Search & Sorting and Patterns and Other Common Codes

```
public class binarySearch {

   public static void main(String[] args) {

      int[] arr = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50};
      int key = 35;

      int result = BinarySearch(arr, key);

      if (result == -1) {
         System.out.println("Element not found in array");
      } else {
         System.out.println("Element found at index: " + result);
      }
   }

   public static int BinarySearch(int[] arr, int key) {

      int low = 0;
      int high = arr.length - 1;

      while (low <= high) {
         int mid = (low + high) / 2;

         if (arr[mid] == key) {
            return mid;
         } else if (arr[mid] < key) {
            low = mid + 1;
         } else {
            high = mid - 1;
         }
      }
      return -1;
   }
}
```

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

    public static void main(String[] args) {
        int[] arr = { 5, 3, 1, 4, 2 };
        int key = 4;
        int index = LinearSearch(arr, key);
        if(index == -1) {
            System.out.println("Element not found!");
        } else {
            System.out.println("Element found at index " + index);
        }
    }
}
```

```
public class bubbleSort {
    public static void main(String[] args) {
        int[] arr = {5, 2, 9, 1, 5, 6};

        System.out.println("Array before sorting: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }

        BubbleSort(arr);

        System.out.println("\nArray after sorting: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }

    public static void BubbleSort(int[] arr) {
        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // swap arr[j] and arr[j+1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}
```

```
public class SelectionSort {

    public static void main(String[] args) {
        int[] arr = {64, 25, 12, 22, 11};
        selectionSort(arr);
        System.out.println("Sorted array:");
        printArray(arr);
    }

    public static void selectionSort(int[] arr) {
        int n = arr.length;
        // One by one move boundary of unsorted subarray
        for (int i = 0; i < n-1; i++) {
            // Find the minimum element in unsorted array
            int minIndex = i;
            for (int j = i+1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }
            // Swap the found minimum element with the first element
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
    }

    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```

```
public class InsertionSort {

    public static void main(String[] args) {

        int[] arr = {5, 2, 4, 6, 1, 3};

        System.out.println("Before sorting:");
        printArray(arr);

        insertionSort(arr);

        System.out.println("After sorting:");
        printArray(arr);

    }

    public static void insertionSort(int[] arr) {
        int n = arr.length;
        for (int i = 1; i < n; ++i) {
            int key = arr[i];
            int j = i - 1;

            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }

    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; ++i) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```

```
import java.util.Scanner;

public class LPattern {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int n = input.nextInt();

        for (int i = 0; i < n; i++) { // for each row
            for (int j = 0; j < n; j++) { // for each element in row
                if (i == n - 1 || j == 0) { // if last row or first element in row
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
        input.close();
    }
}
```

```
import java.util.Scanner;

public class rightAngle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int n = scanner.nextInt();
        scanner.close();

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

```
import java.util.Scanner;

public class reverseRightAngle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        scanner.close();

        for(int i = 0; i < rows; i++) {
            for(int j = 0; j <= rows - i -1; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

```
public class inverseRightAngle {
    public static void main(String[] args) {
        System.out.print("Enter the number of rows: ");
        int rows = Integer.parseInt(System.console().readLine());

        for (int i = 0; i < rows; i++){
            for (int j = 0; j < i; j++){
                System.out.print(" ");
            }
            for (int k = 0; k < rows - i; k++){
                System.out.print("*");
            }
            System.out.println("");
        }
    }
}
```

```
public class inverseReverseRightAngle {
    public static void main(String[] args) {
        System.out.print("Enter the number of rows: ");
        int rows = Integer.parseInt(System.console().readLine());

        for (int i = 0; i < rows; i++){
            for (int j = 0; j < rows - i; j++){
                System.out.print(" ");
            }
            for (int k = 0; k < i + 1; k++){
                System.out.print("*");
            }
            System.out.println("");
        }
    }
}
```

```java
import java.util.Scanner;

public class DiamondPattern {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of rows (odd number): ");
        int rows = sc.nextInt();

        // upper half of the diamond
        for (int i = 1; i <= rows/2 + 1; i++) {
            for (int j = 1; j <= rows/2 + 1 - i; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("*");
            }
            System.out.println();
        }

        // lower half of the diamond
        for (int i = rows/2; i >= 1; i--) {
            for (int j = 1; j <= rows/2 + 1 - i; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```java
import java.util.Scanner;

public class pascalTriangle {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = in.nextInt();

        int[][] pascal = new int[rows][rows];
        pascal[0][0] = 1;

        for (int i = 0; i < rows; i++) { // for each row
            for (int j = 0; j < rows - i; j++) { // print spaces
                System.out.print(" ");
            }
            for (int j = 0; j <= i; j++) { // for each element in row
                if (j == 0 || j == i) { // if first or last element of row
                    pascal[i][j] = 1; // set to 1
                } else {
                    pascal[i][j] = pascal[i - 1][j - 1] + pascal[i - 1][j]; // else add the two elements above it
                }
                System.out.printf("%d ", pascal[i][j]);
            }
            System.out.println();
        }
        in.close(); // close scanner
    }
}
```

```java
import java.lang.Math;

public class PrimeNumber {
    public static void main(String[] args) {
        System.out.print("Enter a number: ");
        int n = Integer.parseInt(System.console().readLine());
        if (isPrime(n)) {
            System.out.println(n + " is a prime number.");
        } else {
            System.out.println(n + " is not a prime number.");
        }
    }

    public static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i < Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

```
public class Factorial {
    public static void main(String[] args) {
        System.out.print("Enter the number: ");
        int n = Integer.parseInt(System.console().readLine());
        System.out.println("The factorial of " + n + " is " + factorial(n));
    }

    public static int factorial(int n) {
        if (n == 0) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }
}
```

```
public class Reverse {
    public static void main(String[] args) {
        System.out.print("Enter the number: ");
        String str = System.console().readLine();
        reverse(str);
    }

    public static void reverse(String str) {
        int i = str.length() - 1;
        System.out.print("Reverse: ");
        while (i >= 0) {
            System.out.print(str.charAt(i));
            i--;
        }
        System.out.println("");
    }
}
```

```
public class Pallindrome {
    public static void main(String[] args) {
        System.out.print("Enter the string: ");
        String str = System.console().readLine();
        if (isPallindrome(str)) {
            System.out.println(str + " is a pallindrome.");
        } else {
            System.out.println(str + " is not a pallindrome.");
        }
    }

    public static boolean isPallindrome(String str) {
        int i = 0;
        int j = str.length() - 1;
        while (i < j) {
            if (str.charAt(i) != str.charAt(j)) {
                return false;
            }
            i++;
            j--;
        }
        return true;
    }
}
```