ASSIGNMENT-LAB 04

Course Code: **CSE-2340** **Course Title:** **Software Development 1**
Course Teacher: **Md. Mahadi Hassan**

Name: Ezaz Ahmed ID: C223009 Section: 3AM

Problem 01: **Write a program to convert an integer from decimal to binary.**

Answer:

```
import java.util.Scanner;
public class DTB
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter an integer in decimal: ");
        int decimal = s.nextInt();
        String binary = Integer.toBinaryString(decimal);
        System.out.println(" Binary Form Of " + decimal + " is "
+ binary);
        s.close();
    }
}
```

Problem 02: **Write a program to sort n numbers taken from keyboard.**

Answer:

```
import java.util.Scanner;
import java.util.Arrays;
public class Sort
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");
        int n = s.nextInt();
        int[] num = new int[n];
        System.out.print("Enter the elements: ");
        for (int i = 0; i < n; i++)
        {
            num[i] = s.nextInt();
        }
        Arrays.sort(num);
        System.out.println("Sorted elements:");
        for (int k : num)
        {
            System.out.print(k + " ");
        }
        s.close();
    }
}
```

Problem 03: Write a program to calculate row sum and column sum.

Answer:

```
import java.util.Scanner;
public class RCS
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int r = s.nextInt();
        System.out.print("Enter the number of columns: ");
        int c = s.nextInt();
        int[][] mat = new int[r][c];
        System.out.println("Enter the elements of the matrix:");
        for(int i = 0; i < r; i++)
        {
            for(int j = 0; j < c; j++)
            {
                mat[i][j] = s.nextInt();
            }
        }
        int[] rS = new int[r];
        int[] cS = new int[c];
        for (int i = 0; i < r; i++)
        {
            for (int j = 0; j < c; j++)
            {
                rS[i] += mat[i][j];
            }
        }
        for (int j = 0; j < c; j++)
        {
            for (int i = 0; i < r; i++)
            {
                cS[j] += mat[i][j];
            }
        }

        System.out.println("Row sums:");
        for(int i = 0; i < r; i++)
        {
            System.out.println("Row " + (i + 1) + ": " + rS[i]);
        }
        System.out.println("Column sums:");
        for(int j = 0; j < c; j++)
        {
            System.out.println("Column " + (j + 1) + ": " +
cS[j]);
        }
        s.close();
    }
}
```

Problem 04: Write a program that reads a positive integer n and then prints a diamond of asterisks in 2n-1 rows. For example, if n is 4, then the output would be

```
      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

Answer:

```
import java.util.Scanner;

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

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

        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= n - i; j++)
            {
                System.out.print("  ");
            }
            for (int k = 1; k <= 2 * i - 1; k++)
            {
                System.out.print("* ");
            }
            System.out.println();
        }
        for (int i = n - 1; i >= 1; i--)
        {
            for (int j = 1; j <= n - i; j++)
            {
                System.out.print("  ");
            }
            for (int k = 1; k <= 2 * i - 1; k++)
            {
                System.out.print("* ");
            }
            System.out.println();
        }

        s.close();
    }
}
```

Problem 05: **Write a program that reads a positive integer n and then prints a pyramid of numbers in 2n-1 rows. For example, if n is 4, then the output would be**

```
    1
  1 2 1
1 2 3 2 1
```

Answer:

```
import java.util.Scanner;
public class NP
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter a positive integer n: ");
        int n = s.nextInt();
        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= n - i; j++)
            {
                System.out.print("  ");
            }
            for (int k = 1; k <= i; k++)
            {
                System.out.print(k + " ");
            }
            for (int k = i - 1; k >= 1; k--)
            {
                System.out.print(k + " ");
            }
            System.out.println();
        }
        s.close();
    }
}
```

Problem 06: Linear search.

Answer:

```
import java.util.Scanner;
public class LS
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of elements in the 
list: ");
        int n = s.nextInt();
        int[] arr = new int[n];

        System.out.print("Enter elements: ");
        for (int i = 0; i < n; i++)
        {
            arr[i] = s.nextInt();
        }
        System.out.print("Enter the element to search: ");
        int k = s.nextInt();
        int index = -1;

        for (int i = 0; i < n; i++)
        {
            if (arr[i] == k)
            {
                index = i;
                break;
            }
        }
        if (index != -1)
        {
            System.out.println("Element found at index: " + 
index);
        }
        else
        {
            System.out.println("Element not found in the 
list.");
        }
        s.close();
    }
}
```

Problem 07: Binary search

Answer:

```
import java.util.Scanner;
public class BS
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter the number of elements in the
sorted list: ");
        int n = s.nextInt();
        int[] arr = new int[n];
        System.out.print("Enter elements: ");
        for (int i = 0; i < n; i++)
        {
            arr[i] = s.nextInt();
        }
        System.out.print("Enter the element to search for: ");
        int k = s.nextInt();
        int low = 0;
        int high = n - 1;
        int index = -1;
        while (low <= high)
        {
            int mid = (high + low) / 2;
            if (arr[mid] == k)
            {
                index = mid;
                break;
            }
            else if (arr[mid] < k)
            {
                low = mid + 1;
            }
            else
            {
                high = mid - 1;
            }
        }
        if (index != -1)
        {
            System.out.println("Element found at index: " +
index);
        }
        else
        {
            System.out.println("Element not found in the
list.");
        }

        s.close();
    }
}
```