

PSG COLLEGE OF TECHNOLOGY, COIMBATORE
DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES
20XW57 – JAVA PROGRAMMING LAB

PROBLEM SHEET I – ARRAYS (1D &2D)

1. Write a Java program to perform linear search in the following array

int a[] = {23, 45, 86, 77, 9, 54, 13, 36, 61, 98}

The element to be searched int x = 54

```
public class Main {  
    public static int search(int arr[], int x)  
    {  
        int n = arr.length;  
        for (int i = 0; i < n; i++)  
        {  
            if (arr[i] == x)  
                return i;  
        }  
        return -1;  
    }  
    public static void main(String[] args) {  
        int arr[] = {23, 45, 86, 77, 9, 54, 13, 36, 61, 98};  
        int x = 54;  
        int result = search(arr, x);  
        if (result == -1)  
            System.out.print("Element " + x + " is not present in the array");  
        else  
            System.out.print("Element " + x + " is present at index " + result);  
    }  
}
```

2. Write a function rotate(ar[], d, n) that rotates arr[] of size n by d elements.

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Rotation of the above array by 2 will make array

3	4	5	6	7	1	2
---	---	---	---	---	---	---

```

public class Main {

    public static void rotate(int arr[], int d, int n)
    {
        int p = 1;
        while (p <= d) {
            int last = arr[0];
            for (int i = 0; i < n - 1; i++) {
                arr[i] = arr[i + 1];
            }
            arr[n - 1] = last;
            p++;
        }

        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
    }

    public static void main(String[] args)
    {
        int arr[] = { 1, 2, 3, 4, 5, 6, 7 };
        int N = arr.length;
        int d = 6;
        rotate(arr, d, N);
    }
}

```

3. Given an array of integers, our task is to write a program that efficiently finds the second largest element present in the array.

Input: arr[] = {12, 35, 1, 10, 34, 1}

Output: The second largest element is 34.

4. Write a Java program to perform binary search in the following array

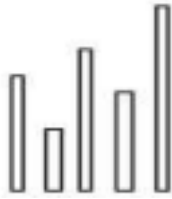
int a[] = {3, 15, 26, 37, 49, 55, 63, 76, 81, 93}

The element to be searched int x = 76. Also display the number of comparisons needed to search this element.

5. Write a Java program to perform selection sort in the following array

int a[] = {46, 89, 97, 4, 62, 13, 71, 58, 25, 30}

6. Given an array representing heights of buildings. The array has buildings from left to right as shown in below diagram, count number of buildings facing the sunset. It is assumed that heights of all buildings are distinct and display their heights.



Input : arr[] = {7, 4, 8, 2, 9}

Output:

Count: 3

Buildings : 7, 8, 9

7. Write a Java program to perform linear search in the following two dimensional array int a[][] = {{33, 25, 16, 7},
 {8, 64},
 {53, 96, 81, 78},
 {83, 15}}

The element to be searched int x = 96

8. Write a Java program to perform multiplication of two matrixes where first matrix has m rows, n columns and second matrix has n rows and p columns.

9. Given a 2D array, print it in spiral form. See the following examples.

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Output: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

8)Sorting as Twisted Matrix

A matrix of same rows and same columns is given. Pooja wants to Sort the matrix in such a manner that she will start from the first element and traverse the matrix in clockwise manner at the end, and she should be at the middle position with a largest element. Find the solution of Pooja's problem.

Input Format:

First line will take an integer input as dimension of matrix i.e. N x N (only one input N). Second input the elements of matrix.

Output Format:

Sorted Matrix in clockwise manner.

(For more description, see the explanation)

Constraints :

0<N<10

Sample Input:

```
3
2 5 12
22 45 55
```

1 6 8

Sample Output:

1 2 5
45 55 6
22 12 8

Explanation:

The output is as below twisted matrix contains sorted elements in clockwise order.

1	2	5
45	55	6
22	12	8

Example

4	2 4 5 6
99 67 54 77	55 60 67 12
34 55 12 2	54 99 77 31
4 5 6 31	50 49 34 32
32 49 50 60	

9)Game on strings

A 7 year old boy surya is very good in playing with jumbling english alphabets in a string.He found out a new way,a kind of encryption,from a given string S1 by his way he can move to string S2.Both S1 and S2 contain the same number of letters.

In a single day, surya can increase any letter of s1 by one, that is, he can convert letter A to B ,B to C,and so on. He can also convert letter Z to letter A.And also he has ability to move a letter by 13 i.e A to N in a single day.

write a program to know the minimum number of days in which he can convert the move S1 into move S2?

Input and Output Format:

First line of input corresponds to length of a string.

Second line of input corresponds to a string S1.

Third line of input corresponds to a string S2.

Output belongs to number of days he takes to convert S1 to S2.

Sample Input 1:

5
ABCDE
BCDEF

Sample Output 1:

5

Sample Input 2:

1
Z
A

Sample Output 2:

1

5	5
ABCDE	
BCDEF	

10) Skyline of Building's

Given n rectangular buildings in a 2-dimensional city, computes the skyline of these buildings, eliminating hidden lines. The main task is to view buildings from a side and remove all sections that are not visible. All buildings share a common bottom and every building is represented by triplet (left, ht, right) 'left': is x coordinate of left side (or wall).

'right': is x coordinate of right side

'ht': is the height of a building.

A skyline is a collection of rectangular strips. A rectangular strip is represented as a pair (left, ht) where the left is the x coordinate of the left side of the strip and ht is height of strip.

So we have to find the skyline for the given building's. The main task is to view buildings from a side and remove all sections that are not visible.

Input Format:

The first line of input consists of N value which represents the number of buildings. Next N lines of inputs consist of triplet for building as left, height, and then right of same common bottom for all buildings.

Output Format:

Output consists of N skylines for each building.

A skyline is a collection of rectangular strips. A rectangular strip is represented as a pair (left, ht) where the left is the x coordinate of the left side of the strip and ht is height of strip.

Sample Input 1:

8
1 11 5
2 6 7
3 13 9
12 7 16
14 3 25
19 18 22
23 13 29
24 4 28

Sample Output 1:

1,11
3,13

9,0
12,7
16,3
19,18
22,3
23,13
29,0

Sample Input 2:

1
1 11 5

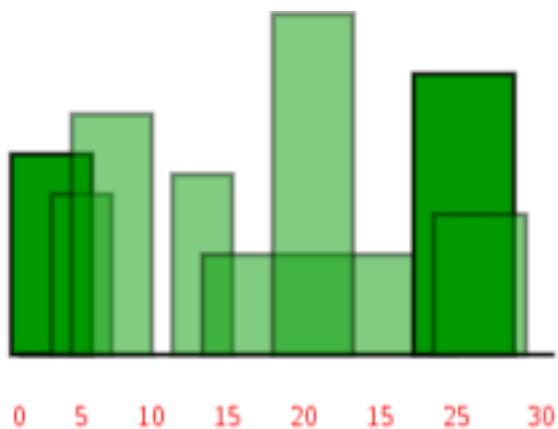
Sample Output 2:

1,11
5,0

Explanation of the sample input 1:

The idea is similar to merge of merge sort, start from first strips of two skylines, compare x coordinates. Pick the strip with smaller x coordinate and add it to result. The height of added strip is considered as maximum of current heights from skyline1 and skyline2.

Below image is for input 1:



Example

6	2,4
2 4 6	6,0
7 5 9	7,5
11 10 14	9,0
15 5 18	11,10
19 6 24	14,0
26 7 30	15,5
	18,0
	19,6

24,0 26,7 30,0

11)Ferry Loading II

Before bridges were common, ferries were used to transport cars across rivers. River ferries, unlike their larger cousins, run on a guide line and are powered by the river's current. Cars drive onto the ferry from one end, the ferry crosses the river, and the cars exit from the other end of the ferry. There is a ferry across the river that can take n cars across the river in t minutes and return in t minutes. m cars arrive at the ferry terminal by a given schedule. Find the earliest time that all the cars can be transported across the river. Find the minimum number of trips that the operator must make to deliver all cars by

that time. **Input format:**

Input begins with $n(\text{int})$, $t(\text{int})$, $m(\text{int})$. m lines follow, each giving the arrival time for a car (in minutes since the beginning of the day). The operator can run the ferry whenever he or she wishes, but can take only the cars that have arrived up to that time.

Output format:

For each test case, output a single line with two integers: the time, in minutes since the beginning of the day, when the last car is delivered to the other side of the river, and the minimum number of trips made by the ferry to carry the cars within that time. You may assume that $0 < n, t, m < 1440$. The arrival times for each test case are in non-decreasing order.

Sample Input and Output:

[All text in bold corresponds to input and the rest corresponds to output]

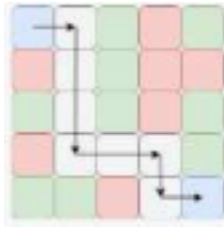
```
2 10 10
10
20
30
40
50
60
70
80
90
100 5
```

12)Given a 2D array($m \times n$). The task is to check if there is any path from top left to bottom right. In the matrix, -1 is considered as blockage (can't go through this cell) and 0 is considered path cell (can go through it).

Input :

```
arr[][] = {{ 0, 0, 0, -1, 0},
{-1, 0, 0, -1, -1},
{ 0, 0, 0, -1, 0},
{-1, 0, 0, 0, 0},
{ 0, 0, -1, 0, 0}}
```

Output : Yes



Input :

```
arr[][] = {{ 0, 0, 0, -1, 0},
{-1, 0, 0, -1, -1},
{ 0, 0, 0, -1, 0},
{-1, 0, -1, 0, 0},
{ 0, 0, -1, 0, 0}}
```

Output : No



13) Create a class named **ItemType** with the following private attributes.

Attributes	Datatype
name	String
deposit	Double
costPerDay	Double

Include appropriate getters and setters for the class.

Include default and parameterized constructors for the class.

Create a class named **ItemTypeBO** class and include the following methods.

Method	Description
public void add(ItemType object ,ItemType[] itemTypeArray,Integer index)	This method accepts an array of ItemType objects, an ItemType object and the index at which the object is to be added. The method appends the object to the array of ItemType objects at the specified index and prints " New item added successfully "
public void search(String search, ItemType[] itemTypeArray)	This method accepts the search term and the array of ItemType objects as input. It displays the details of the ItemType if matched. else prints " Searched Item Type not found "
public void display(ItemType[] itemTypeArray)	This method accepts an array of ItemType objects as arguments. It displays the details one by one.

Create a driver class called **Main**. In the Main method, obtain input from the user and call the ItemTypeBO functions appropriately.

Input and Output format:

The size of the itemType Array should be 10. ($1 \leq n \leq 10$).

Display the value of the deposit amount correct to 1 decimal place.

The search statement "**Enter the Name of the item to be searched**" should be in the Main method. Refer to sample Input and Output for formatting specifications.

Sample Input and Output 1:

[All text in bold corresponds to the input and rest corresponds to output]

Enter the Number of Item Type

3

Enter the Item Type 1 Name

Electronics

Enter the Deposit Amount

2500

Enter the Cost per day

150

New item added successfully

Enter the Item Type 2 Name

Chemicals

Enter the Deposit Amount

500

Enter the Cost per day

50

New item added successfully

Enter the Item Type 3 Name

Construction

Enter the Deposit Amount

2000

Enter the Cost per day

500

New item added successfully

Item Type Number 1:

Name:Electronics

Deposit Amount:2500.0

Cost Per Day:150.0

Item Type Number 2:

Name:Chemicals
Deposit Amount:500.0
Cost Per Day:50.0
Item Type Number 3:
Name:Construction
Deposit Amount:2000.0
Cost Per Day:500.0
Enter the Name of the item to be searched

Construction

Searched Item Type is:
Name:Construction
Deposit Amount:2000.0
Cost Per Day:500.0

14)Create a class named **User** with following private attribute,

Attribute	Data Type
name	String
mobileNumber	String
username	String
password	String

Include appropriate getters/setters method, default constructor, and parameterized constructor **public User(String name, String mobileNumber, String username, String password)**.

Now create a class named **UserBO** for manipulating user objects, create an array for storing user objects in UserBO class.

Include the following methods in the **UserBO** class

Method	Description

public void addUser(User[] userArray,User userIns)	The method accepts an array of user objects and the new user object to be added as the arguments. This method adds the new user object to the array.
public void sortUsers(User[] userArray)	The method accepts an array of user objects as an argument. This method sort the array of user objects based

	on the name
public Boolean deleteUser(User[] userArray,String name)	The method accepts 2 arguments. The first argument is an array of user objects and the name of the user. This method deletes the specific user from the array and returns true if the user presents. Otherwise returns false.
public void displayAll(User[] userArray)	The method accepts an array of user objects. This method used to displays all the user details

Create a driver class called **Main** to test the above classes.

Input and Output format:

If the user to be deleted is not present print "**No user found with given name**" else print "**User deleted successfully**" in the main method.

Refer to Sample Input and Output for formatting specifications.

Sample Input and Output:

[All text in bold corresponds to the input and rest corresponds to output]

Enter the number of users:

2

Enter the details of User 1

Enter the name of the user:

John

Enter the mobile number of the user:

12345

Enter the username of the user:

john

Enter the password of the user:

john

Enter the details of User 2

Enter the name of the user:

Joe

Enter the mobile number of the user:

67890

Enter the username of the user:

joe

Enter the password of the user:

joe

User details as entered:

User Details:

User 1

Name:John

Mobile Number:12345

User 2

Name:Joe

Mobile Number:67890

After sorting:

User Details:

User 1

Name:Joe

Mobile Number:67890

User 2

Name:John

Mobile Number:12345

Enter the user to be deleted:

John

User deleted successfully

After Deleting:

User Details:

User 1

Name:Joe

Mobile Number:67890

15) Given an array of n integers where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are m students, the task is to distribute chocolate packets such that:

1. Each student gets one packet.
2. The difference between the number of chocolates in the packet with maximum chocolates and packet with minimum chocolates given to the students is minimum.

Examples:

Input : arr[] = {7, 3, 2, 4, 9, 12, 56} , m = 3

Output: Minimum Difference is 2

Explanation:

We have seven packets of chocolates and we need to pick three packets for 3 students. If we pick 2, 3 and 4, we get the minimum difference between maximum and minimum packet sizes.

Input : arr[] = {3, 4, 1, 9, 56, 7, 9, 12} , m = 5

Output: Minimum Difference is 6

Explanation:

The set goes like 3,4,7,9,9 and the output is $9-3 = 6$

Input : arr[] = {12, 4, 7, 9, 2, 23, 25, 41,

30, 40, 28, 42, 30, 44, 48,

43, 50} , m = 7

Output: Minimum Difference is 10

Explanation:

We need to pick 7 packets. We pick 40, 41, 42, 44, 48, 43 and 50 to minimize difference between maximum and minimum.