

Exercise 2 – Arrays and Strings

Q1: Write a java program to get 'n' elements in an array. Perform the linear and binary search.

Code:

```
import java.util.Scanner;

class Search {
    //function to display the array
    public static void display (int a[]) {
        for (int i=0; i<a.length; i++)
            System.out.print(a[i]+" ");
        System.out.println();
    }

    //function to sort an array in ascending order
    public static void sort (int a[]) {
        int small, pos;
        for (int i=0; i<a.length; i++) {
            small=a[i];
            pos=i;
            for (int j=i+1; j<a.length; j++) {
                if (a[j]<small) {
                    small=a[j];
                    pos=j;
                }
            }
            a[pos]=a[i];
            a[i]=small;
        }

        //to print the sorted array
        System.out.print("Sorted array: ");
        display(a);
    }

    //function to perform linear search on an array
    public static void lsearch (int a[], int num) {
        int flag=0;
        for (int i=0; i<a.length; i++) {
            if (a[i]==num) {
                System.out.println("Number found at position "+(i+1)+" of the sorted
array.");
                flag=1;
            }
        }
        if (flag==0)
```

```
        System.out.println("Number not found in array!");
    }

    //function to perform binary search on an array
    public static void bsearch (int a[], int num) {
        int flag=0, mid, beg=0, last=a.length-1;
        while (beg<last && flag==0) {
            mid=(beg+last)/2;
            if (a[mid]==num) {
                flag=1;
                System.out.println("Number found at position "+(mid+1)+ " of the
sorted array.");
                break;
            }
            else if (a[mid]>num)
                last=mid;
            else
                beg=mid+1;
        }
        if (flag==0)
            System.out.println("Number not found in array!");
    }

    public static void main (String a[]) {
        Scanner sc = new Scanner(System.in);
        //to get the size of the array
        System.out.print("Enter size of array: ");
        int n = sc.nextInt();

        //to get the array input
        System.out.println("Enter array elements: ");
        int arr[] = new int[n];
        for (int i=0; i<n; i++) {
            arr[i] = sc.nextInt();
        }

        //to sort the array
        sort(arr);

        //to create a menu for performing linear and binary search
        System.out.print("Do you wish to perform a search? (y/n): ");
        String ch = sc.next();
        while (ch.equalsIgnoreCase("y")) {
            System.out.print("Enter number to search for: ");
            int reqd = sc.nextInt();
            System.out.print("1: linear search, 2: binary search; Enter choice: ");
            int choice = sc.nextInt();
            switch (choice) {
                case 1: lsearch(arr,reqd); break;
```

```
        case 2: bsearch(arr, reqd); break;
        default: System.out.println("Invalid choice!");
    }
    System.out.print("Do you wish to perform another search? (y/n): ");
    ch = sc.next();
}
}
```

Output:

```
kri@kri-ubuntu:~/workspace$ javac Search.java
kri@kri-ubuntu:~/workspace$ java Search
Enter size of array: 5
Enter array elements:
57
32
68
91
24
Sorted array: 24 32 57 68 91
Do you wish to perform a search? (y/n): y
Enter number to search for: 68
1: linear search, 2: binary search; Enter choice: 1
Number found at position 4 of the sorted array.
Do you wish to perform another search? (y/n): y
Enter number to search for: 32
1: linear search, 2: binary search; Enter choice: 2
Number found at position 2 of the sorted array.
Do you wish to perform another search? (y/n): y
Enter number to search for: 38
1: linear search, 2: binary search; Enter choice: 2
Number not found in array!
Do you wish to perform another search? (y/n): y
Enter number to search for: 102
1: linear search, 2: binary search; Enter choice: 1
Number not found in array!
Do you wish to perform another search? (y/n): y
Enter number to search for: 10
1: linear search, 2: binary search; Enter choice: 2
Number not found in array!
Do you wish to perform another search? (y/n): y
Enter number to search for: 91
1: linear search, 2: binary search; Enter choice: 1
Number found at position 5 of the sorted array.
Do you wish to perform another search? (y/n): y
Enter number to search for: 24
1: linear search, 2: binary search; Enter choice: 2
Number found at position 1 of the sorted array.
Do you wish to perform another search? (y/n): n
```

Q2: Write a java program to perform matrix addition, subtraction and multiplication.

Code:

```
import java.util.Scanner;

class Matrix {
    public static void display (int m[][]) {
        for (int i=0; i<m.length; i++) {
            for (int j=0; j<m[i].length; j++) {
                System.out.print(m[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println();
    }

    public static void add (int a[][], int b[][]) {
        System.out.println("Addition of second matrix to first matrix: ");
        for (int i=0; i<a.length; i++) {
            for (int j=0; j<a[i].length; j++) {
                System.out.print(a[i][j]+b[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println();
    }

    public static void subtract (int a[][], int b[][]) {
        System.out.println("Subtraction of second matrix from first matrix: ");
        for (int i=0; i<a.length; i++) {
            for (int j=0; j<a[i].length; j++) {
                System.out.print(a[i][j]-b[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println();
    }

    public static void multiply (int a[][], int b[][]) {
        System.out.println("Multiplication of the two matrices: ");
        int [][] c = new int[a.length][a[0].length];
        for (int i=0; i<a.length; i++) {
            for (int j=0; j<a[i].length; j++) {
                for (int k=0; k<a.length; k++) {
                    c[i][j]+=a[i][k]*b[k][j];
                }
            }
        }
    }
}
```

```
        System.out.print(c[i][j]+" ");
    }
    System.out.println();
}
System.out.println();
}
```

```
public static void main (String a[]) {
    Scanner sc = new Scanner(System.in);
    //to get no. of rows and columns as input from the user
    System.out.print("Enter no. of rows: ");
    int r = sc.nextInt();
    System.out.print("Enter no. of columns: ");
    int c = sc.nextInt();
```

```
    //to get the matrix input
    //matrix A
    int [][] A = new int[r][c];
    System.out.println("Enter matrix A elements: ");
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            A[i][j] = sc.nextInt();
        }
    }
```

```
    //matrix B
    int [][] B = new int[r][c];
    System.out.println("Enter matrix B elements: ");
    for (int i=0; i<r; i++) {
        for (int j=0; j<c; j++) {
            B[i][j] = sc.nextInt();
        }
    }
```

```
    //displaying matrices A and B
    System.out.println("\nDisplaying A: ");
    display(A);
    System.out.println("Displaying B: ");
    display(B);
```

```
    //addition, subtraction, multiplication - menu
    do {
```

```
        System.out.print("1: addition, 2: subtraction, 3: multiplication, 4: quit\nEnter
choice: ");
```

```
        int choice = sc.nextInt();
        switch (choice) {
            case 1: add(A,B); break;
            case 2: subtract(A,B); break;
            case 3: multiply(A,B); break;
```

```
        case 4: System.exit(0);  
        default: System.out.println("Invalid choice!");  
        }  
    } while (true);  
}  
}
```

Output:

```
kri@kri-ubuntu:~/workspace$ javac Matrix.java  
kri@kri-ubuntu:~/workspace$ java Matrix  
Enter no. of rows: 2  
Enter no. of columns: 2  
Enter matrix A elements:  
2  
1  
0  
3  
Enter matrix B elements:  
4  
2  
1  
5  
  
Displaying A:  
2 1  
0 3  
  
Displaying B:  
4 2  
1 5  
  
1: addition, 2: subtraction, 3: multiplication, 4: quit  
Enter choice: 1  
Addition of second matrix to first matrix:  
6 3  
1 8  
  
1: addition, 2: subtraction, 3: multiplication, 4: quit  
Enter choice: 2  
Subtraction of second matrix from first matrix:  
-2 -1  
-1 -2  
  
1: addition, 2: subtraction, 3: multiplication, 4: quit  
Enter choice: 3  
Multiplication of the two matrices:  
9 9  
3 15  
  
1: addition, 2: subtraction, 3: multiplication, 4: quit  
Enter choice: 4
```

Q3: Write a Java program to get a sentence and find the longest word in it. Also find its index position.

Code:

```
import java.util.Scanner;

class FindLongestWord {
    public static void longest (String s) {
        int max=0, count=0, index=0;
        String word="", long_word="";
        for (int i=0; i<s.length(); i++) {
            String c=s.substring(i,i+1);
            if (!c.equals(" ") && !c.equals("\n") && !c.equals(".") && !c.equals("?")) {
                count++;
                word=word.concat(c);
                if (i==s.length()-1) {
                    if (count>max) {
                        max=count;
                        long_word=word;
                    }
                    count=0;
                    word="";
                }
            }
            else {
                if (count>max) {
                    max=count;
                    long_word=word;
                    index=i-count;
                }
                count=0;
                word="";
            }
        }
        System.out.println("Longest word in sentence: "+long_word);
        System.out.println("Index position: "+index+"\tWord length: "+max);
    }

    public static void main (String a[]) {
        Scanner sc = new Scanner(System.in);
        //to get a sentence as input from the user
        System.out.print("Enter a sentence: ");
        String sent = sc.nextLine();
        //call function to find longest word
        longest(sent);
    }
}
```

Output:

```
kri@kri-ubuntu:~/workspace$ javac FindLongestWord.java
kri@kri-ubuntu:~/workspace$ java FindLongestWord
Enter a sentence: This is my first java program on strings.
Longest word in sentence: program
Index position: 22      Word length: 7
kri@kri-ubuntu:~/workspace$ java FindLongestWord
Enter a sentence: This is my first java program involving string manipulation.
Longest word in sentence: manipulation
Index position: 47      Word length: 12
kri@kri-ubuntu:~/workspace$ java FindLongestWord
Enter a sentence: A sentence must always end with a
Longest word in sentence: sentence
Index position: 2       Word length: 8
```


Q4: Write a Java program to get a string and verify whether it's a palindrome or not.

Code:

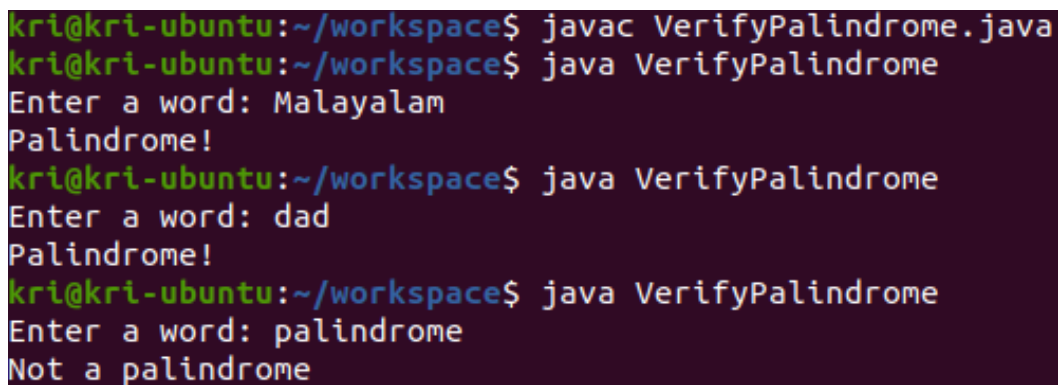
```
import java.util.Scanner;

class VerifyPalindrome {
    public static void palCheck (String s) {
        int l=s.length(), flag=0;
        for (int i=0; i<l; i++) {
            if (!s.substring(i,i+1).equalsIgnoreCase(s.substring(l-1-i,l-i))) {
                System.out.println("Not a palindrome");
                flag=1;
                break;
            }
        }
        if (flag==0)
            System.out.println("Palindrome!");
    }

    public static void main (String a[]) {
        Scanner sc = new Scanner(System.in);
        //to get a word as input from the user
        System.out.print("Enter a word: ");
        String word = sc.next();

        //to call the function to check if word is a palindrome
        palCheck(word);
    }
}
```

Output:



```
kri@kri-ubuntu:~/workspace$ javac VerifyPalindrome.java
kri@kri-ubuntu:~/workspace$ java VerifyPalindrome
Enter a word: Malayalam
Palindrome!
kri@kri-ubuntu:~/workspace$ java VerifyPalindrome
Enter a word: dad
Palindrome!
kri@kri-ubuntu:~/workspace$ java VerifyPalindrome
Enter a word: palindrome
Not a palindrome
```

Q5: Write a Java program to check if the given sentence is a pangram. Also count the number of occurrences of each letter.

Code:

```
import java.util.Scanner;

class Pangram {
    public static boolean checkPangram (String s) {
        //creating an array where each index represents a letter of the alphabet
        int count[] = new int[26]; //default count is 0 for each letter
        //for traversing the array and string to check if each letter is present
        int index=0;
        for (int i=0; i<s.length(); i++) {
            //if letter is in uppercase, subtract 65
            if ('A'<=s.charAt(i) && s.charAt(i)<='Z') {
                index = s.charAt(i)-'A';
            }
            //if letter is in lowercase, subtract 96
            else if ('a'<=s.charAt(i) && s.charAt(i)<='z') {
                index = s.charAt(i)-'a';
            }
            //if character is not a letter, go to next character
            else
                continue;
            //increment the letter's counter if the letter is encountered
            count[index]++;
        }

        //to print the number of occurrences of each letter
        for (index=0; index<26; index++) {
            System.out.print((char)('a'+index)+": "+count[index]+' ');
            //for formatting the output
            if ((index+1)%9==0)
                System.out.println();
        }

        //check if occurrence is at least 1 for all letters
        //returns false if any letter isn't present, else true
        for (index=0; index<26; index++) {
            if (count[index]==0)
                return false;
        }
        return true;
    }

    public static void main (String a[]) {
        Scanner sc = new Scanner(System.in);
```

```
//to get a sentence as input from the user
System.out.print("Enter a sentence: ");
String sent = sc.nextLine();

//to call the function to check for pangram
if (checkPangram(sent)==true)
    System.out.println("\nThe sentence is a pangram.");
else
    System.out.println("\nThe sentence is not a pangram.");
}
```

Output:

```
kri@kri-ubuntu:~/workspace$ javac Pangram.java
kri@kri-ubuntu:~/workspace$ java Pangram
Enter a sentence: The quick brown fox jumps over the lazy dog.
a: 1   b: 1   c: 1   d: 1   e: 3   f: 1   g: 1   h: 2   i: 1
j: 1   k: 1   l: 1   m: 1   n: 1   o: 4   p: 1   q: 1   r: 2
s: 1   t: 2   u: 2   v: 1   w: 1   x: 1   y: 1   z: 1
The sentence is a pangram.
kri@kri-ubuntu:~/workspace$ java Pangram
Enter a sentence: The quick brown fox jumps over the dog.
a: 0   b: 1   c: 1   d: 1   e: 3   f: 1   g: 1   h: 2   i: 1
j: 1   k: 1   l: 0   m: 1   n: 1   o: 4   p: 1   q: 1   r: 2
s: 1   t: 2   u: 2   v: 1   w: 1   x: 1   y: 0   z: 0
The sentence is not a pangram.
kri@kri-ubuntu:~/workspace$ java Pangram
Enter a sentence: The quick brown fox jumped over the lazy dog.
a: 1   b: 1   c: 1   d: 2   e: 4   f: 1   g: 1   h: 2   i: 1
j: 1   k: 1   l: 1   m: 1   n: 1   o: 4   p: 1   q: 1   r: 2
s: 0   t: 2   u: 2   v: 1   w: 1   x: 1   y: 1   z: 1
The sentence is not a pangram.
```