

**SN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS2313 – Object Oriented Programming Lab**  
**II Year CSE - B Section ( III Semester)**  
**Academic Year 2022-23**  
**Batch: 2021- 2025**  
**Faculty Incharge :S.Rajalakshmi**

---

**Exercise 2. Java Programs- Arrays and Strings**

**Aim:**

To implement java programs using arrays and strings by getting input from the user.

1. Write a java program to get 'n' elements in an array. Perform the linear and binary search.

Program code:

```
import java.util.Scanner;
class Search
{
    public void binarysearch(int arr[],int key)    //pass the sorted array in
ascending order
    {
        int first=0;
        int last=arr.length-1;
        int mid=(first+last)/2;
        while(first<=last)
        {
            if(arr[mid]<key)    first=mid+1;
            else if(arr[mid]==key)
            {
                System.out.println(key+" found at index "+mid);
                break;
            }
        }
    }
}
```

```
        }
        else    last=mid-1;
        mid=(first+last)/2;
    }
    if(first>last) System.out.println(key+" not found...");
}

public int linearsearch(int arr[],int key)
{
    for(int i=0;i<arr.length;i++)
    {
        if(arr[i]==key)
            return i;
    }
    return -1;
}

}

class Main
{
    public static void main(String []args)
    {
        Scanner scanner=new Scanner(System.in);
        int temp,key,res;
        System.out.println("Enter the size of the array: ");
        int n=scanner.nextInt();
        int []arr=new int[n];    //declaring array in java
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the element: ");
            arr[i]=scanner.nextInt();
        }

        System.out.println("Enter the element to be searched: ");
        key=scanner.nextInt();

        //sorting the array using bubble sort
        for(int i=0;i<n;i++)
        {
            for(int j=1;j<n-i;j++)
            {
                if(arr[j-1]>arr[j])
                {
                    temp=arr[j-1];
                    arr[j-1]=arr[j];
                    arr[j]=temp;
                }
            }
        }
    }
}
```

```
    }  
}  
  
System.out.println("The array sorted in ascending order is: ");  
for(int i=0;i<n;i++) System.out.print(arr[i]+" ");  
System.out.println();  
Search search=new Search();  
search.binarysearch(arr,key);  
res=search.linearsearch(arr,key);  
if(res!=-1) System.out.println(key+" found at index "+res);  
else      System.out.println(key+" not found...");  
}  
}
```

Output:

```
Enter the size of the array:  
3  
Enter the element:  
10  
Enter the element:  
20  
Enter the element:  
30  
Enter the element to be searched:  
20  
The array sorted in ascending order is:  
10 20 30  
20 found at index 1  
20 found at index 1
```

## 2. Write a java program to find matrix addition, subtraction and multiplication.

Program code:

```
import java.util.Scanner;
class Matrix
{
    void create_matrix()
    {
        Scanner scanner=new Scanner(System.in);

        System.out.println("Enter the no. of rows of the matrix 1: ");
        int rows1=scanner.nextInt();
        System.out.println("Enter the no. of columns of the matrix 1: ");
        int columns1=scanner.nextInt();
        int [][]mat1=new int[rows1][columns1];
        System.out.println("Enter the elements for matrix 1: ");
        for(int i=0;i<rows1;i++)
        {
            for(int j=0;j<columns1;j++)
            {
                System.out.println("Enter element["+(i+1)+(j+1)+"]: ");
                mat1[i][j]=scanner.nextInt();
            }
        }

        System.out.println("Enter the no. of rows of the matrix 2: ");
        int rows2=scanner.nextInt();
        System.out.println("Enter the no. of columns of the matrix 2: ");
        int columns2=scanner.nextInt();
        int [][]mat2=new int[rows2][columns2];
        System.out.println("Enter the elements for matrix 2: ");
        for(int i=0;i<rows2;i++)
        {
            for(int j=0;j<columns2;j++)
            {
                System.out.println("Enter element["+(i+1)+(j+1)+"]: ");
                mat2[i][j]=scanner.nextInt();
            }
        }

        System.out.println("Matrix 1: ");
        for(int i=0;i<rows1;i++)
        {
            for(int j=0;j<columns1;j++)
                System.out.print(mat1[i][j]+" ");
            System.out.println();
        }
    }
}
```

```
}

System.out.println("Matrix 2: ");
for(int i=0;i<rows2;i++)
{
    for(int j=0;j<columns2;j++)
        System.out.print(mat2[i][j]+" ");
    System.out.println();
}

System.out.println("Matrix addition: ");
if(rows1==rows2 && columns1==columns2)
{
    int [][]add=new int [rows1][columns1];
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns1;j++)
            add[i][j]=mat1[i][j]+mat2[i][j];
    }
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns1;j++)
            System.out.print(add[i][j]+" ");
        System.out.println();
    }
}
else
    System.out.println("Cannot perform matrix addition...");

System.out.println("Matrix subtraction: ");
if(rows1==rows2 && columns1==columns2)
{
    int [][]sub=new int [rows1][columns1];
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns1;j++)
            sub[i][j]=mat1[i][j]-mat2[i][j];
    }
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns1;j++)
            System.out.print(sub[i][j]+" ");
        System.out.println();
    }
}
else
    System.out.println("Cannot perform matrix subtraction...");
```

```
System.out.println("Matrix multiplication: ");
if(rows2==columns1)
{
    int [][]mul=new int [rows1][columns2];
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns2;j++)
        {
            mul[i][j]=0;
            for(int k=0;k<rows2;k++)
                mul[i][j]+=mat1[i][k]*mat2[k][j];
        }
    }
    for(int i=0;i<rows1;i++)
    {
        for(int j=0;j<columns2;j++)
            System.out.print(mul[i][j]+" ");
        System.out.println();
    }
}
else
    System.out.println("Cannot perform matrix multiplication...");
}

class Main
{
    public static void main(String[] args)
    {
        Matrix matrix=new Matrix();
        matrix.create_matrix();
    }
}
```

Ex no.:2  
Date: 12-10-2022

Name: M.Rohith  
Reg. no : 3122 21 5001 085

Output:

```
Enter the no. of rows of the matrix 1:
3
Enter the no. of columns of the matrix 1:
4
Enter the elements for matrix 1:
Enter element[11]:
1
Enter element[12]:
1
Enter element[13]:
1
Enter element[14]:
1
Enter element[21]:
2
Enter element[22]:
2
Enter element[23]:
2
Enter element[24]:
2
Enter element[31]:
3
Enter element[32]:
3
Enter element[33]:
3
Enter element[34]:
3
```

```
Enter the no. of rows of the matrix 2:
4
Enter the no. of columns of the matrix 2:
3
Enter the elements for matrix 2:
Enter element[11]:
1
Enter element[12]:
1
Enter element[13]:
1
Enter element[21]:
1
Enter element[22]:
2
Enter element[23]:
2
Enter element[31]:
3
Enter element[32]:
3
Enter element[33]:
3
Enter element[41]:
4
Enter element[42]:
4
Enter element[43]:
4
```

Ex no.:2  
Date: 12-10-2022

Name: M.Rohith  
Reg. no : 3122 21 5001 085

```
Matrix 1:
1 1 1 1
2 2 2 2
3 3 3 3
Matrix 2:
1 1 1
1 2 2
3 3 3
4 4 4
Matrix addition:
Cannot perform matrix addition...
Matrix subtraction:
Cannot perform matrix subtraction...
Matrix multiplication:
9 10 10
18 20 20
27 30 30
```



**3. Write a Java program to get a sentence and find the longest word in it. Also find it's index position.**

Program code:

```
import java.util.Scanner;

class SmallestLargestWord
{
    void large_small(String string)
    {
        String word = "", small = "", large="";
        String[] words = new String[100];
        int length = 0;

        string = string + " ";

        for(int i = 0; i < string.length(); i++)
        {
            if(string.charAt(i) != ' ')
            {
                word = word + string.charAt(i);
            }
            else
            {
                words[length] = word;
                length++;
                word = "";
            }
        }

        small = large = words[0];

        for(int k = 0; k < length; k++)
        {
            if(small.length() > words[k].length())
                small = words[k];
            if(large.length() < words[k].length())
                large = words[k];
        }

        int result1=0,result2=0,j=0;
        System.out.println("Smallest word: " + small);
    }
}
```

```
for(int i=0;i<string.length();i++)
{
    if(j<small.length() && string.charAt(i)==small.charAt(j))
    {
        result1=1;
        j++;
        if(j==small.length())
        {
            System.out.println(small+" found at index "+(i-j+1));
            break;
        }
    }
    else
    {
        result1=0;
        j=0;
    }
}

j=0;
System.out.println("Largest word: " + large);
for(int i=0;i<string.length();i++)
{
    if(j<large.length() && string.charAt(i)==large.charAt(j))
    {
        result2=1;
        j++;
        if(j==large.length())
        {
            System.out.println(large+" found at index "+(i-j+1));
            break;
        }
    }
    else
    {
        result2=0;
        j=0;
    }
}
}

class Main
{
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the string: ");
    }
}
```

Ex no.:2  
Date: 12-10-2022

Name: M.Rohith  
Reg. no : 3122 21 5001 085

```
String string = scanner.nextLine();  
SmallestLargestWord s=new SmallestLargestWord();  
s.large_small(string);  
}  
}
```

Output:

```
Enter the string:  
Well known programming languages are C, C++, Java, Python.  
Smallest word: C,  
C, found at index 37  
Largest word: programming  
programming found at index 11
```

**4. Write a Java program to get a string and verify whether it's a Palindrome or not.**

Program code:

```
import java.util.Scanner;

class Palindrome
{
    public static boolean isPalindrome(String str)
    {
        String rev = "";
        boolean ans = false;

        for (int i = str.length()-1; i>=0; i--) {
            rev = rev + str.charAt(i);
        }

        if (str.equals(rev))
        {
            ans = true;
        }
        return ans;
    }
}

class Main
{
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the string to check palindrome: ");
        String str =scanner.nextLine();
        Palindrome palindrome=new Palindrome();

        str = str.toLowerCase();
        boolean res = palindrome.isPalindrome(str);
        System.out.println(res);
    }
}
```

Output:

```
Enter the string to check palindrome:
Malayalam
true
```

5. Write a Java program to check if the given sentence is a Pangram. Pangram string is the string in which all the alphabets should occur atleast once. Also count the number of occurrences of each letter.

Example:

The quick brown fox jumps over the lazy dog

Program code:

```
import java.util.Scanner;

class Example
{
    void pangram(String str)
    {
        boolean[] alphaList = new boolean[26];
        int[] countlist=new int[26];
        int index = 0;
        int flag = 1;
        for(int i=0;i<26;i++)
        {
            countlist[i]=0;
        }
        for (int i = 0; i < str.length(); i++)
        {
            if ( str.charAt(i) >= 'A' && str.charAt(i) <= 'Z')
            {
                index = str.charAt(i) - 'A';
            }
            else if( str.charAt(i) >= 'a' && str.charAt(i) <= 'z')
            {
                index = str.charAt(i) - 'a';
            }
            alphaList[index] = true;
            countlist[index]+=1;
        }
        for (int i = 0; i <= 25; i++)
        {
            if (alphaList[i] == false)
                flag = 0;
        }

        System.out.println("String: " + str);
        if (flag == 1)
        {
            int i=0;
            System.out.println("The above string is a pangram.");
            for(char c='A';c<='Z';c++)
            {
                System.out.print(countlist[i] + " ");
                i++;
            }
        }
    }
}
```

```
        System.out.println(c+": "+countlist[i]);
        i++;
    }
}
else
{
    int i=0;
    System.out.println("The above string is not a pangram.");
    for(char c='A';c<='Z';c++)
    {
        System.out.println(c+": "+countlist[i]);
        i++;
    }
}
}

class Main
{
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the string to check pangram: ");
        String str = scanner.nextLine();
        Example e=new Example();
        e.pangram(str);
    }
}
```

Ex no.:2  
Date: 12-10-2022

Name: M.Rohith  
Reg. no : 3122 21 5001 085

Output:

```
Enter the string to check pangram:  
The quick brown fox jumps over the lazy dog  
String: The quick brown fox jumps over the lazy dog  
The above string is a pangram.  
A: 1  
B: 1  
C: 1  
D: 1  
E: 5  
F: 1  
G: 1  
H: 2  
I: 1  
J: 1  
K: 2  
L: 1  
M: 1  
N: 2  
O: 4  
P: 1  
Q: 1  
R: 3  
S: 2  
T: 2  
U: 2  
V: 1  
W: 1  
X: 2  
Y: 2  
Z: 1
```

Learning Outcome:

Thus java programs using arrays and strings has been implemented and executed successfully by getting the input from the user.