboard
4          Scanner sc = new Scanner(System.in);
5          int i = sc.nextInt();
6          System.out.print(a.findSqr(i));
7      }
8  }
```

**OUTPUT :**

| Public Test Cases | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| Test Case 1 | 10 | 100 | 100 | Passed |

5. **This program is to find the GCD (greatest common divisor) of two integers writing a recursive function** findGCD(n1,n2)**. Your function should return -1, if the argument(s) is(are) other than positive number(s).**
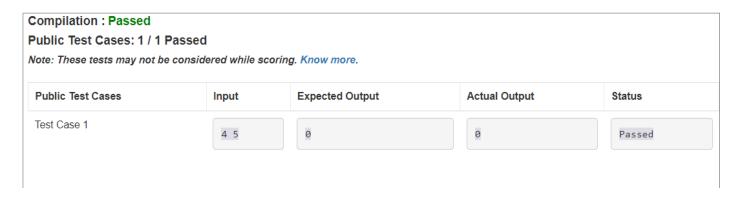
```java
1  import java.util.Scanner;
2
3  interface GCD {
4      public int findGCD(int n1,int n2);
5  }
6  //Create a class B, which implements the interface GCD.
7  class B implements GCD {
8      int n1,n2;
9
10     //Create a method to calculate GCD
11     public int findGCD(int n1, int n2){
12         if(n1==0&& n2==0){
13             return -1;
14         }
15         else if(n2 == 0){
16             return n1;
17         }
18
19         else {
20             return findGCD(n2, n1%n2);
21         }
22     }
23 }
24
25
```

```java
0  public class Question5_2{
1      public static void main (String[] args){
2          B a = new B();   //Create an object of class B
3          // Read two numbers from the keyboard
4          Scanner sc = new Scanner(System.in);
5          int p1 = sc.nextInt();
6          int p2 = sc.nextInt();
7          System.out.print(a.findGCD(p1,p2));
8      }
9  }
```

**OUTPUT :**

| Public Test Cases | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| Test Case 1 | 40 60 | 20 | 20 | Passed |

| Private Test cases used for Evaluation | Status |
|---|---|
| Test Case 1 | Passed |
| Test Case 2 | Passed |

6. **Complete the code segment to catch the ArithmeticException in the following, if any. On the occurrence of such an exception, your program should print "Exception caught: Division by zero." If there is no such exception, it will print the result of division operation on two integer values.**
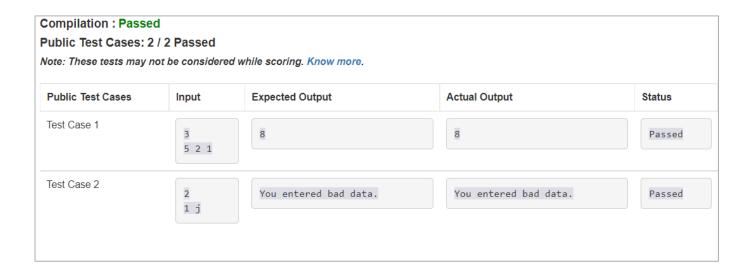
```java
1  import java.util.Scanner;
2
3    public class Question5_3 {
4    public static void main(String[] args) {
5        int a, b;
6        Scanner input = new Scanner(System.in);
7        int result;
8        a = input.nextInt();
9        b = input.nextInt();
10
11       // try block to divide two numbers and display the result
12       try {
13           result = a/b;
14           System.out.print(result);
15       }
16       // catch block to catch the ArithmeticException
17       catch (ArithmeticException e) {
18           System.out.println("Exception caught: Division by zero.");
19       }
```

**OUTPUT :**

7. In the following program, an array of integer data to be initialized. During the initialization, if a user enters a value other than integer value, then it will throw InputMismatchException exception. On the occurrence of such an exception, your program should print "You entered bad data." If there is no such exception it will print the total sum of the array.

```java
1  //Prefixed Fixed Code:
2  import java.util.*;
3  public class Question5_4{
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int length = sc.nextInt();
7          // create an array to save user input
8          int[] name = new int[length];
9          int sum=0;//save the total sum of the array.

10         try{
11             for(int i=0;i<length;i++)
12             {
13                 int userInput=sc.nextInt();
14                 name[i] = userInput;
15                 sum=sum+name[i];
16             }
17             System.out.print(sum);
18         }
19         catch(InputMismatchException e) {
20             System.out.print("You entered bad data.");
21         }

0  }
1  }
```

**OUTPUT :**

| Public Test Cases | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| Test Case 1 | 3<br>5 2 1 | 8 | 8 | Passed |
| Test Case 2 | 2<br>1 j | You entered bad data. | You entered bad data. | Passed |

8. **In the following program, there may be multiple exceptions. You have to complete the code using only one try-catch block to handle all the possible exceptions.**
   **For example, if user's input is 1, then it will throw and catch "java.lang.NullPointerException".**

```java
1 import java.util.Scanner;
2 public class Question5_5{
3     public static void main (String   args[ ] ) {
4             Scanner scan = new Scanner(System.in);
5             int i = scan.nextInt();
6         int j;
7             // Put the following code under try-catch block to handle exceptions
8             try {
9                 switch(i)
10                {
11                    case 0 : int zero = 0;
12                            j = 92/ zero;
13                            break;
14                    case 1 : int b[ ] = null;
15                            j = b[0] ;
16                            break;
17                    default: System.out.print("No exception");
18                }
19            }
20            // catch block
21            catch (Exception e) {
22              System.out.print(e) ;
23            }
0 }
1 }
```

**OUTPUT :**

Compilation : **Passed**

Public Test Cases: 3 / 3 Passed

*Note: These tests may not be considered while scoring. Know more.*

| Public Test Cases | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| Test Case 1 | 5 | `No exception` | `No exception` | Passed |
| Test Case 2 | 0 | `java.lang.ArithmeticException: / by zero` | `java.lang.ArithmeticException: / by zero` | Passed |
| Test Case 3 | 1 | `java.lang.NullPointerException` | `java.lang.NullPointerException` | Passed |

9. **Define a class Point with two fields x and y each of type double. Also, define a method distance(Point p1, Point p2) to calculate the distance between points p1 and p2 and return the value in double.Complete the code segment given below. Use Math.sqrt( ) to calculate the square root.**

```java
1  import java.util.Scanner;
2
3  public class Circle extends Point{
4
5  public static void main(String[] args) {
6
7      Scanner sc = new Scanner(System.in);
8      Point p1=new Point();
9      p1.x=sc.nextDouble();
10     p1.y=sc.nextDouble();
11     Point p2=new Point();
12     p2.x=sc.nextDouble();
13     p2.y=sc.nextDouble();
14     Circle c1=new Circle();
15     c1.distance(p1,p2);
16
17  }
18
19 }
20 |
21 class Point{
22    double x;
23    double y;
24
25 public static void distance(Point p1,Point p2){
26        double d;
27        d=Math.sqrt((p2.x-p1.x)*(p2.x-p1.x) + (p2.y-p1.y)*(p2.y-p1.y));
28        System.out.println(d);
29    }
30 }
31
```

**OUTPUT :**

| | Input | Output |
|---|---|---|
| Test Case 1 | 2.0 1.0<br>1.0 1.0 | 1.0 |
| Test Case 2 | 2.0 3.0<br>1.0 2.0 | 1.4142135623730951 |
| Test Case 3 | 2.0 3.0<br>1.0 2.0 | 1.4142135623730951 |

10. **This program to exercise the call of static and non-static methods. A partial code is given defining two methods, namely sum( ) and multiply ( ). You have to call these methods to find the sum and product of two numbers. Complete the code segment as instructed.**

```java
1  import java.util.Scanner;
2  class QuestionScope {
3  int sum(int a, int b){ //non-static method
4          return a + b;
5      }
6  static int multiply(int a, int b){ //static method
7          return a * b;
8      }
9  }
10 public class Test3{
11 public static void main( String[] args ) {
12         Scanner sc = new Scanner(System.in);
13         int n1=sc.nextInt();
14         int n2=sc.nextInt();
15
16 //Called the method sum() to find the sum of two numbers.
17 //Called the method multiply() to find the product of two numbers.
18 QuestionScope x=new QuestionScope();
19 int i=x.sum(n1,n2);
20 int j=x.multiply(n1,n2);
21 System.out.println(i);
22 System.out.print(j);
23
24 }
25 }
```

**OUTPUT :**

| Private Test cases used for evaluation | Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| Test Case 1 | 3 5 | 8\n<br>15\n | 8\n<br>15 | Passed |
| Test Case 2 | 1 2 | 3\n<br>2 | 3\n<br>2 | Passed |