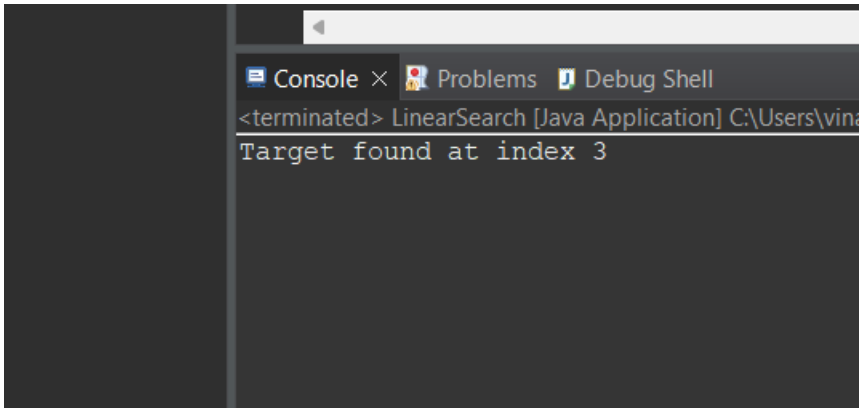
1.Writing a program in Java implementing the linear search algorithm.?

Source Code:

```
package smpilelearn;

public class LinearSearch {
    public static int linearSearch(int[] array, int target) {
        for (int i = 0; i < array.length; i++) {
            if (array[i] == target) {
                return i; // Return the index if the target is found
            }
        }
        return -1; // Return -1 if the target is not found
    }

    public static void main(String[] args) {
        int[] array = {5, 10, 15, 20, 25, 30};
        int target = 20;

        int index = linearSearch(array, target);

        if (index != -1) {
            System.out.println("Target found at index " + index);
        } else {
            System.out.println("Target not found");
        }
    }
}
```

Output:



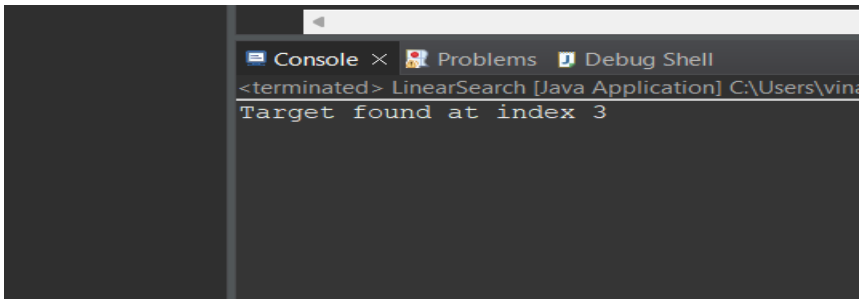
2. Writing a program in Java implementing the binary search algorithm..?

Source Code:

```
public class BinarySearch {

    public static int binarySearch(int[] array, int target) {

        int left = 0;

        int right = array.length - 1;
```

```
        while (left <= right) {
            int mid = left + (right - left) / 2;

            if (array[mid] == target) {
                return mid; // Return the index if the target is found
            } else if (array[mid] < target) {
                left = mid + 1; // Target is in the right half
            } else {
                right = mid - 1; // Target is in the left half
            }
        }

        return -1; // Return -1 if the target is not found
    }

    public static void main(String[] args) {
        int[] array = {5, 10, 15, 20, 25, 30};
        int target = 20;

        int index = binarySearch(array, target);

        if (index != -1) {
            System.out.println("Target found at index " + index);
        } else {
            System.out.println("Target not found");
        }
    }
}
```

Output:



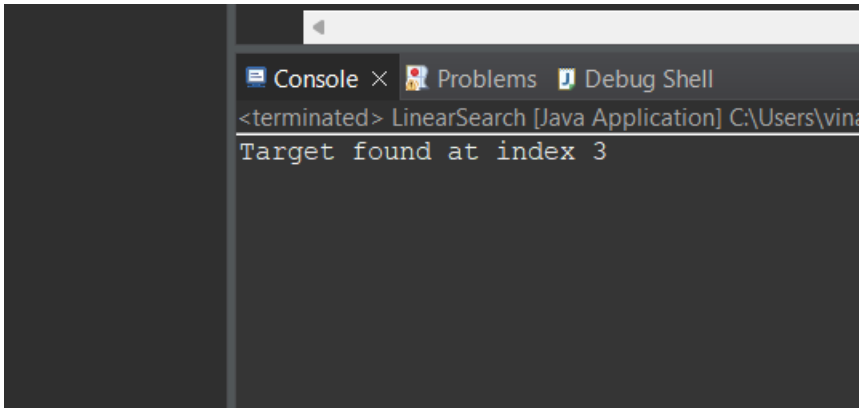
3. Writing a program in Java implementing the exponential search algorithm..?

Source Code:

```
public class ExponentialSearch {
    public static int exponentialSearch(int[] array, int target) {
        if (array[0] == target) {
            return 0; // Return index 0 if target is at the beginning
        }

        int i = 1;
        while (i < array.length && array[i] <= target) {
            i *= 2; // Double the index to expand the search range
        }

        return binarySearch(array, target, i / 2, Math.min(i, array.length - 1));
    }

    public static int binarySearch(int[] array, int target, int left, int right) {
        while (left <= right) {
            int mid = left + (right - left) / 2;

            if (array[mid] == target) {
                return mid; // Return the index if the target is found
            } else if (array[mid] < target) {
                left = mid + 1; // Target is in the right half
            } else {
```

```
            right = mid - 1; // Target is in the left half
        }
    }

    return -1; // Return -1 if the target is not found
  }

  public static void main(String[] args) {
    int[] array = {5, 10, 15, 20, 25, 30};
    int target = 20;

    int index = exponentialSearch(array, target);

    if (index != -1) {
      System.out.println("Target found at index " + index);
    } else {
      System.out.println("Target not found");
    }
  }
}
```

Output:



4. Writing a program in Java implementing the selection sort algorithm..?

SourceCode:

```
public class SelectionSort {
```

```
public static void selectionSort(int[] array) {
    int n = array.length;

    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (array[j] < array[minIndex]) {
                minIndex = j;
            }
        }

        // Swap the minimum element with the current element
        int temp = array[minIndex];
        array[minIndex] = array[i];
        array[i] = temp;
    }
}

public static void main(String[] args) {
    int[] array = {5, 2, 8, 12, 1, 6};

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

    selectionSort(array);

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

public static void printArray(int[] array) {
```

```
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }
}
```

Output:

```
Before sorting:
5 2 8 12 1 6
After sorting:
1 2 5 6 8 12
```

5. Writing a program in Java implementing the bubble sort algorithm..?

SourceCode:

```
public class BubbleSort {
    public static void bubbleSort(int[] array) {
        int n = array.length;

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

```
    public static void main(String[] args) {
        int[] array = {5, 2, 8, 12, 1, 6};

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

        bubbleSort(array);

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

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

Output:

```
Before sorting:
5 2 8 12 1 6
After sorting:
1 2 5 6 8 12
```

6. Writing a program in Java implementing the insertion sort algorithm..?

SourceCode:

```
public class InsertionSort {
    public static void insertionSort(int[] array) {
        int n = array.length;
```

```
    for (int i = 1; i < n; i++) {
        int key = array[i];
        int j = i - 1;

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

        array[j + 1] = key;
    }
}

public static void main(String[] args) {
    int[] array = {5, 2, 8, 12, 1, 6};

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

    insertionSort(array);

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

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

```
    }
}
```

Output:

```
Before sorting:
5 2 8 12 1 6
After sorting:
1 2 5 6 8 12
```

7. Writing a program in Java implementing the merge sort algorithm…?

SourceCode:

```
public class MergeSort {
    public static void mergeSort(int[] array) {
        if (array.length <= 1) {
            return; // Base case: already sorted
        }

        int mid = array.length / 2;
        int[] left = new int[mid];
        int[] right = new int[array.length - mid];

        // Divide the array into two halves
        System.arraycopy(array, 0, left, 0, mid);
        System.arraycopy(array, mid, right, 0, array.length - mid);

        // Recursively sort the two halves
        mergeSort(left);
        mergeSort(right);

        // Merge the sorted halves
        merge(array, left, right);
    }
```

```
public static void merge(int[] array, int[] left, int[] right) {
    int i = 0; // Index for left array
    int j = 0; // Index for right array
    int k = 0; // Index for merged array

    while (i < left.length && j < right.length) {
        if (left[i] <= right[j]) {
            array[k] = left[i];
            i++;
        } else {
            array[k] = right[j];
            j++;
        }
        k++;
    }

    // Copy remaining elements from left array, if any
    while (i < left.length) {
        array[k] = left[i];
        i++;
        k++;
    }

    // Copy remaining elements from right array, if any
    while (j < right.length) {
        array[k] = right[j];
        j++;
        k++;
    }
}
```

```
    public static void main(String[] args) {
        int[] array = {5, 2, 8, 12, 1, 6};

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

        mergeSort(array);

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

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

Output:

```
Before sorting:
5 2 8 12 1 6
After sorting:
1 2 5 6 8 12
```

8. Writing a program in Java implementing the quick sort algorithm..?

SourceCode:

```
public class QuickSort {
    public static void quickSort(int[] array) {
        quickSort(array, 0, array.length - 1);
```

```
}

public static void quickSort(int[] array, int low, int high) {
    if (low < high) {
        int pivotIndex = partition(array, low, high);
        quickSort(array, low, pivotIndex - 1);
        quickSort(array, pivotIndex + 1, high);
    }
}

public static int partition(int[] array, int low, int high) {
    int pivot = array[high];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (array[j] <= pivot) {
            i++;
            swap(array, i, j);
        }
    }

    swap(array, i + 1, high);
    return i + 1;
}

public static void swap(int[] array, int i, int j) {
    int temp = array[i];
    array[i] = array[j];
    array[j] = temp;
}
```

```
    public static void main(String[] args) {
        int[] array = {5, 2, 8, 12, 1, 6};

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

        quickSort(array);

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

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

Output:

```
Before sorting:
5 2 8 12 1 6
After sorting:
1 2 5 6 8 12
```