Solution.java

```
package mat;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

public class Solution {

        public static void main(String args[]) throws IOException, MatrixMismatchException {

                BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

                System.out.println("1. ADD-MATRIX () \t 2.ADD-MATRIX(SIZE) \t 3.ADD-MATRIX(ROW,COL) \t 4.MULTIPLY-MATRIX() \t 5.MULTIPLY-MATRIX(SIZE) \t 6.MULTIPLY-MATRIX(ROW,COL)");

                System.out.println("Enter choice  : ");

                int choice= Integer.parseInt(bf.readLine());

                Matrix mObj1,mObj2,mObj3;

                switch(choice){

                case 1:{

                        System.out.println("1st Matrix Elements : ");

                        mObj1 = new Matrix();

                        mObj1.read(bf);

                        System.out.println("2nd Matrix Elements : ");

                        mObj2 = new Matrix();

                        mObj2.read(bf);

                        mObj3 = mObj1.add(mObj2);

                        System.out.println(mObj3);

                        break;

                       }

                case 2:{


                                System.out.println(" Size of 1st matrix :");

                                int size = Integer.parseInt(bf.readLine());

                                mObj1 = new Matrix(size);

                                System.out.println("Matrix1 Elements : ");
                mObj1.read(bf);
                System.out.println("Size of 2nd matrix :");
                size = Integer.parseInt(bf.readLine());
    mObj2 = new Matrix(size);
    System.out.println("Matrix2 Elements : ");
    mObj2.read(bf);
    mObj3 = mObj1.add(mObj2);
    System.out.println(mObj3);
    break;
}
case 3:{
                System.out.println("Size of 1st matrix :");
                int size1 = Integer.parseInt(bf.readLine());
                int size2 = Integer.parseInt(bf.readLine());
                mObj1 = new Matrix(size1,size2);
                System.out.println("Matrix1 Elements : ");
    mObj1.read(bf);
    System.out.println("Size of 2nd matrix :");
    size1 = Integer.parseInt(bf.readLine());
                size2 = Integer.parseInt(bf.readLine());
    mObj2 = new Matrix(size1,size2);
    System.out.println("Matrix2 Elements : ");
    mObj2.read(bf);
    mObj3 = mObj1.add(mObj2);
    System.out.println(mObj3);
    break;
}
case 4:{
          System.out.println("1st Matrix Elements : ");
          mObj1 = new Matrix();
          mObj1.read(bf);
```

```
         System.out.println("2st Matrix Elements : ");
         mObj2 = new Matrix();
         mObj2.read(bf);
         mObj3 = mObj1.multiply(mObj2);
         System.out.println(mObj3);
         break;
        }
case 5:{
                System.out.println(" Size of 1st matrix :");
                int size = Integer.parseInt(bf.readLine());
                mObj1 = new Matrix(size);
                System.out.println("Matrix1 Elements : ");
                mObj1.read(bf);
                System.out.println("Size of 2nd matrix :");
                size = Integer.parseInt(bf.readLine());
    mObj2 = new Matrix(size);
    System.out.println("Matrix2 Elements : ");
    mObj2.read(bf);
    mObj3 = mObj1.multiply(mObj2);
    System.out.println(mObj3);
    break;
}
case 6:{
                System.out.println("Size of 1st matrix :");
                int size1 = Integer.parseInt(bf.readLine());
                int size2 = Integer.parseInt(bf.readLine());
                mObj1 = new Matrix(size1,size2);
                System.out.println("Matrix1 Elements : ");
    mObj1.read(bf);
    System.out.println("Size of 2nd matrix :");
    size1 = Integer.parseInt(bf.readLine());
```

```
                                 size2 = Integer.parseInt(bf.readLine());
                     mObj2 = new Matrix(size1,size2);
                     System.out.println("Matrix2 Elements : ");
                     mObj2.read(bf);
                     mObj3 = mObj1.multiply(mObj2);
                     System.out.println(mObj3);
                     break;
                 }
                 }


         }
}
```

MatrixMismatchException.java

```
package mat;


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

Matrix.java

```
package mat;
import java.io.BufferedReader;
import java.io.IOException;
public class Matrix {
         public Matrix add(Matrix another) throws MatrixMismatchException {
          Matrix output;
          if(this.input.length == another.input.length)
          {
                 output = new Matrix(this.input.length,another.input.length);
                 for(int i= 0;i<this.input.length;i++)
```

```
       {
           for(int j =0;j<this.input[i].length;j++)
           {
               output.input[i][j] = this.input[i][j] + another.input[i][j];
           }
       }
   }
  else {
 throw new MatrixMismatchException("Length Varies in one of the set you have passed");
}
   return output;
}
   public Matrix multiply(Matrix another) throws MatrixMismatchException{
       Matrix output;
       int sum = 0;
       if(this.input[0].length == another.input.length){
           output = new Matrix(this.input.length,another.input[0].length);
           for(int i=0; i<this.input.length;i++)
           {
               for(int j=0;j<another.input[0].length;j++)
               {
                   for(int k = 0;k<another.input.length ;k++)
                   {
                       sum = sum + this.input[i][k] * another.input[k][j];
                   }
                   output.input[i][j] = sum;
                   sum = 0;
               }
           }
       }
      else {
```

```
                         throw new MatrixMismatchException("Length Varies in one of the set you have
passed");
                       }
                      return output;
           }
    private int input[][];
    public Matrix(){
           this.input = new int [2][2];
    }
    public Matrix(int size){
           this.input = new int [size][size];
    }
          public Matrix(int size1, int size2) {
       this.input = new int[size1][size2];
          }
          public void read(BufferedReader bf) throws IOException {
       for(int i = 0; i < this.input.length; i++) {
            for(int j = 0;j <this.input[i].length;j++) {
          this.input[i][j] = Integer.parseInt(bf.readLine());
            }
       }
     }
    public String toString() {
       String ret = "[";
       for(int i = 0; i < this.input.length; i++) {
            for(int j = 0; j<this.input[i].length; j++){
          ret += this.input[i][j] + " ,";
            }
            ret = ret.substring(0, ret.length()-2);
            ret += "\n";
       }
```

```
        ret = ret.substring(0, ret.length()-1);
        ret += "]";
        return ret;
    }
}
```