

Sessional Test II – May, 2023 Semester IV

Title of the Course: Core Java

Course Code: CS 109

(5 Marks Coding Questions)

SET-H

Q1. Given a string **S**, find length of the longest substring with all distinct characters.

Input format:

Input will consist of string.

Output format:

The output will be a single integer denoting the length of the longest substring.

Sample Input:

Chitkara

Sample Output:

7

Sample Input:

acfagg

Sample Output:

4

Solution: -

```
import java.util.Scanner;
public class Input {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        int result = Solution.longestSubstrDistinctChars(s);
        System.out.println(result);
    }
}
```

```
class Solution{
    static int longestSubstrDistinctChars(String S){
        int n = S.length();
        if (n == 0) return 0;

        int maxLen = 1;
        for (int i = 0; i < n-1; i++) {
            boolean[] visited = new boolean[128];
            visited[S.charAt(i)] = true;
            for (int j = i+1; j < n; j++) {
                if (visited[S.charAt(j)]) {
```

```

        break;
    }
    visited[S.charAt(j)] = true;
    maxLen = Math.max(maxLen, j-i+1);
}
}
return maxLen;
}
}

```

Q2. Write a program that takes a string input and returns the number of vowels in the string. The program should handle exceptions that may occur during input or processing.

Input format:

A single string input.

Output format:

An integer representing the number of vowels in the input string.

Sample Input 1:

"Hello, World!"

Sample Output 1:

3

Sample Input 2:

"bcd"

Sample Output 2:

0

Solution:

```

import java.util.Scanner;
class Solution {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        //System.out.print("Enter a string: ");
        try {
            String str = input.nextLine();
            int numVowels = countVowels(str);
            System.out.println( numVowels);
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
        public static int countVowels(String str) {
            int count = 0;
            for (int i = 0; i < str.length(); i++) {
                char c = Character.toLowerCase(str.charAt(i));
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
                    count++;
                }
            }
            return count;
        }
    }
}

```

Q1. Write a function to check whether two given strings are Permutation of each other or not.

A Permutation of a string is another string that contains same characters, only the order of characters can be different.

For example, “abcd” and “dabc” are Permutation of each other.

Input format:

Accept two strings

Output format:

1 if the strings are permutations of each other. 0 if the strings are not permutation of each other. As specified in sample Ouput

Test Cases:

Input	Output
abc bca	1: abc and bca are permutable
raw was	0: raw and was are not permutable

SOLUTION:

```
import java.io.*;
import java.util.*;
```

```
class Main{
```

```
    static int NO_OF_CHARS = 256;
```

```
    static boolean arePermutation(String str1, String str2)
    {
```

```
        int count1[] = new int [NO_OF_CHARS];
        Arrays.fill(count1, 0);
        int count2[] = new int [NO_OF_CHARS];
        Arrays.fill(count2, 0);
        int i;
```

```
        for (i = 0; i < str1.length() && i < str2.length() ;
              i++)
```

```
        {
            count1[str1.charAt(i)]++;
```

```

        count2[str2.charAt(i)]++;
    }

    if (str1.length() != str2.length())
        return false;

    // Compare count arrays
    for (i = 0; i < NO_OF_CHARS; i++)
        if (count1[i] != count2[i])
            return false;

    return true;
}

public static void main(String args[])
{
    String str1, str2;

    Scanner sc = new Scanner(System.in);
    str1 = sc.next();
    str2 = sc.next();

    if (arePermutation(str1, str2))
        System.out.print(1 + ": " + str1 + " and " + str2 + " are permutable");
    else
        System.out.print(0 + ": " + str1 + " and " + str2 + " are not permutable"); } }

```

Q2: Given total runs scored and total overs faced as the input. Write a program to calculate the run rate with the formula, $\text{Run rate} = \frac{\text{total runs scored}}{\text{total overs faced}}$. Use Scanner class to get the inputs from user. This program may generate Arithmetic Exception / InputMismatchException. Use exception handling mechanisms to handle this exception. Use a single catch block. In the catch block, print the class name of the exception thrown.

Input format:

First line of the input contains Enter the total runs scored.

Second line contains Enter the total overs faced

Output format:

First Line contains Current Run Rate : then value or Exception Message

Sample Input:

Enter the total runs scored

79

Enter the total overs faced

14

Sample Output:

Current Run Rate : 5.00

Sample Input:

Enter the total runs scored

50

Enter the total overs faced

0

Sample Output:

java.lang.ArithmeticException

SOLUTION:

```
import java.util.InputMismatchException;
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.println("Enter the total runs scored");
            int totalRun = sc.nextInt();
            System.out.println("Enter the total overs faced");
            int totalOver = sc.nextInt();
            int p = totalRun / totalOver;
            float runOver = (float) (totalRun / totalOver);
            System.out.println("Current Run Rate : " + runOver);

        } catch (ArithmeticException | InputMismatchException e) {
            System.out.println(e.getClass().getName());
        }
    }
}
```

SET-F

Q1. You are developing a drawing application that supports different shapes such as rectangles and circles. Implement an abstract class to represent a shape and its specific subclasses for rectangles and circles. Prompt the user to enter the dimensions and properties of a shape and display its area and perimeter.

abstract class will hold gerArea() and getPerimeter() method.

Input Format

First line input consists integer value, that is choice of shape (1. Rectangle, 2. Circle)

Second line input consists of double values (for Rectangle it will be 2 space separated values that is Length and Width, for Circle it will be only 1 value that is Radius, fi choice is 3 print “Invalid Choice”).

Output Format

Output will be lines separated double values showing the Area and Perimeter/ Circumference of the chosen shape.

Sample Input 1

1

5.2

3.7

Sample Output 1

Area: 19.24

Perimeter: 17.80

Sample Input 2

2

4.5

Sample Output 2

Area: 63.62

Circumference: 28.27

Sample Input 2

3

Sample Output 2

Invalid choice!

#Solution

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                double length = scanner.nextDouble();
                double width = scanner.nextDouble();
                Rectangle rectangle = new Rectangle(length, width);
                System.out.printf("Area: %.2f", rectangle.getArea());
                System.out.printf("\nPerimeter: %.2f", rectangle.getPerimeter());
                break;
            case 2:
                double radius = scanner.nextDouble();
                Circle circle = new Circle(radius);
                System.out.printf("Area: %.2f", circle.getArea());
                System.out.printf("\nCircumference: %.2f", circle.getPerimeter());
                break;
            default:
                System.out.println("Invalid choice!");
                break;
        }
        scanner.close();
    }
}

abstract class Shape {
    public abstract double getArea();
    public abstract double getPerimeter();
}
```

```

class Rectangle extends Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double getArea() {
        return length * width;
    }
    public double getPerimeter() {
        return 2 * (length + width);
    }
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }
}

```

Q2. In a thermos steel company while entering the names of the employees in the excel file the clerk made a mistake. Some leading and trailing spaces have been added by mistake. Now the company wants the final fair sheet of the names. So, help the company with the same.

Write a JAVA program to remove all the leading and trailing spaces from the names.

Input Format

The input consists of string followed by leading and trailing whitespaces.

Output Format

The output consists of string with no leading and trailing whitespaces.

Sample Input 1

Welcome

Sample Output 1

Welcome

Sample Input 2

Chitkara

Sample Output 2

Chitkara

#Solution

```
import java.util.Scanner;
```

```

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        System.out.println(str.trim());
    }
}

```

SET-E

Q1. Chitkara University has a huge amount of data of students, they want to maintain the register of student's details in alphabetical order, write a JAVA program to help the University to arrange the names in alphabetical order.

Input Format

First line inputs consist single integer value that is number of records to be entered N,

Second line input consists N String values that are Names of student.

Output Format

Output consists N student names in Alphabetical order each in new line.

Sample Input 1

5

Mahesh Sharma

Ashish Gil

Rahul Dhingra

Aman Gupta

Mayur Chabra

Sample Output 1

Aman Gupta

Ashish Gil

Mahesh Sharma

Mayur Chabra

Rahul Dhingra

Sample Input 2

3

Abhishek Pathak

Varun Nair

Preenu Tyagi

Sample Output 2

Abhishek Pathak

Preenu Tyagi

Varun Nair

#Solution

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
    }
}

```



```

int n = sc.nextInt();
sc.nextLine();
String names[] = new String[n];
for (int i=0 ; i<n ; i++)
{
    names[i] = sc.nextLine();
}
String temp;
for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        if (names[i].compareTo(names[j]) > 0) {
            temp = names[i];
            names[i] = names[j];
            names[j] = temp;
        } } }
for (int i = 0; i < n; i++) {
    System.out.println(names[i]);
} } }

```

Q2. Anagram is a word or phrase made by transposing the letters of another word or phrase

The word "secure" is an anagram of "rescue."

Write a program that takes two strings as input and checks if they are anagrams of each other. The program should handle exceptions that may occur during input or processing.

Input format:

Two strings separated by a space.

Output format:

"YES" if the strings are anagrams of each other, "NO" otherwise as shown in sample test case.

	Test Case 1	Test Case 2
Input	listen silent	gram arm
Output	YES listen silent are anagram	NO gram arm are not anagram

#Solution

```

import java.util.Arrays;
import java.util.Scanner;

```

```

class Main {

```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    //System.out.print("Enter two strings separated by a space: ");
    try {
        String str1 = input.next();
        String str2 = input.next();
        boolean isAnagram = checkAnagram(str1, str2);

        System.out.print(isAnagram ? "YES " + str1 + " " + str2+ " are anagram": "NO " + str1 +
" " + str2+ " are not anagram");
        //System.out.print(" " + str1 + " " + str2+ " are anagram" );
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
}

public static boolean checkAnagram(String str1, String str2) {
    char[] arr1 = str1.toCharArray();
    char[] arr2 = str2.toCharArray();
    Arrays.sort(arr1);
    Arrays.sort(arr2);
    return Arrays.equals(arr1, arr2);
}
}

```

SET-D

Q1. A finance company wants to calculate the total sales and the accountant needs to calculate the exact amount of the total sales and he has been instructed to roundoff the obtained total to nearest multiple of 10. So write a JAVA program to help the accountant to round the given number to nearest multiple of 10

Input Format

Input consists of a positive integer value.

Output Format

Output consists of positive integer value which is rounded to the nearest whole number having zero as last.

Sample Input 1

4722

Sample Output 1

4720

Sample Input 2

10

Sample Output 2

10

#Solution

```

import java.util.*;
class Main {
static int round(int n)
{
    int a = (n / 10) * 10;
    int b = a + 10;
    return (n - a > b - n)? b : a;
}
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    System.out.println(round(n));
}
}

```

Q2. Rajan is writing some code in a project that contains an array of employee. He needs to search for an employee with the given id. In case the employee is found, the employee object is returned else an exception is thrown with the message “employee not found”. Write the method to search the employee and create a class that extends Exception and override the method getMessage().

Input format:

Input is N, number of employees

N lines containing, id and name of the employee

X, an integer containing the id to search

Output format:

Name of the employee if found.

Exception, in case the employee is not found.

Sample Input:

```

3
1 sumit
2 rajan
3 sumit
5

```

Sample Output:

employee not found

Sample Input:

```

5
1 sumit
2 rajan
3 geeta
4 sunita
5 kriti
4

```

Sample Output:

Sunita

SOLUTION:

```
import java.util.Scanner;
public class Main {
    public static void main(String args[])
    {
        Scanner sc =new Scanner(System.in);
        int n = sc.nextInt();
        Employee[] ar = new Employee[n];
        for(int i =0; i<n; i++){
            int id= sc.nextInt();
            String name = sc.next();
            Employee emp = new Employee(id,name);
            ar[i] = emp;
        }
        int x = sc.nextInt();
        try
        {
            Employee emp = search(ar, x);
            System.out.println(emp.name);
        }
        catch(EmployeeNotFoundException e)
        {
            System.out.println(e.getMessage());
        }
    }
    public static Employee search(Employee[] ar, int x) throws
EmployeeNotFoundException
    {
        for(Employee emp : ar)
        {
            if(emp.id == x) return emp;
        }
        throw new EmployeeNotFoundException();
    }
}
class Employee {
    public int id;
    public String name;
    Employee(int id, String name) { this.id= id; this.name = name;}
}
class EmployeeNotFoundException extends Exception
{
    public String getMessage() {
        return "employee not found"; }
}
```

SET-C

Q1. A company has a program that processes employee data, including their names, IDs, and salaries. Sometimes, the program encounters errors when processing the data, such as invalid IDs or negative salaries. You need to implement exception handling to handle these errors and display appropriate error messages to the user.

Input Format

First line input consists of String value that is name of Employee,

Second line input consists of Integer value that is Employee ID,

Third input consists double value that is Salary of Employee.

Output Format

The program provides the following outputs based on the input,

If all inputs are valid, it displays the message "Data Processed",

If the ID entered is less than or equal to 0, it displays the Exception "Invalid ID"

If the salary entered is negative, it displays the Exception "Invalid salary"

Note: if exception occurred in any input line, program will not continue it will just display exception message.

Sample Input 1

John

-1

Sample Output 1

Invalid ID

Sample Input 2

Sarah

1234

-5500

Sample Output 2

Invalid salary

#Solution

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            String name = scanner.nextLine();
            int id = scanner.nextInt();
            if (id <= 0) {
                throw new IllegalArgumentException("Invalid ID");
            }
            double salary = scanner.nextDouble();
            if (salary < 0) {
                throw new IllegalArgumentException("Invalid salary");
            }
            System.out.println("Data Processed");
        }
```

```

    }
    catch (IllegalArgumentException e) {
        System.out.println(e.getMessage());
    }
    catch (Exception e) {
        System.out.println(e.toString());
    }
    finally {
        scanner.close();
    }
} }

```

Q2: **Anagram is a word or phrase made** by transposing the letters of another word or phrase

The word "secure" is an anagram of "rescue."

Write a program that takes two strings as input and checks if they are anagrams of each other. The program should handle exceptions that may occur during input or processing.

Input format:

Two strings separated by a space.

Output format:

"YES" if the strings are anagrams of each other, "NO" otherwise as shown in sample test case.

#Solution

```

import java.util.Arrays;
import java.util.Scanner;

```

```

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        //System.out.print("Enter two strings separated by a space: ");
        try {
            String str1 = input.next();
            String str2 = input.next();
            boolean isAnagram = checkAnagram(str1, str2);

            System.out.print(isAnagram ? "YES " + str1 + " " + str2 + " are anagram": "NO " + str1 +
" " + str2 + " are not anagram");
            //System.out.print(" " + str1 + " " + str2 + " are anagram" );
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

```

    }

    public static boolean checkAnagram(String str1, String str2) {
        char[] arr1 = str1.toCharArray();
        char[] arr2 = str2.toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        return Arrays.equals(arr1, arr2);
    }
}

```

SET-B

Q1. You are developing a banking application that allows users to withdraw money from their accounts. As part of the withdrawal process, the application needs to check if the requested withdrawal amount is within the available balance. If the withdrawal amount exceeds the balance, an exception should be thrown and an appropriate error message should be displayed to the user. Write a Java program that handles this exception and provides a user-friendly error message.

Input Format

First input consists decimal number, that is available balance,

Second input consists decimal number, that is withdrawal amount.

Output Format

The program will display the appropriate output based on the given input. The output will include information such as the “Withdrawal Successful. Available balance: ????” or “Withdrawal Unsuccessful”.

Sample Input 1

500

200

Sample Output 1

Withdrawal Successful. Available balance: 300.0

Sample Input 2

1000

1200

Sample Output 2

Withdrawal Unsuccessful

#Solution

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double accountBalance = scanner.nextDouble();
        double withdrawalAmount = scanner.nextDouble();

        try {
            if (withdrawalAmount > accountBalance) {
                throw new InsufficientBalanceException("Withdrawal Unsuccessful");
            }
        }
    }
}

```

```

        } else {
            accountBalance -= withdrawalAmount;
            System.out.println("Withdrawal Successful. Available balance: " + accountBalance);
        }
    } catch (InsufficientBalanceException e) {
        System.out.println(e.getMessage());
    }
}

}

class InsufficientBalanceException extends Exception {
    public InsufficientBalanceException(String message) {
        super(message);
    }
}

```

Q2. In a manufacturing company while registering the names of the employees the clerk made some typing mistakes like he entered the names with the punctuation marks in it. So, help the organization to find the proper names from the mistaken names. Write a JAVA program to remove punctuations from a given string.

Input Format

The input consists of string containing punctuations in it.

Output Format

The output consists of string without any punctuations.

Sample Input 1

My!@#Ana%*#tomy

Sample Output 1

MyAnatomy

Sample Input 2

Ch!@it#ka%^ra

Sample Output 2

Chitkara

#Solution

```

import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        str = str.replaceAll("\\p{Punct}", "");
        System.out.println(str);
    }
}

```

SET-A

Q1. You are working on a calculator program that accepts two integer numbers and performs various mathematical operations. Implement exception handling to handle the case when the user enters invalid input, such as zero or non-numeric values.

Input Format

First line of input consists integer value, that is num1,

Second line of input consists integer value, that is num2.

Output Format

Output will be dependent on input,

If user enters $\text{num1} < 0$, it should print – “First number is Zero”

If user enters $\text{num2} < 0$, it should print – “Second number is Zero”

If user enters non integer value handle the related java exception,

If no exception found, print addition, subtraction, multiplication, division of num1, num2 each in new line.

Sample Input 1

0

1111

Sample Output 1

First number is Zero

Sample Input 2

1111

0

Sample Output 2

Second number is Zero

#Solution

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            int num1 = scanner.nextInt();
            if (num1 < 1)
            {
                throw new ArithmeticException();
            }
        }
        try
        {
            int num2 = scanner.nextInt();
            if (num2 < 1)
            {
                throw new ArithmeticException();
            }
        }
```

```

        System.out.println(num1 + num2);
        System.out.println(num1 - num2);
        System.out.println(num1 * num2);
        System.out.println(num1 / num2);
    } catch (ArithmeticException e)
    {
        System.out.println("Second number is Zero");
    }
} catch (ArithmeticException e) {
    System.out.println("First number is Zero");
} catch (Exception e) {
    System.out.println("Non-Integer Input");
}
scanner.close();
}
}

```

Q2: You are given provided with a string S and you have to take all even-indexed characters and odd-indexed characters from a string and concatenates them together.

Input Format

The first line of the input contains the string S.

Output format

Print the string after merging.

Constraints

1 <= S <=1000

Time Limit

1 second

Sample Input

abcdefg

Sample Output

acegbdf

Default Code

```

import java. util.*;
import java.io.*;
public class Main {
    public static void main(String args[]) throws IOException {
        //write your code here
    }
}

```

SOLUTION

```

import java.util.*;
import java.io.*;
public class Main
{
    static String indexShuffle(String str)

```

```

{
    String e = "";
    String o = "";

    for (int i = 0 ; i < str.length() ; i++)
    {
        if (i%2==0)
        {
            e += str.charAt(i);
        }
    else
    {
        o += str.charAt(i);
    }
    }

    return e+o;
}

public static void main(String[] args) throws IOException
{
    Scanner scanner = new Scanner(System.in);

    String S = scanner.nextLine();

    String ans = indexShuffle (S);
    System.out.println(ans);
    scanner.close();
}
}

```

(Pool of 10 Marks Coding Questions)**SET-H****Problem Statement**

You are provided with three numbers w,h,r where w is the width and h is the height of the rectangle and r is the radius of the circle and you have to check if the rectangle fits inside the circle or not.

Input Format

The first line of the input three integers w,h and r.

Output format

Print “true” if the rectangle fits inside the circle else print “false”(without quotes).

Constraints

$1 \leq w, h, r \leq 1000$

Time Limit

1 second

Example**Sample Input**

8 6 5

Sample Output

true

Default Code

```
import java.util.*;  
import java.io.*;
```

```
public class Main {  
    public static void main(String args[]) throws IOException {
```

```
        //write your code here
```

```
    }  
}
```

SOLUTION

```
import java.util.*;  
import java.io.*;
```

```
public class Main  
{  
    static boolean rectangleInCircle(int w, int h, int r)
```

```

        {
            return r*2>= Math.sqrt(w*w+h*h);
        }

    public static void main(String[] args) throws IOException
    {
        Scanner scanner = new Scanner(System.in);

        int w = scanner.nextInt();
        int h = scanner.nextInt();
        int r = scanner.nextInt();

        if(rectangleInCircle(w,h,r))
        {
            System.out.println("true");
        }
        else
        {
            System.out.println("false");
        }
        scanner.close();
    }
}

```

SET-G

Write a program in Java to implement an integer array and perform following operations in form of functions one after another in same sequence as mentioned:

(Applying, Evaluating)

1. Create an integer array having length of five (05) elements.
2. Input all five elements one after another.
3. Find maximum element from the input array.
4. Find minimum element from the input array.
5. Find Subtraction of all elements of the input array consecutively. Subtract first element from second, second element from third and so on. Subtraction starts from index 0 to index 4.
 - a. Raise exception “Subtract is greater than equal to Zero”, if subtraction result is in positive or zero.
 - b. Raise exception “Subtract is less than Zero”, if subtraction result is zero.

Input format:

Each line of the input contains array of five integers separated with a space.

Constraints:

Entered elements should be greater than 0 and lesser than 10

(1 >= Arr[i] <= 9)

Output format:

Each line of the output contains the result

1. Maximum integer of input array,
2. Minimum integer of input array,
3. Subtract all integers in input array as discussed above,
 - a. If subtraction result is in zero or positive, then raise exception "Subtract is greater than equal to Zero", and
 - b. If subtraction result is in negative, then raise exception "Subtract is less than Zero",

Sample Input:

1 2 3 4 5

Sample Output:

5

1

-13

java.lang.Exception: Subtract is less than Zero

Explanation:

1. Code should be able to identify maximum and minimum elements of input array as 5 and 1 shown in above example.
2. Code should be able to find subtraction and also be able to raise exceptions as discussed above. For example: If array elements are 1 2 3 4 5 then
$$\begin{aligned}1 - 2 &= -1 \\ -1 - 3 &= -4 \\ -4 - 4 &= -8 \\ -8 - 5 &= -13\end{aligned}$$

Code:

//Yellow highlighted part should be fixed on CN

```
import java.util.Scanner;
```

```
public class Main
```

```
{
```

```
public static final MyArray myarr = new MyArray();
```

```

public static void main(String[] args)
{
    myarr.input();
    myarr.max();
    myarr.min();
    try
    {
        myarr.subfn();
    }
    catch (Exception e)
    {
        System.out.println(e);
    }
}
}
}

```

SOLUTION:

```

import java.util.Scanner;
public class Main
{
    public static final MyArray myarr = new MyArray();
    public static void main(String[] args)
    {
        myarr.input();
        myarr.max();
        myarr.min();
        try
        {
            myarr.subfn();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

    }
    }
class MyArray
{
Scanner sc = new Scanner(System.in);
public static final int[] Arr = new int[5];

    public void input()
    {
for(int i=0; i<5; i++)
    {
Arr[i]=sc.nextInt();
    }
    }

    public void max()
    {
int max = 0;
for (int i =0 ;i<5;i++)
    {
if (Arr[i] > max)
    {
max = Arr[i];
    }
    }
System.out.println(max);
    }
public void min()
    {
int min = 10;
for (int i =0 ;i<5;i++)
    {
if (Arr[i] < min)
    {
min = Arr[i];
    }
    }
System.out.println(min);
    }
public void subfn() throws Exception
    {
int sub = Arr[0];
for (int i =1 ;i<5;i++)
    {
sub = sub - Arr[i];

```



```

}
System.out.println(sub);
if (sub<0)
{
throw new Exception("Subtract is less than Zero");
}
else
{
throw new Exception("Subtract is greater than equal to Zero");
}
}
}
}

```

SET-F

Given an input string s, reverse the order of the words.

A word is defined as a sequence of non-space characters. The words in s will be separated by at least one space. Print a string of the words in reverse order concatenated by a single space.

Please note that the input string s may contain leading or trailing spaces, or multiple spaces between two words. However, the output string should only have a single space separating the words, without any extra spaces.

Input format:

Accept a string

Output format:

Print the string of the words in reverse order

Sample Input 1:

Chitkara is best

Sample Output 1:

best is Chitkara

Sample Input 2:

I am the last of the stars

Sample Output 2:

stars the of last the am I

Sample Input 3:

the happy ending

Sample Output 3:

ending happy the

Code:

```

import java.util.Scanner;
public class Main {
    static String reverse(String str)

```

```

{
    String rev="";
    for(int i=str.length()-1;i>=0;i--)
    {
        rev+=str.charAt(i);
    }
    return rev;
}
static String reverseWords(String str) {

    int i = str.length()-1;
    String ans="";

    while(i>=0)
    {
        while(i>=0 && str.charAt(i)==' ')
        {
            i--;
        }

        String word="";
        while(i>=0 && str.charAt(i)!=' ')
        {
            word+=str.charAt(i);
            i--;
        }

        ans+=reverse(word);

        if(word.length() !=0)
            ans+=" ";

    }

    return ans.substring(0,ans.length()-1);

}

public static void main(String args[]) {
    String str;
    Scanner sc=new Scanner(System.in);
    str=sc.nextLine();

```

```

        System.out.print(reverseWords(str));
    }
}

```

SET-E

Q.1. An e-commerce store's daily sales are stored in the form of a string S tagged 'a-z' or 'A-Z' in the company's database online. Each character in the string represents the product code. The ascii value of the product code represents the sale count of that product. The company wishes to find out the total sale count of N desired products. The total sale count of desired products is the sum of ascii values of occurrence of all the desired products in the string S. If a particular product occurs k times in the string then it's ascii value is added k times while calculating the total sale count.

Write a JAVA program to help the company find out the total sale count of the desired N products. Note: The characters in the product Code are always unique.

Input Format

First line of input consists of a string - sale, representing the daily sales of the company(S).

Second line of input consists of an integer size representing the number of product codes shortlisted by the company (N).

Third line of input consists of N space-separated characters - productCode, representing the desired product code.

Output Format

Print an integer representing the total sales count of the shortlisted products.

Sample Input 1

aAbcDbfdab

3

a A b

Sample Output 2

553

Explanation:

'a' occurs 2 times i.e. $2 * 97 = 194$

'A' occurs 1 time i.e. 65.

'b' occurs 3 times i.e. $3 * 98 = 294$

Total sales count is $194+65+294 = 553$.

Hence the output is 553.

Sample Input 2

iamneo

2

a e

Sample Output 2

198

#Solution

```
import java.util.*;
```

```

class Main
{
    public static int productcode(String name,int l,char[] c)
    {
        int count=0,sum=0;
        for(int i=0; i<c.length; i++)
        {
            count=0;
            for(int j=0;j<name.length();j++)
            {
                if(name.charAt(j) == c[i])
                    count++;
            }
            sum=sum+((int)c[i]*count);
        }
        return sum;
    }
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        String name=s.nextLine();
        int l=s.nextInt();
        char[] c=new char[l];
        for(int i=0;i<l;i++)
        {
            char x =s.next().charAt(0);
            c[i]=x;
        }
        System.out.println(productcode(name,l,c));
    }
}

```

SET-D

Write a program to print the count of words and print the first letter of every word as well as their corresponding ASCII values in a user input string separated by space.

Note : Assume there is a single space between two words and there are no extra spaces before and after words.

Input format:

First line of the input contains the user input string.

Output format:

On a single line of output print the count of word <space> first letter of each word with their corresponding ASCII values(space separated).

Constraints :

$$0 \leq |S| \leq 10^7$$

where $|S|$ represents the length of string, S .

Sample Input:

Hello My a

Sample Output:

3 H72 M77 a97

Sample Input:

Who are you m friend?

Sample Output:

5 W87 a97 y121 m109 f102

Solution:

```
import java.util.Scanner;
class Count_Words {
    public static int countWords(String str, char arr[]){
        int count=0;
        int num_words = 0;
        for(int i=1;i<str.length();i++)
        {
            if(i - 1 == 0){
                arr[num_words] = str.charAt(i-1);
                num_words++;
            }
            else if(str.charAt(i-1) == ' '){
                arr[num_words] = str.charAt(i);
                num_words++;
                count++;
            }
        }
        if(str.length() == 0){
            return 0;
        }
        else if(str.length() == 1){
            arr[num_words] = str.charAt(0);
        }
        return count+1;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        char str_array[] = new char[str.length()];
        int count = countWords(str, str_array);
```

```

        System.out.print(count + " ");
        for(int i = 0; i < str_array.length; i++){
            if(str_array[i] == 0)
                break;
            else
                System.out.print(str_array[i] + "" + (int)str_array[i] + " ");
        }
    }
}

```

SET-C

Given a string, S, find and return the highest occurring character present in the given string.

If there are 2 characters in the input string with the same frequency, return the character which comes first.

Note : Assume all the characters in the given string are either uppercase or lowercase or both. Spaces are not considered as characters therefore should not be outputted or counted. If length is zero then print count only.

Input format :

String S

Output format :

Highest occurring character

Constraints :

$0 \leq |S| \leq 10^7$

where |S| represents the length of string, S.

Sample Input 1:

H M N RR SS rr

Sample Output 1:

2 R

Sample Input 2:

Hello

Sample Output 2:

2 l

Solution:

```
import java.util.Scanner;
```

```

class Highest_Occur {
    public static void highest_occur_char(String inputString, int[] char_arr)
    {
        // Write your code here
        int arr[]=new int[256];
        for(int i=0;i<inputString.length();i++){
            if(inputString.charAt(i) != ' ')
                arr[inputString.charAt(i)]=arr[inputString.charAt(i)]+1;
        }
    }
}

```

```

    }

    int largest=Integer.MIN_VALUE;
    for(int j=0;j<inputString.length();j++){
        int k=arr[inputString.charAt(j)];
        if(k > largest){
            largest=k;
            char_arr[0] = largest;
            char_arr[1]=inputString.charAt(j);
        }
    }
}

}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    int char_arr[] = new int[2];
    highest_occur_char(str, char_arr);
    if(str.length() != 0)
        System.out.print(char_arr[0] + " " + (char)char_arr[1]);
    else
        System.out.print(char_arr[0]);
}
}

```

SET-B

Problem Statement

You are given provided with a string S and you have to reverse case i.e. all lower-cased characters should be upper-cased and all upper-cased character should be lower-cased.

Input Format

The first line of the input contains a string S.

Output format

Print a string after reversing the case.

Constraints

$1 \leq S.length \leq 1000$

Time Limit

1 second

Example

Sample Input
Happy Birthday

Sample Output
hAPPY bIRTHDAY

Default Code

```
import java.util.*;  
import java.io.*;
```

```
public class Main {  
    public static void main(String args[]) throws IOException {  
  
        //write your code here  
  
    }  
}
```

SOLUTION

```
import java.util.*;  
import java.io.*;
```

```
public class Main  
{  
    static String reverseCase(String str)  
    {  
        String result = "";  
  
        for (char c : str.toCharArray()) {  
            if (Character.isUpperCase(c)) {  
                result += Character.toLowerCase(c);  
            } else {  
                result += Character.toUpperCase(c);  
            }  
        }  
        return result;  
    }  
}
```

```
    public static void main(String[] args) throws IOException  
    {  
        Scanner scanner = new Scanner(System.in);  
  
        String S = scanner.nextLine();
```



```

        String ans = reverseCase(S);
        System.out.println(ans);

        scanner.close();
    }
}

```

SET-A

Write a Java program to reverse each word of a string individually.

Input Format :

String S

Output Format :

Modified reversed string

Constraints :

$0 \leq |S| \leq 10^7$

where $|S|$ represents the length of string, S.

Sample Input 1:

Welcome to Chitkara

Sample Output 1:

emocleW ot araktihC

Sample Input 2:

Give proper names to Class

Sample Output 2:

eviG reporp seman ot ssalC

Solution:

```

import java.util.Scanner;
public class Reverse_String {
    public static String reverseWord(String str){
        String words[]=str.split("\\s");
        String reverseWord="";
        for(String w:words){
            StringBuilder sb=new StringBuilder(w);
            sb.reverse();
            reverseWord+=sb.toString()+" ";
        }
        return reverseWord.trim();
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        if(str.length() != 0)
            System.out.print(reverseWord(str));
        else
            System.out.print("");
    } }

```