Rajalakshmi Engineering College

Name: Nandhini Velmurugan
Email: 241901063@rajalakshmi.edu.in
Roll no: 241901063
Phone: 9043367699
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 3_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Rina is managing the inventory for a library, where each row of a 2D matrix represents the number of different genres of books available on each shelf.

She wants to perform the following operations:

Transformation: Replace each element in a row with the sum of all elements in that row.Merging: After transformation, Rina will provide one additional matrix, and specify whether to merge the transformed matrix with this new matrix row-wise or column-wise.

Input Format

The first line contains two integers R and C, representing the number of rows and columns of the initial matrix.

The next R lines contain C space-separated integers, representing the book counts in the library.

The next line contains two integers MR and MC, representing the dimensions of the second matrix (to be merged).

The next MR lines contain MC space-separated integers, representing the second matrix.

The last line contains an integer mergeType:

- 0 Row-wise merging (append the second matrix below the transformed matrix).
- 1 Column-wise merging (append the second matrix to the right of the transformed matrix).

Output Format

The output prints "Transformed matrix: "followed by the transformed 2D matrix where each element in a row is replaced with the sum of the elements in that row.

The output prints "Final merged matrix: ", followed by the merging based on mergeType.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 4
8 2 4 9
4 5 6 1
7 8 9 3
2 4
3 5 7 2
6 1 4 9
0
Output: Transformed matrix:
23 23 23 23
16 16 16 16

27 27 27 27
Final merged matrix:
23 23 23 23
16 16 16 16
27 27 27 27
3 5 7 2
6 1 4 9

Answer

```
import java.util.*;

public class Main{
   public static int[][] transform(int[][] matrix, int R, int C){
      int[][] transformed = new int[R][C];
      for (int i = 0; i < R; i ++){
         int rowsum = 0;
         for (int j = 0; j < C; j ++){
            rowsum += matrix[i][j];
         }
         Arrays.fill(transformed[i], rowsum);
      }
      return transformed;
   }
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int R = sc.nextInt();
      int C = sc.nextInt();
      int[][] matrix1 = new int[R][C];
      for (int i = 0; i < R; i ++){
         for (int j = 0; j < C; j ++){
            matrix1[i][j] = sc.nextInt();
         }
      }
      int[][] transformed = transform(matrix1, R, C);
      int MR = sc.nextInt();
      int MC = sc.nextInt();
      int[][] matrix2 = new int[MR][MC];
      for (int i = 0; i < MR; i++){
         for (int j = 0; j < MC; j ++){
            matrix2[i][j] = sc.nextInt();
         }
      }
```

```
        int merge = sc.nextInt();
        System.out.println("Transformed matrix:");
        for (int i = 0; i < R; i ++){
            for (int j = 0; j < C; j ++){
                System.out.print(transformed[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("Final merged matrix:");
        if (merge == 0){
            for (int i = 0; i < R; i++){
                for (int j = 0; j < C; j ++){
                    System.out.print(transformed[i][j] + " ");
                }
                System.out.println();
            }
            for (int i = 0; i < MR; i ++){
                for (int j = 0; j < MC; j ++){
                    System.out.print(matrix2[i][j] + " ");
                }
                System.out.println();
            }
        } else{
            for (int i = 0; i < R; i ++){
                for (int j = 0; j < C; j ++){
                    System.out.print(transformed[i][j] + " ");
                }
                for (int j = 0; j < MC; j ++){
                    System.out.print(matrix2[i][j] + " ");
                }
                System.out.println();
            }
        }
        sc.close();
    }
}
```

Status : Correct **Marks : 10/10**

2. Problem Statement:

Mason is participating in a coding challenge where he must manipulate an integer array. His task is to replace every element in the array with the next greatest element to its right. The last element of the array remains unchanged, as there is no element to its right.

Your job is to help Mason write a program that performs this transformation and outputs the modified array.

Input Format

The first line of input contains an integer n representing the number of elements in the array.

The second line of input contains n space-separated integers representing the elements of the array.

Output Format

The output prints the modified array of n integers, where each element (except the last one) is replaced by the maximum element to its right, and the last element remains unchanged.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
12 3 91 15 12 14
Output: 91 91 15 14 14 14

Answer

```
import java.util.*;

public class Main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] arr = new int[n];
      for (int i = 0; i < n; i ++){
         arr[i] = sc.nextInt();
```

```
        }
        int maxfromright = arr[n-1];
        for (int i = n-2; i >= 0; i --){
            int temp = arr[i];
            arr[i] = maxfromright;
            if (temp > maxfromright){
                maxfromright = temp;
            }
        }
        for (int i = 0; i < n; i ++){
            System.out.print(arr[i] + " ");
        }
    }
}
```

Status : Correct **Marks : 10/10**

3. Problem Statement

Emma is a data analyst working with a grid-based system where each cell contains important numerical data. The grid represents spatial data, inventory records, or structured reports that require periodic updates.

Due to system updates and new requirements, Emma needs to modify the grid in the following ways:

She wants to insert either a new row or a new column at a given position.Later, she needs to delete either a row or a column from the modified matrix.

Input Format

The first line contains two integers rows and cols (the dimensions of the matrix).

The next rows lines contain cols space-separated integers representing the initial matrix.

The next line contains two integers insertType and insertIndex:

- insertType = 0 for row insertion, 1 for column insertion.
- insertIndex is the position where the new row/column should be added.

If inserting a row, the next cols integers represent the new row or If inserting a column, the next rows integers represent the new column.

The next line contains two integers deleteType and deleteIndex:

- deleteType = 0 for row deletion, 1 for column deletion.
- deleteIndex is the position to be deleted.

Output Format

The first line of output prints the string "After insertion" followed by the modified matrix with the inserted row or column.

Each row of the matrix is printed on a new line with space-separated integers.

The next line prints the string "After deletion" followed by the final matrix after the specified deletion operation.

Each row of the resulting matrix is printed on a new line with space-separated integers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 3
1 2 3
4 5 6
7 8 9
0 1
10 11 12
1 2
Output: After insertion
1 2 3
10 11 12
4 5 6
7 8 9
After deletion
1 2
10 11

4 5
7 8

Answer

```
import java.util.*;

public class Main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int rows = sc.nextInt();
      int cols = sc.nextInt();
      int[][] matrix = new int[rows][cols];
      for (int i = 0; i < rows; i ++){
         for (int j = 0; j < cols; j ++){
            matrix[i][j] = sc.nextInt();
         }
      }
      int insertype = sc.nextInt();
      int insertindex = sc.nextInt();
      if (insertype == 0){
         int[] newrow = new int[cols];
         for (int j = 0; j < cols; j ++){
            newrow[j] = sc.nextInt();
         }
         int[][] newmatrix = new int[rows + 1][cols];
         for (int i = 0, k = 0; i < rows + 1; i++){
            if (i == insertindex){
               newmatrix[i] = newrow;
            } else{
               newmatrix[i] = matrix[k++];
            }
         }
         matrix = newmatrix;
         rows++;
      } else{
         int[] newcol = new int[rows];
         for (int i = 0; i < rows; i ++){
            newcol[i] = sc.nextInt();
         }
         int[][] newmatrix = new int[rows][cols + 1];
         for (int i = 0; i < rows; i ++){
            for (int j = 0, k = 0; j < cols+1; j ++){
```

```
            if (j == insertindex){
                newmatrix[i][j] = newcol[i];
            } else{
                newmatrix[i][j] = matrix[i][k++];
            }
        }
    }
    matrix = newmatrix;
    cols++;
}
System.out.println("After insertion");
for (int i = 0; i < rows; i ++){
    for (int j = 0; j < cols; j++){
        System.out.print(matrix[i][j] + " ");
    }
    System.out.println();
}
int deletetype = sc.nextInt();
int deleteindex = sc.nextInt();
if (deletetype == 0){
    int[][] newmatrix = new int[rows-1][cols];
    for (int i = 0, k = 0; i < rows; i ++){
        if (i == deleteindex) continue;
        newmatrix[k++] = matrix[i];
    }
    matrix = newmatrix;
    rows--;
} else{
    int[][] newmatrix = new int[rows][cols-1];
    for (int i = 0; i < rows; i ++){
        for (int j = 0, k = 0; j < cols; j ++){
            if (j == deleteindex) continue;
            newmatrix[i][k++] = matrix[i][j];
        }
    }
    matrix = newmatrix;
    cols--;
}
System.out.println("After deletion");
for (int i = 0; i < rows; i ++){
    for (int j = 0; j < cols; j ++){
        System.out.print(matrix[i][j] + " ");
```

```
            }
            System.out.println();
        }
        sc.close();
    }
}
```

Status : Correct **Marks : 10/10**

4. Problem Statement:

Imagine you have an array of integer values, and you're tasked with identifying a pair of elements within the array. This pair of elements should have a sum that is the closest to zero when compared to any other pair in the array.

Your goal is to create a program that solves this problem efficiently. The program should accept an array of integers and return the pair of elements whose sum is closest to zero.

Input Format

The first line of the input is an integer N representing the size of the array.

The second line of the input contains N space-separated integer values.

Output Format

The output is displayed in the following format:

"Pair with the sum closest to zero: {value} and {value}"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
9 10 -3 -5 -2
Output: Pair with the sum closest to zero: 9 and -5

Answer

```
import java.util.*;

public class Main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] arr = new int[n];
      for (int i = 0; i < n; i ++){
         arr[i] = sc.nextInt();
      }
      int pair1 = arr[0] , pair2 = arr[n-1];
      int minsum = pair1 + pair2;
      for (int i = 0; i < n - 1; i ++){
         for (int j = i + 1; j < n; j ++){
            int sum = arr[i] + arr[j];
            if (Math.abs(sum) < Math.abs(minsum)){
               minsum = sum;
               pair1 = arr[i];
               pair2 = arr[j];
            }
         }
      }
      System.out.println("Pair with the sum closest to zero: " + pair1 + " and " +
pair2);
   }
}
```

Status : Correct ***Marks : 10/10***