

## Array, Lesson-14

Q. Write a program to initialise the given data in an array and find the minimum and maximum values along with the sum of the given elements.

Numbers: 2, 5, 4, 1, 3

Output:

Minimum value: 1

Maximum value: 5

Sum of the elements: 15

```
public class MinMaxSum
{
    public static void main(String args[])
    {
        int arr[] = {2, 5, 4, 1, 3};

        int max = arr[0];
        int min = arr[0];
        int sum = 0;

        for (int i = 0; i < arr.length; i++) {
            if (arr[i] > max)
                max = arr[i];

            if (arr[i] < min)
                min = arr[i];

            sum += arr[i];
        }

        System.out.println("Minimum value: " + min);
        System.out.println("Maximum value: " + max);
        System.out.println("Sum of the elements: " + sum);
    }
}
```

Q. Differentiate between Sorting and Searching

Sorting	Searching
Sorting means to arrange the elements of the array in ascending or descending order.	Searching means to search for a term or value in an array.
Bubble sort and Selection sort are examples of sorting techniques.	Linear search and Binary search are examples of search techniques.

Q. Differentiate between Linear search and Binary search

Linear Search	Binary Search
Linear search works on sorted and unsorted arrays	Binary search works on only sorted arrays (ascending or descending)
Each element of the array is checked against the target value until the element is found or end of the array is reached	Array is successively divided into 2 halves and the target element is searched either in the first half or in the second half
Linear Search is slower	Binary Search is faster

Q. How does the linear search find an element in the array? Explain your answer with a suitable example.

Ans. In linear search, we start at the first element of the array and sequentially check each element of the list for the search value until a match is found or all the elements have been searched. As soon as the search value is found, the algorithm quits and returns the position (index) of the target value in the array.

For example, consider the following array:

```
int arr[] = {1, 8, 4, 7, 5};
```

We want to check if 7 is present in the array or not. Linear search will first check if 1 is equal to 7, then it will move on to the next element which is 8. It will keep doing this in a linear progression and when it reaches the element at index 3, it finds a match so it will give us this index 3 which means that 7 is present at index 3 of array arr. Q. Explain the technique of Bubble Sort with an example.

Ans. Bubble Sort is a sorting algorithm that works by repeatedly iterating through the array, comparing each pair of adjoining elements and swapping them if they are in wrong order.

For example, consider the following unsorted array:

9			
	5	2	3

Pass 1

First 9 is compared with 5 and as 9 is greater than 5 the elements are swapped:

5	9	2	3
---	---	---	---

Next, 9 is compared with 2 and as 9 is greater than 2 the elements are swapped:

5			
	2	9	3

Next, 9 is compared with 3 and as 9 is greater than 3 the elements are swapped:

5	2	3	9
---	---	---	---

At the end of first pass, the highest element of the array is at the last position.

Pass 2

5 is compared with 2 and as 5 is greater than 2 the elements are swapped:

2	5	3	9
---	---	---	---

Next, 5 is compared with 3 and as 5 is greater than 3 the elements are swapped:

2	3	5	9
---	---	---	---

At the end of first pass, the second highest element of the array is in its correct position.

Pass 3

2 is compared with 3 and as 2 is less than 3 no swapping takes place.

2	3	5	9
---	---	---	---

With this, the third and final pass ends and the elements of the array are in sorted order.

Q. Write a program to input integer elements into an array of size 20 and perform the following operations:

1. Display largest number from the array
  2. Display smallest number from the array
  3. Display sum of all the elements of the array
- Ans. `import java.util.Scanner;`

```
public class SDAMinMaxSum
{
    public static void main(String args[])
    {
        Scanner in = new
Scanner(System.in);          int arr[] = new
int[20];
        System.out.println("Enter 20 numbers:");
        for (int i = 0; i < 20; i++) {
            arr[i] = in.nextInt();
        }
    }
}
```

```

    }
    int min = arr[0], max = arr[0], sum = 0;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] < min)
            min = arr[i];

        if (arr[i] > max)
            max = arr[i];

        sum += arr[i];
    }

    System.out.println("Largest Number = " + max);
    System.out.println("Smallest Number = " + min);
    System.out.println("Sum = " + sum);
}
}

```

Q. Suppose x is an array of type int[] with 50 elements. Write a code segment that will count and print the frequency of number 42 in the array.

Ans.

```

int c = 0;
for (int i = 0; i < 50; i++) {
    if (x[i] == 42) {
        c++;
    }
}

System.out.println("Frequency of 42 = " + c);

```

Q. A student wrote the following code segment, intending to print 11 22 33 44:

```

int arr[] = {11, 22, 33, 44};
for (int i = 1; i <= 4; i++)
    System.out.println(arr[i]);

```

However, the program crashed with a run-time error. Can you explain the reason for this?

Ans.

Array index starts at 0 not 1. In the given program, the for loop runs from 1 to 4 whereas the indexes of the array range from 0 to 3. When i becomes 4, the program tries to access an index of arr that is not present in the array and this causes the program to crash. The correct way will be to run the for loop from 0 to 3 instead of 1 to 4.

Q. Write a code segment to compute the sum of all positive real numbers stored in the following array.

```
double numb[] = new double[50];

double sum = 0;
for (int i = 0; i < 50; i++) {
    if (numb[i] > 0) {
        sum
        += numb[i];
    }
}
System.out.println("Sum of positive real numbers = " + sum);
```

Q. Given the following declarations:

```
final int SIZE = 20; char[]
name = new char[SIZE];
```

i. Write an assignment statement that stores 'D' into the first element of the array name. ii.

Write an output statement that prints the value of the tenth element of the array name. iii.

Write a for statement that fills the array name with spaces.

Ans.

i. name[0] = 'D'; ii.

```
System.out.println(name[9]);
```

iii. For Statement:

```
for (int i = 0; i < SIZE; i++) {
    name[i] = ' ';
}
```

Q. What happens in Java if you try to access an element that is outside the bounds of the array?

Ans. Accessing an element that is outside the bounds of the array results in a runtime error in the form of `ArrayIndexOutOfBoundsException`.

Q. Write Java statements for the following:

i. Create an array to hold 15 double values.

ii. Assign the value 10.5 to the last element in the array. iii. Display the sum of the first and the last element.

iv. Write a loop that computes the sum of all elements in the array.

Ans.

i. Create an array to hold 15 double values.

```
double arr[] = new double[15];
```

ii. Assign the value 10.5 to the last element in the array.

```
arr[14] = 10.5;
```

iii. Display the sum of the first and the last element.

```
double r = arr[0] + arr[14];  
System.out.println("Result = " + r);
```

iv. Write a loop that computes the sum of all elements in the array.

```
double sum = 0;  
for (int i = 0; i < 15; i++) {  
    sum += arr[i];  
}  
System.out.println("Sum = " + sum);
```

Q. Write a program to accept the year of graduation from school as an integer value from the user. Using the binary search technique on the sorted array of integers given below, output the message "Record exists" if the value input is located in the array. If not, output the message "Record does not exist". Sample Input:

n[0]	n[1]	n[2]	n[3]	n[4]	n[5]	n[6]	n[7]	n[8]	n[9]
1982	1987	1993	1996	1999	2003	2006	2007	2009	2010

Ans.

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

```
public class GraduationYear
```

```
{    public static void main(String args[])
```

```
{        Scanner in = new Scanner(System.in);
```

```
        int n[] = {1982, 1987, 1993, 1996, 1999, 2003, 2006, 2007, 2009, 2010};
```

```
        System.out.print("Enter graduation year to search: ");
```

```
int year = in.nextInt();
```

```
        int l = 0, h = n.length - 1, idx = -1;
```

```
while (l <= h) {            int m = (l + h) /
```

```
2;                if (n[m] == year) {
```

```
idx = m;                break;                }
```

```
                else if (n[m] < year) {
```

```
l = m + 1;
```

```
                }
```

```
else {                h =
```

```
m - 1;
```

```
                }
```

```
        }
```

```
        if (idx == -1)
```

```
            System.out.println("Record does not exist");
```

```
else
```

```
            System.out.println("Record exists");
```

```
        }
```

```
}
```

Q. Write a program that reads ten integers and displays them in the reverse order in which they were read.

Ans. import

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

```
public class KboatSDAReverse
```



```

{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        int arr[] = new int[10];
        System.out.println("Enter 10 integers:");
        for (int i = 0; i < 10; i++) {
            arr[i] = in.nextInt();
        }

        System.out.println("Integers in reverse order:");
        for (int i = 9; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}

```

Q. Write a program that reads a long number, counts and displays the occurrences of each digit in it.

```

import java.util.Scanner;

public class SDANumber
{
    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number: ");
        long num = in.nextLong();          int
        dCount[] = new int[10];

        while (num != 0) {
            int d = (int)(num % 10);
            dCount[d] = dCount[d] + 1;
            num /= 10;
        }

        System.out.println("Digit\tOccurence");
        for (int i = 0; i < 10; i++) {
            if
            (dCount[i] != 0) {
                System.out.println(i + "\t" + dCount[i]);
            }
        }
    }
}

```

Q. Write a program to input 10 integer elements in an array and sort them in descending order using bubble sort technique.

Ans. `import`

`java.util.Scanner;`

```
public class BubbleSortDsc
{
    public static void main(String args[])
    {
        Scanner in = new
Scanner(System.in);        int n = 10;
        int arr[] = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] =
in.nextInt();
        }

        //Bubble Sort
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] < arr[j + 1]) {
                    int t = arr[j];
                    arr[j] =
arr[j+1];
                    arr[j+1] = t;
                }
            }
        }

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

Q. Write a program to perform binary search on a list of integers given below, to search for an element input by the user. If it is found display the element along with its position, otherwise display the message "Search element not found".

5, 7, 9, 11, 15, 20, 30, 45, 89, 97

```

Ans. import
java.util.Scanner;

public class BinarySearch
{
    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);
        int arr[] = {5, 7, 9, 11, 15, 20, 30, 45, 89, 97};

        System.out.print("Enter number to search: ");
        int n = in.nextInt();

        int l = 0, h = arr.length - 1, index = -1;
        while (l <= h) {
            int m = (l + h) / 2;
            if (arr[m] < n)
                l = m + 1;
            else if (arr[m] > n)
                h = m - 1;
            else {
                index = m;
                break;
            }
        }

        if (index == -1) {
            System.out.println("Search element not found");
        }
        else {
            System.out.println(n + " found at position " + index);
        }
    }
}

```

Q. Write a program to store 6 elements in an array P and 4 elements in an array Q. Now, produce a third array R, containing all the elements of array P and Q. Display the resultant array.

Input	Input	Output

P[ ]	Q[ ]	R[ ]
Input	Input	Output
4	19	4
6	23	6
1	7	1
2	8	2
3		3
10		10
		19
		23

		7
		8

Ans. import

java.util.Scanner;

public class Arrays

{

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

int P[] =

new int[6]; int Q[] =

new int[4]; int R[] =

new int[10]; int i =

0;

System.out.println("Enter 6 elements of array P:");

for (i = 0; i < P.length; i++) {

P[i] = in.nextInt();

}

System.out.println("Enter 4 elements of array Q:");

for (i = 0; i < Q.length; i++) {

Q[i] = in.nextInt();

}

i = 0;

while(i < P.length) {

R[i] = P[i]; i++;

}

int j = 0; while(j <

Q.length) {

R[i++] = Q[j++];

}

System.out.println("Elements of Array R:");

for (i = 0; i < R.length; i++) {

System.out.print(R[i] + " ");

```
} } }
```

Q. The annual examination result of 50 students in a class is tabulated in a Single Dimensional Array (SDA) is as follows:

Roll No.	Subject A	Subject B	Subject C
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

Write a program to read the data, calculate and display the following:

- Average marks obtained by each student.
- Print the roll number and the average marks of the students whose average is above 80.
- Print the roll number and the average marks of the students whose average is below 40.

Ans.

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

```
public class ExamResult
{
    public static void main(String args[])
    {
        final int TOTAL_STUDENTS = 50;
        Scanner in = new Scanner(System.in);

        int rollNo[] = new int[TOTAL_STUDENTS];
        int sA[] = new int[TOTAL_STUDENTS];
        int sB[] = new int[TOTAL_STUDENTS];
        int sC[] =
```

```

new int[TOTAL_STUDENTS];          double avg[] =
new double[TOTAL_STUDENTS];

    for (int i = 0; i < TOTAL_STUDENTS; i++) {
System.out.println("Enter student " + (i+1) + " details:");
        System.out.print("Roll No: ");
rollNo[i] = in.nextInt();
        System.out.print("Subject A Marks: ");
sA[i] = in.nextInt();
        System.out.print("Subject B Marks: ");
sB[i] = in.nextInt();
        System.out.print("Subject C Marks: ");
sC[i] = in.nextInt();
        avg[i] = (sA[i] + sB[i] + sC[i]) / 3.0;
    }

    System.out.println("\nRoll No\tAverage Marks");
for (int i = 0; i < TOTAL_STUDENTS; i++) {
System.out.println(rollNo[i] + "\t" + avg[i]);    }

    System.out.println("\nStudents with Average above 80:");
for (int i = 0; i < TOTAL_STUDENTS; i++) {
    if (avg[i] > 80)
        System.out.println(rollNo[i] + "\t" + avg[i]);
}

    System.out.println("\nStudents with Average below 40:");
for (int i = 0; i < TOTAL_STUDENTS; i++) {
    if (avg[i] < 40)
        System.out.println(rollNo[i] + "\t" + avg[i]);
}
} }

```

Q. Declare a single dimensional array of size 28 to store daily temperatures for the month of February. Using this structure, write a program to find:

1. The hottest day of the month
2. The coldest day of the month
3. The average temperature of the month Ans.

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

```

public class FebTemp
{
    public static void main(String args[])
    {
        Scanner in = new
Scanner(System.in);
        double febTemp[]
= new double[28];
        int n =
febTemp.length;

        System.out.println("Enter Feb daily temperatures:");
        for (int i = 0; i < n; i++) {
            febTemp[i] =
in.nextDouble();
        }
        double sum = 0.0;
        int low
= 0, high = 0;
        for (int i = 0; i <
n; i++) {
            if (febTemp[i] <
febTemp[low])
                low = i;

            if (febTemp[i] > febTemp[high])
                high = i;
            sum += febTemp[i];
        }
        double avg = sum / n;

        System.out.println("Hottest day = " + (high + 1));
        System.out.println("Coldest day = " + (low + 1));
        System.out.println("Average Temperature = " + avg);
    }
}

```

Q 27. Write a program that computes the standard deviation of N real numbers. The standard deviation s is computed according to:

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_N - \bar{x})^2}{N}}$$

The variable  $\bar{x}$  is the average of N input values  $x_1$  through  $x_N$ . The program first prompts the user for N and then declares an array of size N.

Ans.

```

import java.util.Scanner;

public class StdDev
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
    }
}

```



```

        System.out.print("Enter N:");
int n = in.nextInt();

        double a[] = new double[n];

        System.out.println("Enter the numbers:");
for (int i = 0; i < n; i++) {
    a[i] = in.nextDouble();
}
double sum
= 0;
for (int i = 0; i < n; i++) {
    sum += a[i];
}
double avg = sum / n;
double z = 0;
for (int i = 0; i < n; i++) {
    z += Math.pow(a[i]
- avg, 2);
}
double s = Math.sqrt(z / n);

        System.out.println("Standard Deviation(s) = " + s);
    }
}

```