ASSIGNMENT – 1

Lab Practice Problems on JAVA

Name – Ashutosh Soni
Id- 2018ucp1505

Question 1: Given two matrices A and B find its product using JAVA where A and B are Cartan matrices.
A Cartan matrix is a square integer matrix who elements (Ai, j) satisfy the following conditions.

- Ai, j is an integer, one of belongs to {-3, -2, -1, 0, 2}.
- Ai, j = 2 the diagonal entries are all 2.
- Ai, j ≤0 of the diagonal.
- Ai, j =0 iff Aj, i =0.

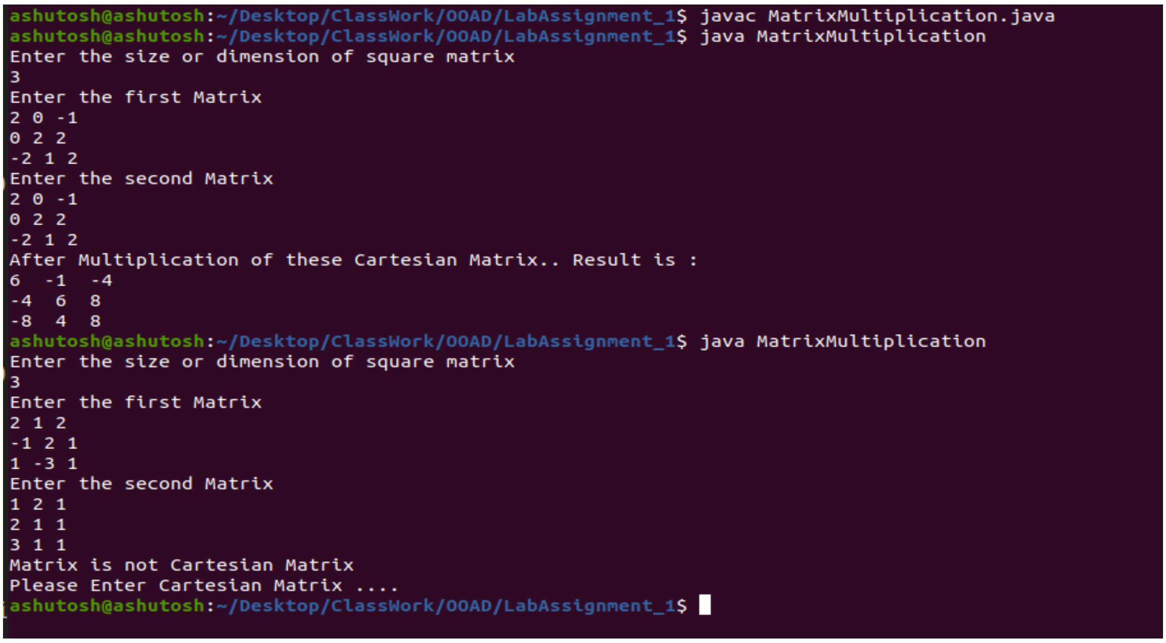
Answer:

Program in Java:

```
// Importing statements
import java.util.*;
public class MatrixMultiplication{
        public static void main(String []args){
                // Taking size of Matrix as Input
                System.out.println("Enter the size or dimension of square matrix");
                Scanner sc = new Scanner(System.in);
                int size = sc.nextInt();

                int matrixA[][] = new int[size][size];
                int matrixB[][] = new int[size][size];
                int result[][] = new int [size][size];

                System.out.println("Enter the first Matrix");
                for(int i=0;i<size;i++){
                        for(int j=0;j<size;j++){
                                matrixA[i][j] = sc.nextInt();
                        }
                }

                System.out.println("Enter the second Matrix");
                for(int i=0;i<size;i++){
                        for(int j=0;j<size;j++){
                                matrixB[i][j] = sc.nextInt();
                        }
                }

                if(checkIsMatrixCartesian(matrixA) && checkIsMatrixCartesian(matrixB)){
                        for(int i=0;i<size;i++){
                                for(int j=0;j<size;j++){
                                        int sum =0;
```

```
                    for(int k=0;k<size;k++){
                        sum += matrixA[i][k]*matrixB[k][j];
                    }
                    result[i][j] = sum;
                }
            }
            System.out.println("After Multiplication of these Cartesian Matrix.. Result is : ");
            for(int i=0;i<size;i++){
                for(int j=0;j<size;j++){
                    System.out.print(result[i][j] + "  ");
                }
                System.out.println(" ");
            }
        }
        else{
            System.out.println("Please Enter Cartesian Matrix ....");
        }
    }

    // Checking for Matrix to be Cartesian Matrix
    static boolean checkIsMatrixCartesian(int matrix[][]){
        for(int i=0;i<matrix.length;i++){
            if(matrix[i][i] != 2){
                System.out.println("Matrix is not Cartesian Matrix");
                return false;
            }
        }
        for(int i=0;i<matrix.length;i++){
            for(int j=0;j<matrix[i].length;j++){
                if(i==j){
                    continue;
                }
                else{
                    if((matrix[i][j] >2 || matrix[i][j]<-3) && matrix[i][j]==1){
                        System.out.println("Matrix is not Cartesian Matrix");
                        return false;
                    }
                    if(matrix[i][j] ==0 && matrix[j][i]!=0){
                        System.out.println("Matrix is not Cartesian Matrix");
                        return false;
                    }
                }
            }
        }
        return true;
    }
}
```

Output on Terminal:



```
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac MatrixMultiplication.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java MatrixMultiplication
Enter the size or dimension of square matrix
3
Enter the first Matrix
2 0 -1
0 2 2
-2 1 2
Enter the second Matrix
2 0 -1
0 2 2
-2 1 2
After Multiplication of these Cartesian Matrix.. Result is :
6  -1  -4
-4  6  8
-8  4  8
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java MatrixMultiplication
Enter the size or dimension of square matrix
3
Enter the first Matrix
2 1 2
-1 2 1
1 -3 1
Enter the second Matrix
1 2 1
2 1 1
3 1 1
Matrix is not Cartesian Matrix
Please Enter Cartesian Matrix ....
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$
```

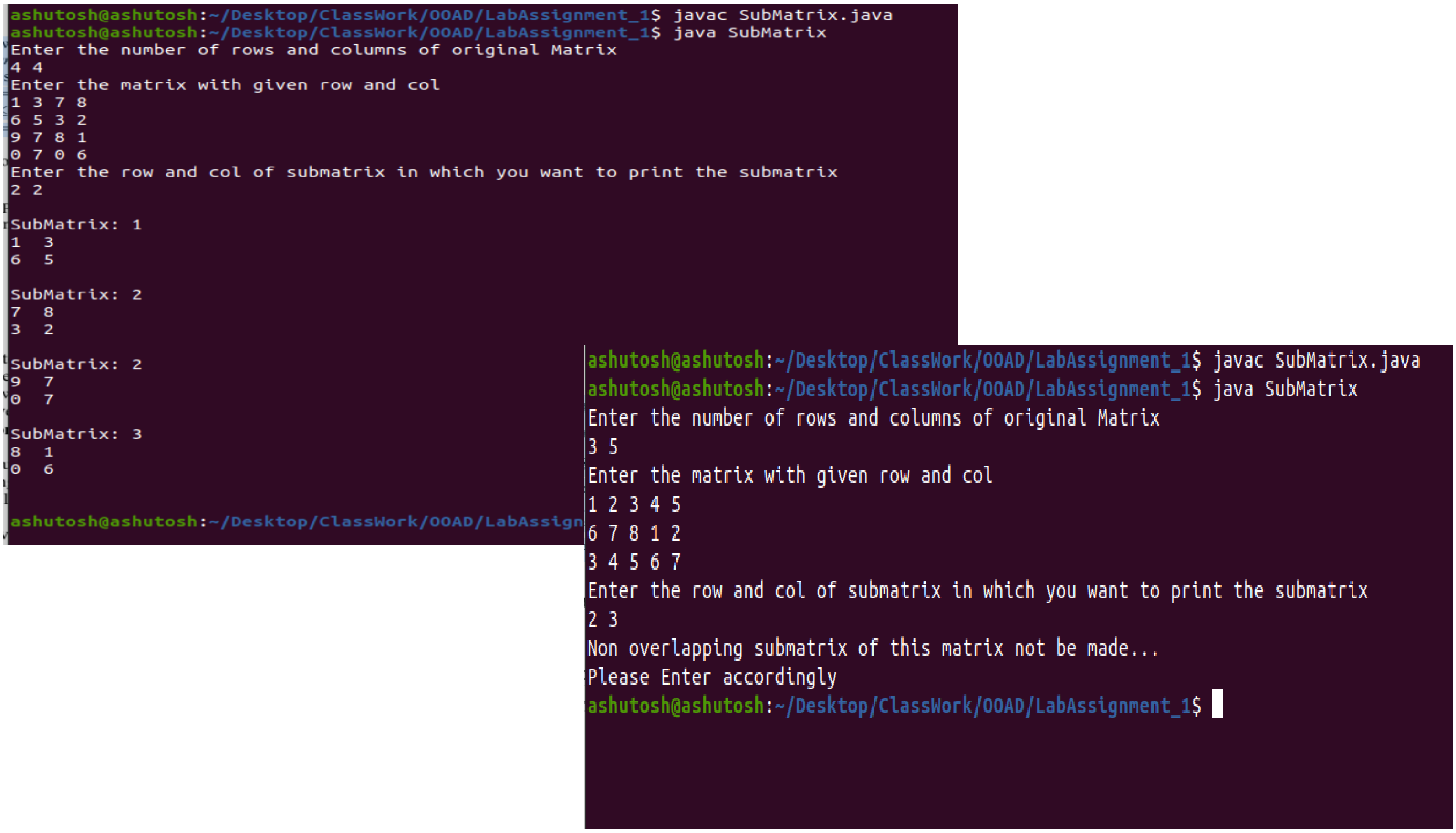
Question 2: WAP to find the non-overlapping sub-matrix of the given matrix of given order.

Answer:

Program in Java:

```
import java.util.*;
public class SubMatrix{
        public static void main(String []args){
                int row,col;
                System.out.println("Enter the number of rows and columns of original Matrix");
                Scanner sc = new Scanner(System.in);
                row = sc.nextInt();
                col = sc.nextInt();
                int originalMatrix[][] = new int[row][col];
                System.out.println("Enter the matrix with given row and col ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                originalMatrix[i][j] = sc.nextInt();
                        }
                }
                int newrow, newcol;
                System.out.println("Enter the row and col of submatrix in which you want to print the
submatrix");
                newrow = sc.nextInt();
                newcol = sc.nextInt();
                if(row%newrow == 0 && col%newcol == 0){
                        System.out.println("");
                        printSubmatrix(originalMatrix,newrow,newcol);
                        System.out.println("");
                }
                else{
                        System.out.println("Non overlapping submatrix of this matrix not be made...");
                        System.out.println("Please Enter accordingly");
                }
        }
        static void printSubmatrix(int matrix[][],int newrow,int newcol){
                int row = matrix.length;
                int col = matrix[0].length;
                int numrow = row/newrow;
                int numcol = col/newcol;
                for(int a=0;a<numrow;a++){
                        for(int b=0;b<numcol;b++){
                                System.out.println("SubMatrix: "+(a+b+1));
                                for(int i=0;i<newrow;i++){
                                        for(int j=0;j<newcol;j++){
                                                System.out.print(matrix[i+a*newrow][j+b*newcol]+ "  ");
                                        }
                                        System.out.println("");
                                }
                                System.out.println("");
                        }
                }
        }
}
```

Output on the Terminal:



```
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac SubMatrix.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java SubMatrix
Enter the number of rows and columns of original Matrix
4 4
Enter the matrix with given row and col
1 3 7 8
6 5 3 2
9 7 8 1
0 7 0 6
Enter the row and col of submatrix in which you want to print the submatrix
2 2

SubMatrix: 1
1  3
6  5

SubMatrix: 2
7  8
3  2

SubMatrix: 2
9  7
0  7

SubMatrix: 3
8  1
0  6


ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssign
```

```
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac SubMatrix.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java SubMatrix
Enter the number of rows and columns of original Matrix
3 5
Enter the matrix with given row and col
1 2 3 4 5
6 7 8 1 2
3 4 5 6 7
Enter the row and col of submatrix in which you want to print the submatrix
2 3
Non overlapping submatrix of this matrix not be made...
Please Enter accordingly
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ 
```

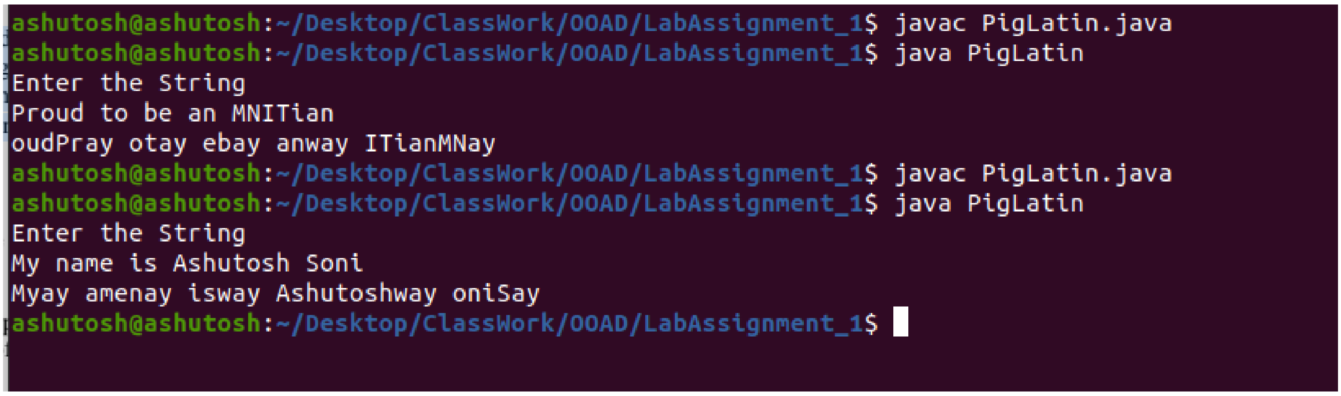
<u>Question 3</u>: Write a Java program that converts an English word into PigLatin. To do that there are three rules: if the word starts with a vowel add way to the end, ex. apple=appleway. If the word has a vowel but doesn't start with it then take the consonants it front of the first vowel and put them to the end of the word and add ay to the end. ex: ball=allbay, strong=ongstray. and if the word has no vowels just add ay to the end. ex. pfft=pfftay.

<u>Answer:</u>

<u>Program in Java:</u>

```
import java.util.*;
import java.lang.String;
public class PigLatin{
        public static void main(String []args){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the String");
                String str = sc.nextLine();
                int len = str.length();
                String temp;
                int start=0;
                for(int i=0;i<len;i++){
                        if(str.charAt(i)==' '){
                                temp=str.substring(start,i);
                                start=i+1;
                                printPigLatin(temp);
                        }
                }
                temp=str.substring(start,len);
                printPigLatin(temp);
                System.out.println(" ");
        }
        public static void printPigLatin(String temp){
                if(isVowel(temp.charAt(0))){
                        String finalString = temp + "way";
                        System.out.print(finalString + " ");
                }
                else{
                        boolean flag = false;
                        for(int j=1;j<temp.length();j++){
                                if(isVowel(temp.charAt(j))){
                                        String test1 = temp.substring(0,j);
                                        String test2 = temp.substring(j,temp.length());
                                        String finalString = test2 + test1 + "ay";
                                        System.out.print(finalString + " ");
                                        flag=true;
                                        break;
                                }
                        }
                if(!flag){
                    String finalString = temp + "ay";
                    System.out.print(finalString + " ");
                }
            }
        }
        public static boolean isVowel(char c){
            if(c=='A' || c=='a' || c=='E' || c=='e' || c=='I' || c=='i' || c=='O' || c=='o' || c=='U' || c=='u'){
                return true;
            }
            return false;
        }
    }
```

<u>Output on the terminal:</u>



```
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac PigLatin.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java PigLatin
Enter the String
Proud to be an MNITian
oudPray otay ebay anway ITianMNay
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac PigLatin.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java PigLatin
Enter the String
My name is Ashutosh Soni
Myay amenay isway Ashutoshway oniSay
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$
```

<u>Question 4</u>: Provided that you have a given number of small rice bags (1 kilo each) and big rice bags (5 kilo each), write a method that returns the minimum number of small bags necessary to package goal kilos of rice. Return -1 if it is not possible to package the required rice amount with the bags provided.

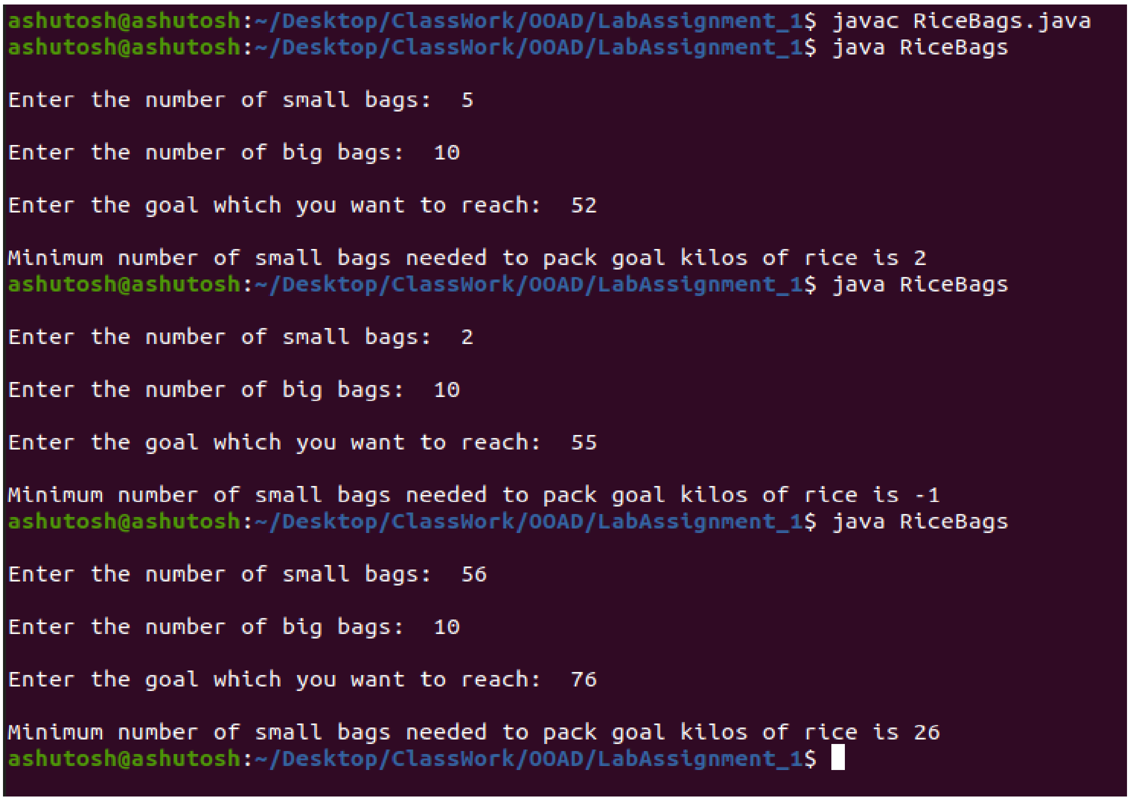
<u>Answer:</u>

<u>Program in Java:</u>

```
    import java.util.*;
    public class RiceBags{
        public static void main(String []args){
            Scanner sc = new Scanner(System.in);
            int smallBags=0,bigBags=0,goal=0;
            System.out.print("\nEnter the number of small bags:  ");
            smallBags = sc.nextInt();
            System.out.print("\nEnter the number of big bags:  ");
            bigBags = sc.nextInt();
            System.out.print("\nEnter the goal which you want to reach:  ");
            goal = sc.nextInt();
            int result = findBags(smallBags,bigBags,goal);
            System.out.println("\nMinimum number of small bags needed to pack goal kilos of rice is "+
    result);
```

```
        }

        public static int findBags(int smallBags,int bigBags,int goal){
                int result = -1;
                if(bigBags*5 >= goal){
                   if(smallBags >= goal%5){
                      result = goal%5;
                   }
                }
                else{
                   if(smallBags >= goal-bigBags*5){
                      result = goal-bigBags* 5;
                   }
                }
                return result;
        }
    }
```

Output on the Terminal:



```
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ javac RiceBags.java
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java RiceBags

Enter the number of small bags:  5

Enter the number of big bags:  10

Enter the goal which you want to reach:  52

Minimum number of small bags needed to pack goal kilos of rice is 2
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java RiceBags

Enter the number of small bags:  2

Enter the number of big bags:  10

Enter the goal which you want to reach:  55

Minimum number of small bags needed to pack goal kilos of rice is -1
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ java RiceBags

Enter the number of small bags:  56

Enter the number of big bags:  10

Enter the goal which you want to reach:  76

Minimum number of small bags needed to pack goal kilos of rice is 26
ashutosh@ashutosh:~/Desktop/ClassWork/OOAD/LabAssignment_1$ 
```