S.No	Program Description	Remarks
1	Program to find the sum of array elements	
2	Program to find the largest element in an array	
3	Program to search an element using linear search in an array	
4	Program to search an element using binary search in an array	
5	Program to sort the elements of an array using bubble and insertion sort	
6	Program to sort the elements of an array using the quick sort algorithm	
7	Program to solve the maximum sub-array problem	
8	Program to display Binary Tree Traversal methods: Pre-order, In-order, and Post-order using recursive algorithms	
9	Program to carry out searching, insertion, and deletion operations in a Binary Search Tree	
10	Program to carry out insertion and deletion operations in an AVL Tree	
11	Program to carry out insertion and deletion in B-trees (with all the cases)	
12	Program to find breadth-first traversals of a given graph	
13	Program to find depth-first traversals of a given graph	
14	Program to find Dijkstra’s shortest path algorithm using a greedy approach	
15	Program to find Prim’s minimum spanning tree algorithm using a greedy approach	
16	Program to implement min-heap and max-heap	

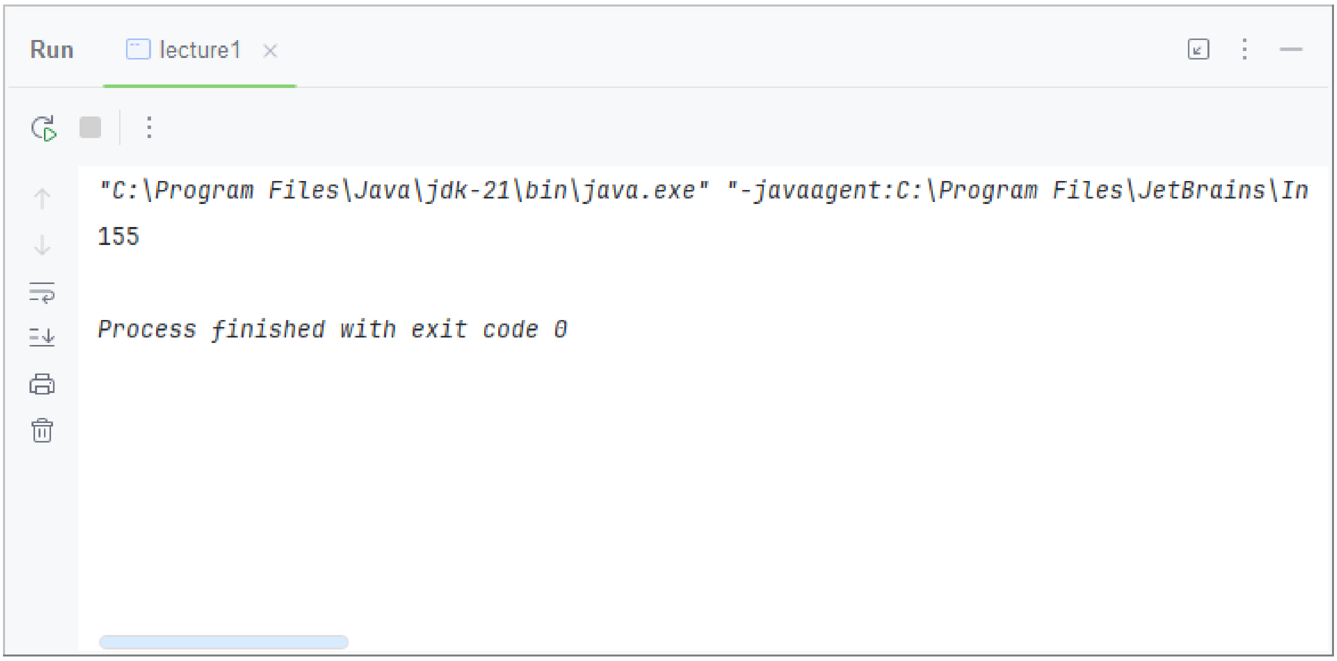
Program 1

A program to find the sum of array elements

Code

```
 public static int sumofarrayelements(int[] nums){
  int sum =0;
  for(int n :nums){
    sum+=n;
  }
  return sum;
}
public static void main(String[] args){
  int[] nums = new int[]{2,4,6,2,6,12,45,78};
  System.out.println(sumofarrayelements(nums));}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
155

Process finished with exit code 0
```

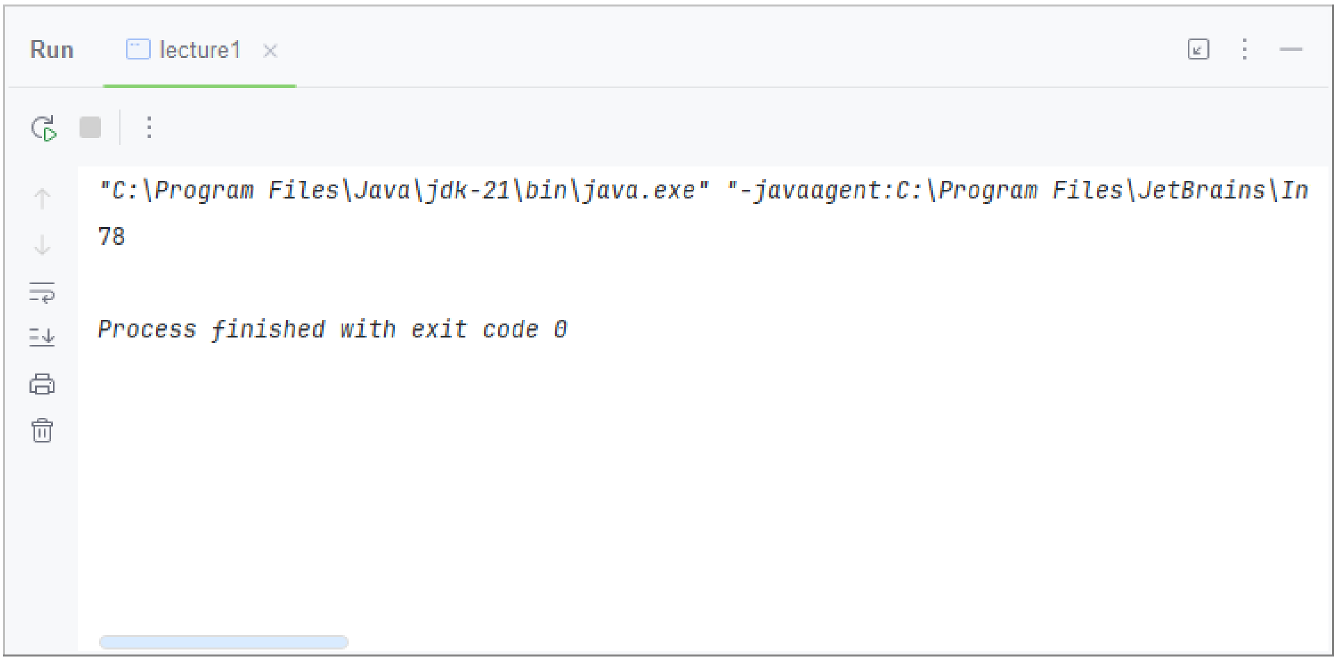
Program 2

Program to find the largest element in array

Code

```
public static int largestelementinarray(int[] nums){
  int max =-(int)(1e9);
  for (int n:nums){
    if(n>max){max=n;}
  }
  return max;
}
public static void main(String[] args){
  int[] nums = new int[]{2,4,6,2,6,12,45,78};
  System.out.println(largestelementinarray(nums));
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
78

Process finished with exit code 0
```

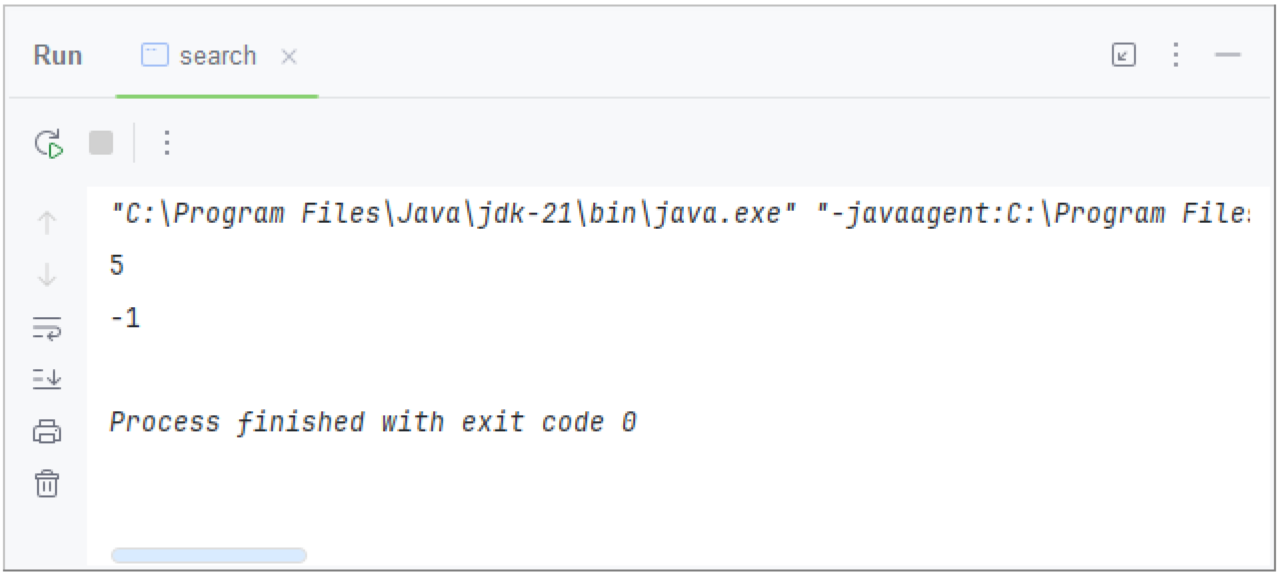
Program 3

Program to search an element using linear search in array

Code

```
 public static int linearsearch(int[] nums, int target)
{
    for (int i = 0; i < nums.length ; i++) {
      if(nums[i]==target){return i;}
    }
    return -1;
  }

  public static void main(String[] args) {
    int[] nums = new int[]{12,23,45,56,65,78,98};
    System.out.println(linearsearch(nums, 78));
   System.out.println(linearsearch(nums, 88));
  }
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program File
5
-1


Process finished with exit code 0
```

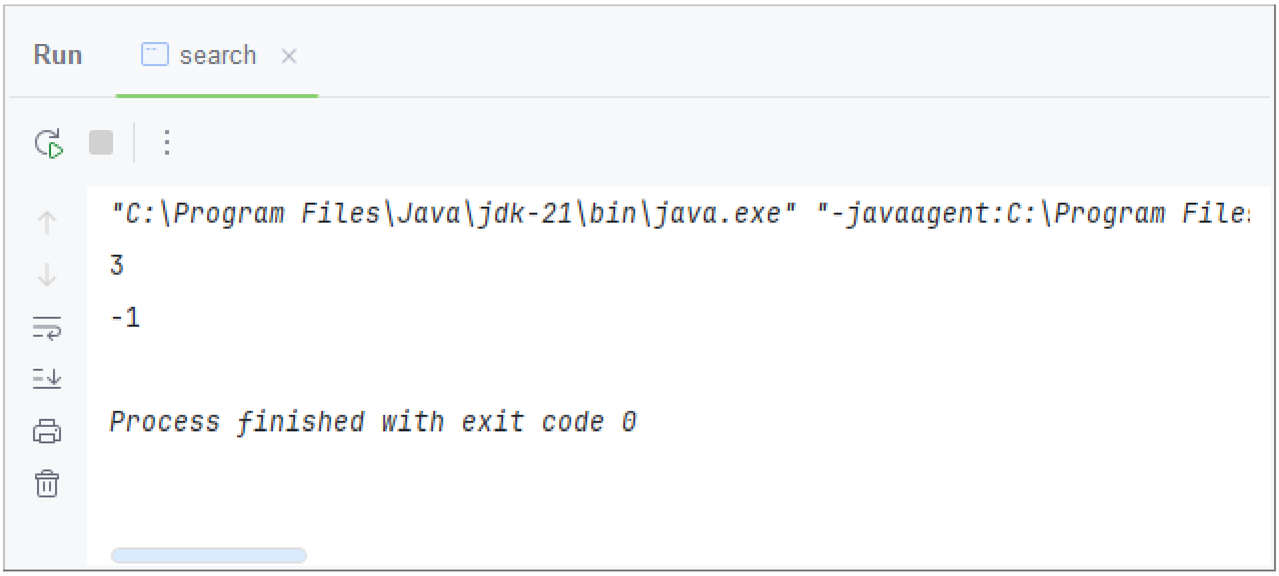
Program 4

Program to search an element using binary search in array

Code

```
public static int binarysearch(int[] nums,int target){
  int left =0;
  int right = nums.length-1;
  while(right>=left){
    int mid = right-(right-left)/2;
    if(nums[mid]==target){return mid;}
    else if (nums[mid]>target) {right=mid-1;}
    else{left=mid+1;}
  }
  return -1;
 }
 public static void main(String[] args) {
  int[] nums = new int[]{12,23,45,56,65,78,98};
  System.out.println(binaryearch(nums, 56));
   System.out.println(binarysearch(nums, 99));
 }
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program File:
3
-1

Process finished with exit code 0
```

Program 5

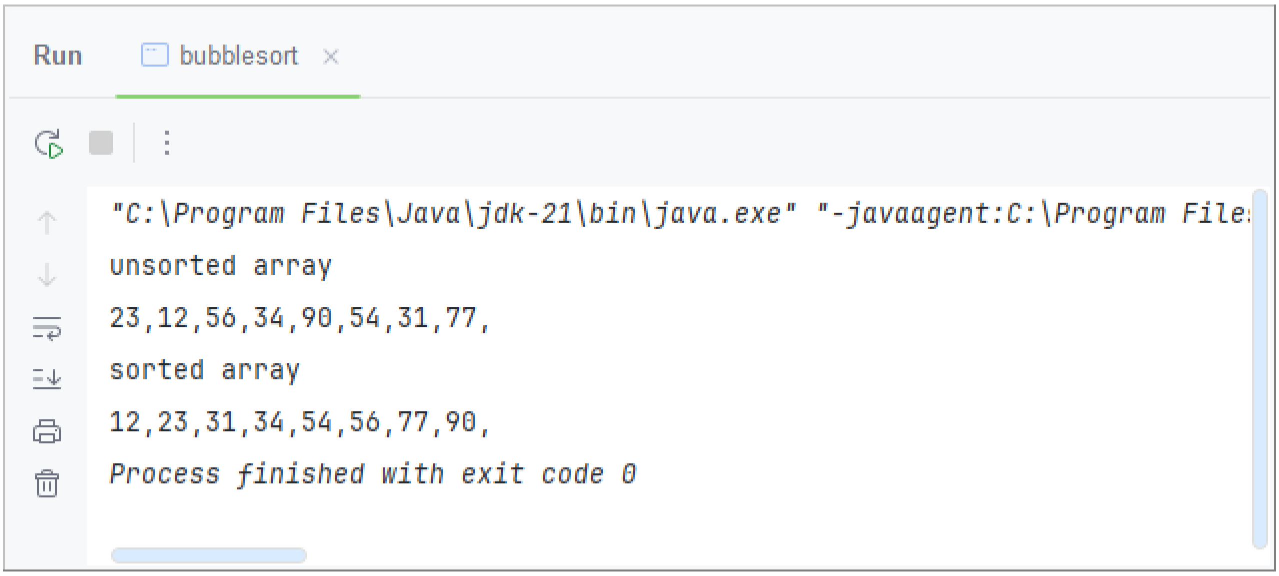
Program to sort the element of an array using bubble and insertion sort

Code

```
public static void bubble(int[] nums){
 if(nums.length<2){return;}
 for (int i = nums.length-1; i >=0 ; i--) {
  for (int j = 0; j <i ; j++) {
   if (nums[j+1]<nums[j]){
    int temp = nums[j];
    nums[j]=nums[j+1];
    nums[j+1]= temp;
   }
  }
 }
}
public static void inserted(int[] nums){
 for (int i = 1; i < nums.length ; i++) {
  for (int j = i-1; j >=0 ; j--) {
   if (nums[j+1]<nums[j]){
    int temp = nums[j+1];
    nums[j+1]=nums[j];
    nums[j]=temp;
   }else{break;}
  }
 }
}

public static void main(String[] args) {
 int[] nums = new int[]{23,12,56,34,90,54,31,77};
 System.out.println(“unsorted array”);
  for(int n :nums){
  System.out.print(n+",");
```

```
      }
 System.out.println();
bubble(nums);
inserted(nums);
 System.out.println(“sorted array”);
for(int n :nums){
  System.out.print(n+",");
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program File
unsorted array
23,12,56,34,90,54,31,77,
sorted array
12,23,31,34,54,56,77,90,
Process finished with exit code 0
```

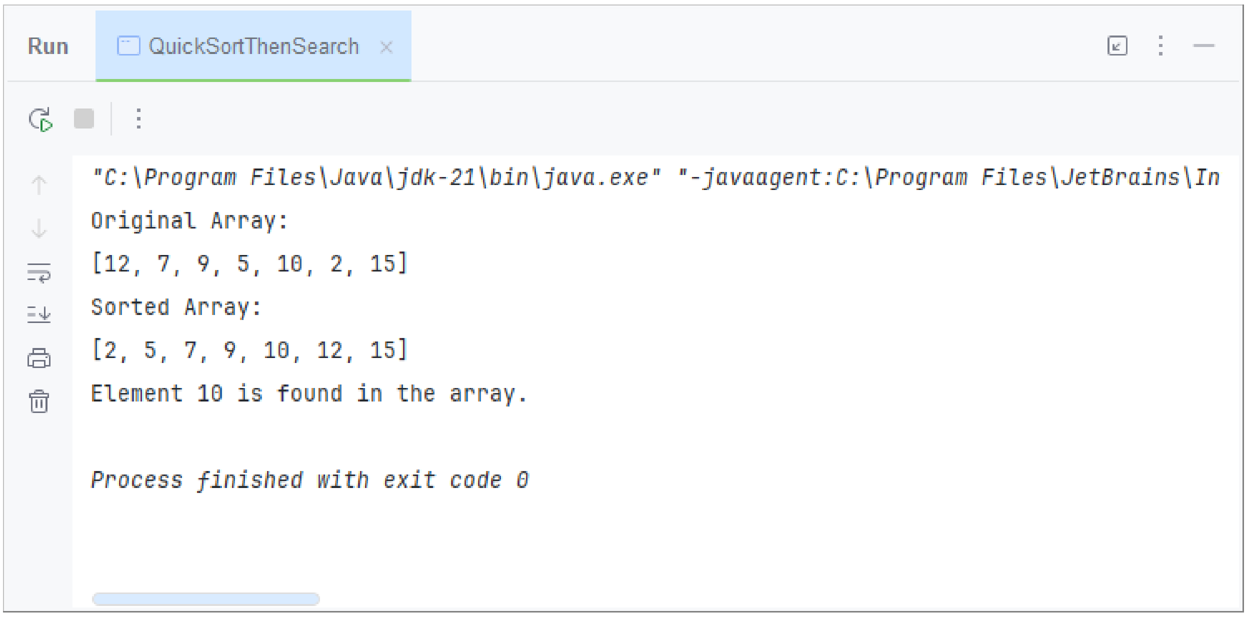
Program 6

A program to find the element of an array using a quick sort algorithm

Code

```
public class QuickSortThenSearch {

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);
  }
}
private 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++;
      int temp = array[i];
      array[i] = array[j];
      array[j] = temp;
    }
  }
  int temp = array[i + 1];
  array[i + 1] = array[high];
  array[high] = temp;

  return i + 1;
}
public static boolean binarySearch(int[] array, int element) {
  int low = 0, high = array.length - 1;
  while (low <= high) {
    int mid = low + (high - low) / 2;
    if (array[mid] == element) {
      return true;
    } else if (array[mid] < element) {
      low = mid + 1;
    } else {
        high = mid - 1;
      }
    }
    return false;
  }

  public static void main(String[] args) {
    int[] array = {12, 7, 9, 5, 10, 2, 15};
    int target = 10;

    System.out.println("Original Array:");
    System.out.println(Arrays.toString(array));

    quickSort(array, 0, array.length - 1);

    System.out.println("Sorted Array:");
    System.out.println(Arrays.toString(array));

    boolean found = binarySearch(array, target);

    if (found) {
      System.out.println("Element " + target + " is found in the
array.");
    } else {
      System.out.println("Element " + target + " is not found in
the array.");
    }
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
Original Array:
[12, 7, 9, 5, 10, 2, 15]
Sorted Array:
[2, 5, 7, 9, 10, 12, 15]
Element 10 is found in the array.

Process finished with exit code 0
```

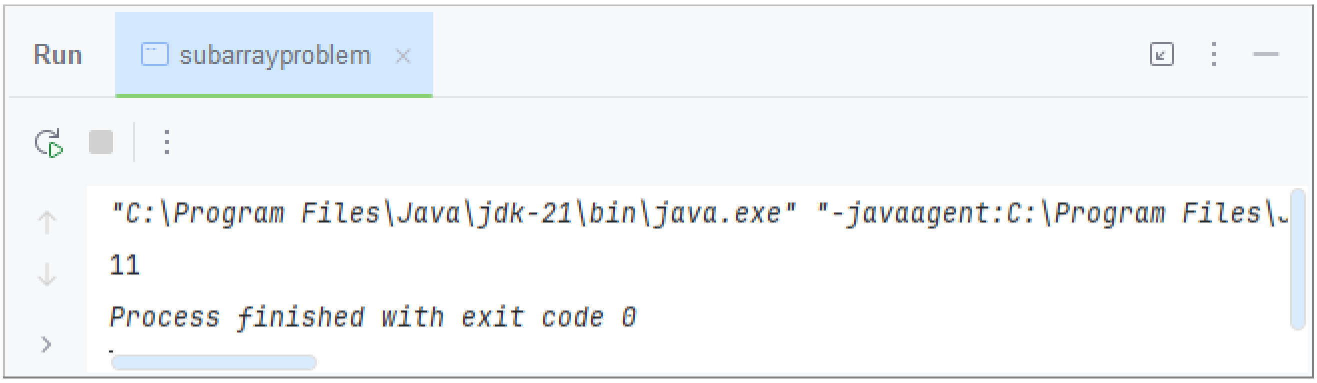
Program 7

A program to solve maximum sub-array problem

Code

```
public static int subarray(int[] nums){
  int sum =0;
  int mx =0;
  for (int i = 0; i < nums.length ; i++) {
    if (sum+nums[i]>=0){sum+=nums[i];}
    else{sum=0;}
    if (sum>mx){mx=sum;}
  }
  if (mx==0){
    mx=Integer.MIN_VALUE;
    for (int j = 0; j < nums.length ; j++) {
      mx=Math.max(mx,nums[j]);
    }
  }
  return mx;
}
public static void main(String[] args) {
  int[] zaza = new int[]{2,3,-8,7,-1,2,3};
  System.out.print(subarray(zaza));
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
11

Process finished with exit code 0
```

Program 8

A program to display Binary tree traversal methods: Pre-order, In-order, Post-order using recursive algorithm

Code

```
  public class Traversal {

  class TreeNode{
    int val;
    TreeNode left;
    TreeNode right;
  }
  public void inorder(TreeNode root){
    if(root==null){return;}
    inorder(root.left);
    System.out.println(root.val);
    inorder(root.right);
  }
  public void preorder(TreeNode root){
    if (root==null){return;}
    System.out.println(root.val);
    preorder(root.left);
    preorder(root.right);
  }
  public void postorder(TreeNode root){
    if(root==null){return;}
    postorder(root.left);
    postorder(root.right);
    System.out.println(root.val);
  }
}
```

Input

```
      1
     / \
    2   3
   / \    \
  4   5    6
```

nums = {1, 2, 3, 4, 5, null, 6};

Output

Traversal Type	Order
Inorder	`[4, 2, 5, 1, 3, 6]`
Preorder	`[1, 2, 4, 5, 3, 6]`
Postorder	`[4, 5, 2, 6, 3, 1]`

Program 9

A program to carry out searching, insertion and deletion operation in Binary Search Tree

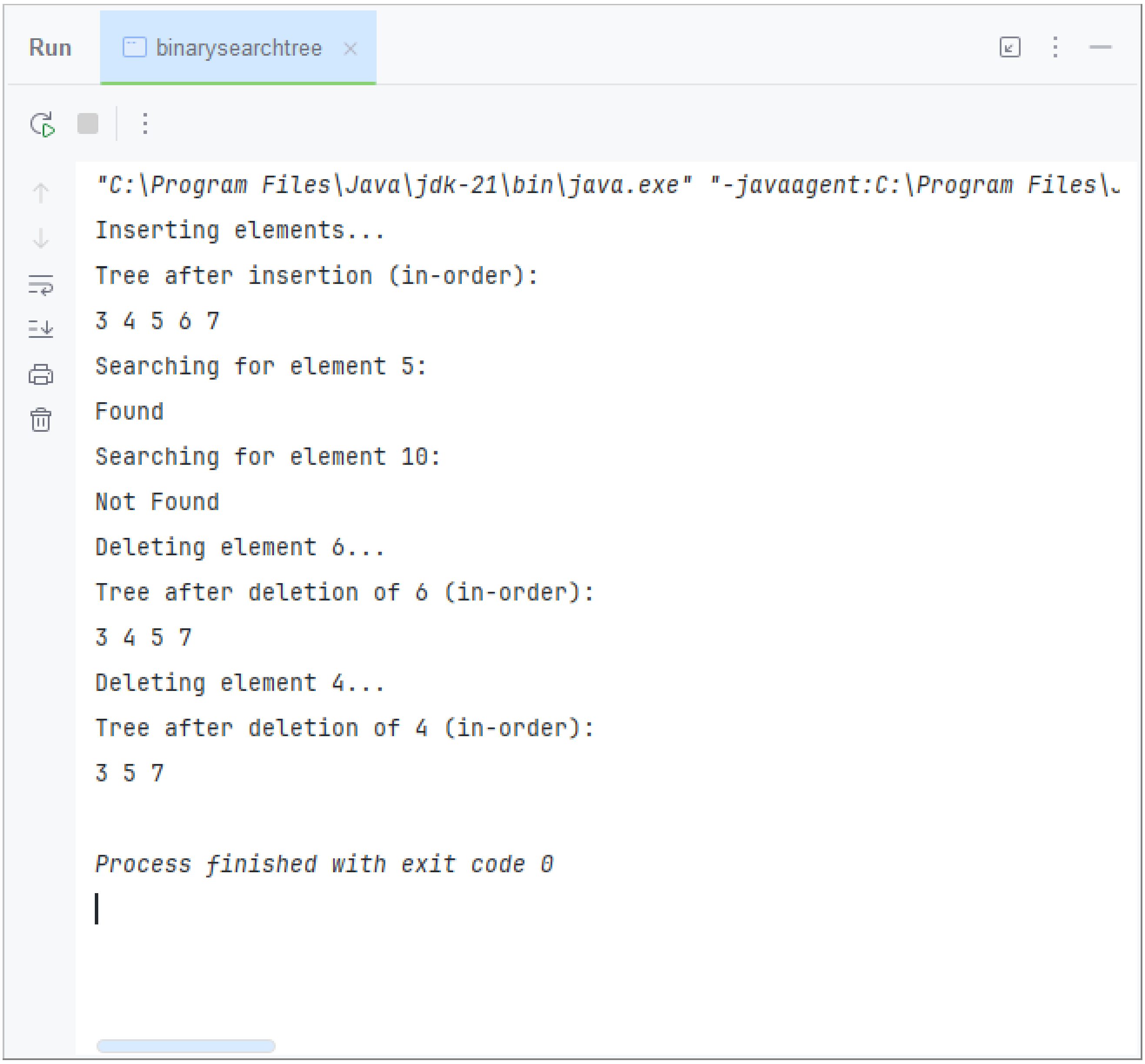
Code

```
public class binarysearchtree {

 class node {
   node left;
   node right;
   int val;
   public node(int val) {
     this.val = val;
   }
 }
 node root;
 public void insert(int val) {
   root = inserter(root, val);
 }
 private node inserter(node root, int val) {
   if (root == null) {
     return new node(val);
   }
   if (root.val > val) {
     root.left = inserter(root.left, val);
   } else {
     root.right = inserter(root.right, val);
   }
   return root;
 }
 public node search(int val) {
   return searcher(val, root);
 }
 private node searcher(int val, node root) {
   if (root == null) {
     return null;
   }
   if (root.val == val) {
     return root;
   }
```

```
  if (root.val > val) {
    return searcher(val, root.left);
  } else {
    return searcher(val, root.right);
  }
}
public void delete(int val) {
  root = deleter(val, root);
}
private node deleter(int val, node root) {
  if (root == null) {
    return null;
  }
  if (root.val == val) {
    // Case 1: No child or one child
    if (root.left == null) {
      return root.right;
    }
    if (root.right == null) {
      return root.left;
    }
    // Case 2: Two children
    node temp = next(root.right);
    root.val = temp.val;
    root.right = deleter(temp.val, root.right);
  } else if (root.val > val) {
    root.left = deleter(val, root.left);
  } else {
    root.right = deleter(val, root.right);
  }
  return root;
}
private node next(node root) {
  node temp = root;
  while (temp.left != null) {
    temp = temp.left;
  }
  return temp;
}
public void inOrderTraversal(node root) {
  if (root != null) {
    inOrderTraversal(root.left);
    System.out.print(root.val + " ");
```

```
      inOrderTraversal(root.right);
    }
  }
  public static void main(String[] args) {
    binarysearchtree t = new binarysearchtree();
    System.out.println("Inserting elements...");
    t.insert(4);
    t.insert(3);
    t.insert(6);
    t.insert(7);
    t.insert(5);
    System.out.println("Tree after insertion (in-order):");
    t.inOrderTraversal(t.root);
    System.out.println();
    System.out.println("Searching for element 5:");
    System.out.println(t.search(5) != null ? "Found" : "Not Found");
    System.out.println("Searching for element 10:");
    System.out.println(t.search(10) != null ? "Found" : "Not Found");
    System.out.println("Deleting element 6...");
    t.delete(6);
    System.out.println("Tree after deletion of 6 (in-order):");
    t.inOrderTraversal(t.root);
    System.out.println();
    System.out.println("Deleting element 4...");
    t.delete(4);
    System.out.println("Tree after deletion of 4 (in-order):");
    t.inOrderTraversal(t.root);
    System.out.println();
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
Inserting elements...
Tree after insertion (in-order):
3 4 5 6 7
Searching for element 5:
Found
Searching for element 10:
Not Found
Deleting element 6...
Tree after deletion of 6 (in-order):
3 4 5 7
Deleting element 4...
Tree after deletion of 4 (in-order):
3 5 7

Process finished with exit code 0
```

Program 10

Program to carry out insertion and deletion operations in AVL Tree

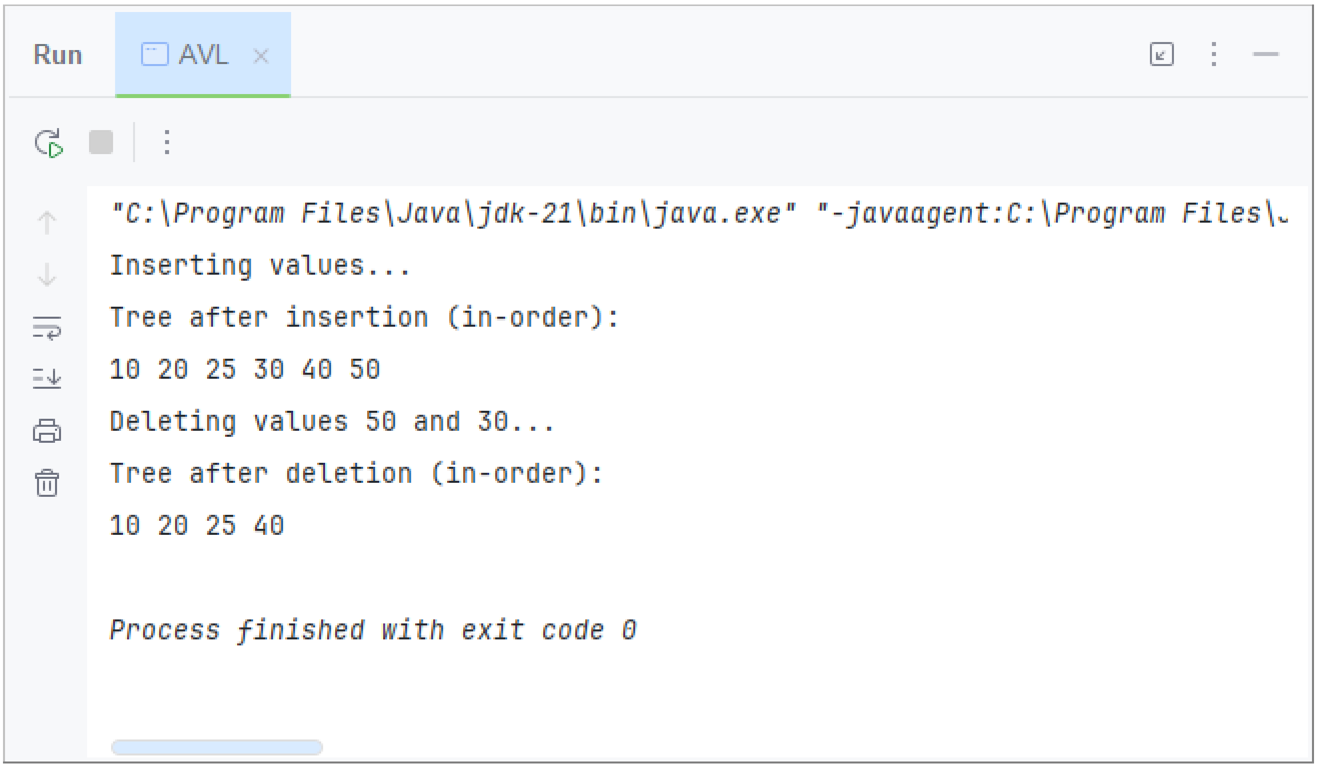
Code

```
public class AVL {

Node root;
class Node {
  int val;
  int height;
  Node left;
  Node right;
  Node(int a) {
    this.val = a;
  }
}
public void add(int a) {
  this.root = insert(a, root);
}
public int height(Node root) {
  if (root == null) {
    return -1;
  } else {
    return root.height;
  }
}
private Node insert(int a, Node root) {
  if (root == null) {
    return new Node(a);
  }
  if (root.val > a) {
    root.left = insert(a, root.left);
  } else {
    root.right = insert(a, root.right);
  }
  int l = height(root.left);
  int r = height(root.right);
  root.height = Math.max(l, r) + 1;
  return balance(root);
  }
  private Node balance(Node root) {
    if (root == null) {
      return null;}
    if (height(root.left) - height(root.right) > 1) {
      if (height(root.left.left) > height(root.left.right)) {
        return Rightrotate(root); // Left-Left case
      } else {
        root.left = Leftrotate(root.left); // Left-Right case
        return Rightrotate(root);
      }
    } else if (height(root.right) - height(root.left) > 1) {
      if (height(root.right.left) > height(root.right.right)) {
        root.right = Rightrotate(root.right); // Right-Left case
        return Leftrotate(root);
      } else {
        return Leftrotate(root); // Right-Right case
      }
    }
    return root;
  }
  private Node Leftrotate(Node parent) {
    Node child = parent.right;
    parent.right = child.left;
    child.left = parent;
    child.height = Math.max(height(child.left),
height(child.right)) + 1;
    parent.height = Math.max(height(parent.left),
height(parent.right)) + 1;
    return child;
  }
  private Node Rightrotate(Node parent) {
    Node child = parent.left;
    parent.left = child.right;
    child.right = parent;
    child.height = Math.max(height(child.left),
height(child.right)) + 1;
    parent.height = Math.max(height(parent.left),
height(parent.right)) + 1;
    return child;
  }
  public void delete(int val) {
    root = deleted(val, root);
```

```
}
private Node deleted(int val, Node root) {
  if (root == null) {
    return null;
  }
  if (val > root.val) {
    root.right = deleted(val, root.right);
  } else if (val < root.val) {
    root.left = deleted(val, root.left);
  } else {
    if (root.left == null) {
      return root.right;
    }
    if (root.right == null) {
      return root.left;
    }
    Node suc = findMin(root.right);
    root.val = suc.val;
    root.right = deleted(suc.val, root.right);
  }
  int l = height(root.left);
  int r = height(root.right);
  root.height = Math.max(l, r) + 1;
  return balance(root);
}
private Node findMin(Node root) {
  while (root.left != null) {
    root = root.left;
  }
  return root;
}
// In-order Traversal for Display
public void inOrderTraversal(Node root) {
  if (root != null) {
    inOrderTraversal(root.left);
    System.out.print(root.val + " ");
    inOrderTraversal(root.right);
  }
}
public static void main(String[] args) {
  AVL tree = new AVL();
  System.out.println("Inserting values...");
  tree.add(10);
```

```
        tree.add(20);
        tree.add(30);
        tree.add(40);
        tree.add(50);
        tree.add(25);

        System.out.println("Tree after insertion (in-order):");
        tree.inOrderTraversal(tree.root);
        System.out.println();

        System.out.println("Deleting values 50 and 30...");
        tree.delete(50);
        tree.delete(30);

        System.out.println("Tree after deletion (in-order):");
        tree.inOrderTraversal(tree.root);
        System.out.println();
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
Inserting values...
Tree after insertion (in-order):
10 20 25 30 40 50
Deleting values 50 and 30...
Tree after deletion (in-order):
10 20 25 40


Process finished with exit code 0
```

Program 11

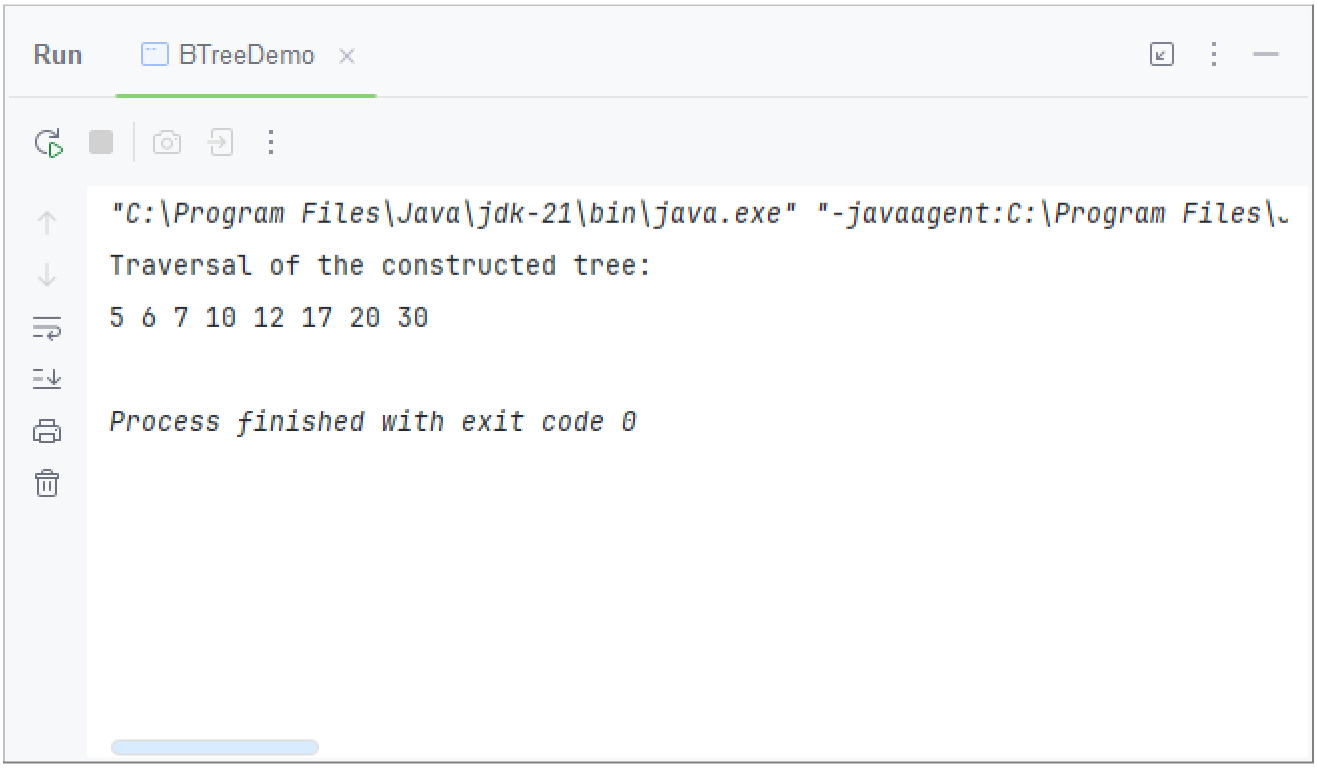
A program to carry out insertion and deletion in B trees

Code

```
class BTreeNode {

  int[] keys;             // Array of keys
  int t;                  // Minimum degree
  BTreeNode[] children;   // Array of child pointers
  int n;                  // Current number of keys
  boolean isLeaf;         // True if leaf node

  public BTreeNode(int t, boolean isLeaf) {
    this.t = t;
    this.isLeaf = isLeaf;
    keys = new int[2 * t - 1];
    children = new BTreeNode[2 * t];
    n = 0;
  }

  public void traverse() {
    for (int i = 0; i < n; i++) {
      if (!isLeaf) children[i].traverse();
      System.out.print(keys[i] + " ");
    }
    if (!isLeaf) children[n].traverse();
  }

  public BTreeNode search(int key) {
    int i = 0;
    while (i < n && key > keys[i]) i++;
    if (i < n && keys[i] == key) return this;
    if (isLeaf) return null;
    return children[i].search(key);
  }

  void insertNonFull(int key) {
    int i = n - 1;
    if (isLeaf) {
      while (i >= 0 && keys[i] > key) {
        keys[i + 1] = keys[i];
        i--;
      }
      keys[i + 1] = key;
      n++;
    } else {
      while (i >= 0 && keys[i] > key) i--;

      if (children[i + 1].n == 2 * t - 1) {
        splitChild(i + 1, children[i + 1]);
        if (keys[i + 1] < key) i++;
      }
      children[i + 1].insertNonFull(key);
    }
  }

  void splitChild(int i, BTreeNode y) {
    BTreeNode z = new BTreeNode(y.t, y.isLeaf);
    z.n = t - 1;

    for (int j = 0; j < t - 1; j++) z.keys[j] = y.keys[j + t];
    if (!y.isLeaf) for (int j = 0; j < t; j++) z.children[j] =
y.children[j + t];

    y.n = t - 1;

    for (int j = n; j >= i + 1; j--) children[j + 1] =
children[j];
    children[i + 1] = z;

    for (int j = n - 1; j >= i; j--) keys[j + 1] = keys[j];
    keys[i] = y.keys[t - 1];
    n++;
  }

  int findKey(int key) {
    int idx = 0;
    while (idx < n && keys[idx] < key) idx++;
    return idx;
  }
  void remove(int key) {
    int idx = findKey(key);

    if (idx < n && keys[idx] == key) {
      if (isLeaf) removeFromLeaf(idx);
      else removeFromNonLeaf(idx);
    } else {
      if (isLeaf) {
        System.out.println("Key " + key + " does not exist in the 
tree.");
        return;
      }
      boolean flag = idx == n;
      if (children[idx].n < t) fill(idx);
      if (flag && idx > n) children[idx - 1].remove(key);
      else children[idx].remove(key);
    }
  }

  void removeFromLeaf(int idx) {
    for (int i = idx + 1; i < n; i++) keys[i - 1] = keys[i];
    n--;
  }

  void removeFromNonLeaf(int idx) {
    int key = keys[idx];

    if (children[idx].n >= t) {
      int pred = getPred(idx);
      keys[idx] = pred;
      children[idx].remove(pred);
    } else if (children[idx + 1].n >= t) {
      int succ = getSucc(idx);
      keys[idx] = succ;
      children[idx + 1].remove(succ);
    } else {
      merge(idx);
      children[idx].remove(key);
    }
  }

  int getPred(int idx) {
    BTreeNode cur = children[idx];
    while (!cur.isLeaf) cur = cur.children[cur.n];
    return cur.keys[cur.n - 1];
  }

  int getSucc(int idx) {
    BTreeNode cur = children[idx + 1];
    while (!cur.isLeaf) cur = cur.children[0];
    return cur.keys[0];
  }

  void fill(int idx) {
    if (idx != 0 && children[idx - 1].n >= t)
borrowFromPrev(idx);
    else if (idx != n && children[idx + 1].n >= t)
borrowFromNext(idx);
    else {
      if (idx != n) merge(idx);
      else merge(idx - 1);
    }
  }

  void borrowFromPrev(int idx) {
    BTreeNode child = children[idx];
    BTreeNode sibling = children[idx - 1];

    for (int i = child.n - 1; i >= 0; i--) child.keys[i + 1] =
child.keys[i];
    if (!child.isLeaf) for (int i = child.n; i >= 0; i--)
child.children[i + 1] = child.children[i];

    child.keys[0] = keys[idx - 1];
    if (!child.isLeaf) child.children[0] =
sibling.children[sibling.n];
    keys[idx - 1] = sibling.keys[sibling.n - 1];
    child.n++;
    sibling.n--;
  }

  void borrowFromNext(int idx) {
    BTreeNode child = children[idx];
    BTreeNode sibling = children[idx + 1];

    child.keys[child.n] = keys[idx];
    if (!child.isLeaf) child.children[child.n + 1] =
sibling.children[0];

    keys[idx] = sibling.keys[0];
    for (int i = 1; i < sibling.n; i++) sibling.keys[i - 1] =
sibling.keys[i];
    if (!sibling.isLeaf) for (int i = 1; i <= sibling.n; i++)
sibling.children[i - 1] = sibling.children[i];

    child.n++;
    sibling.n--;
  }

  void merge(int idx) {
    BTreeNode child = children[idx];
    BTreeNode sibling = children[idx + 1];

    child.keys[t - 1] = keys[idx];
    for (int i = 0; i < sibling.n; i++) child.keys[i + t] =
sibling.keys[i];
    if (!child.isLeaf) for (int i = 0; i <= sibling.n; i++)
child.children[i + t] = sibling.children[i];

    for (int i = idx + 1; i < n; i++) keys[i - 1] = keys[i];
    for (int i = idx + 2; i <= n; i++) children[i - 1] =
children[i];

    child.n += sibling.n + 1;
    n--;
  }
}

class BTree {
  BTreeNode root;
  int t;

  public BTree(int t) {
    this.t = t;
    root = null;
  }

  public void traverse() {
    if (root != null) root.traverse();
  }
  public BTreeNode search(int key) {
    return root == null ? null : root.search(key);
  }

  public void insert(int key) {
    if (root == null) {
      root = new BTreeNode(t, true);
      root.keys[0] = key;
      root.n = 1;
    } else {
      if (root.n == 2 * t - 1) {
        BTreeNode s = new BTreeNode(t, false);
        s.children[0] = root;
        s.splitChild(0, root);
        int i = s.keys[0] < key ? 1 : 0;
        s.children[i].insertNonFull(key);
        root = s;
      } else {
        root.insertNonFull(key);
      }
    }
  }

  public void remove(int key) {
    if (root == null) {
      System.out.println("The tree is empty.");
      return;
    }
    root.remove(key);
    if (root.n == 0) {
      root = root.isLeaf ? null : root.children[0];
    }
  }
}

public class BTreeDemo {
  public static void main(String[] args) {
    BTree tree = new BTree(3);

    tree.insert(10);
    tree.insert(20);
    tree.insert(5);
```

```
        tree.insert(6);
        tree.insert(12);
        tree.insert(30);
        tree.insert(7);
        tree.insert(17);

        System.out.println("Traversal of the constructed tree:");
        tree.traverse();
        System.out.println();
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
Traversal of the constructed tree:
5 6 7 10 12 17 20 30

Process finished with exit code 0
```

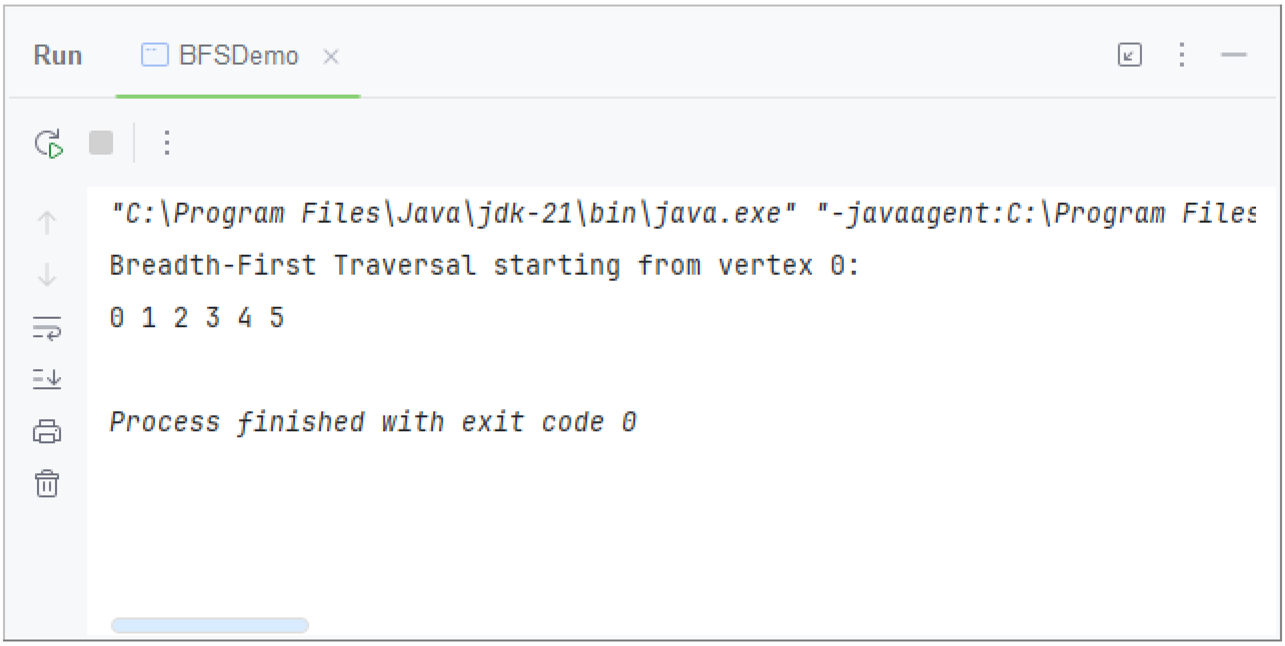
Program 12

A program to find out breadth-first traversals of a given graph.

Code

```
class Graph {

  private int vertices;
  private LinkedList<Integer>[] adjacencyList;

  public Graph(int vertices) {
    this.vertices = vertices;
    adjacencyList = new LinkedList[vertices];
    for (int i = 0; i < vertices; i++) {
      adjacencyList[i] = new LinkedList<>();
    }
  }

  public void addEdge(int source, int destination) {
    adjacencyList[source].add(destination);
  }

  public void BFS(int startVertex) {
    boolean[] visited = new boolean[vertices];
    Queue<Integer> queue = new LinkedList<>();

    visited[startVertex] = true;
    queue.add(startVertex);

    System.out.println("Breadth-First Traversal starting from
vertex " + startVertex + ":");

    while (!queue.isEmpty()) {
      int currentVertex = queue.poll();
      System.out.print(currentVertex + " ");

      for (int neighbor : adjacencyList[currentVertex]) {
        if (!visited[neighbor]) {
                visited[neighbor] = true;
                queue.add(neighbor);
            }
        }
    }
    System.out.println();
  }
}

public class BFSDemo {
  public static void main(String[] args) {
    Graph graph = new Graph(6);

    // Add edges to the graph
    graph.addEdge(0, 1);
    graph.addEdge(0, 2);
    graph.addEdge(1, 3);
    graph.addEdge(1, 4);
    graph.addEdge(2, 4);
    graph.addEdge(3, 5);
    graph.addEdge(4, 5);

    // Perform BFS starting from vertex 0
    graph.BFS(0);
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files
Breadth-First Traversal starting from vertex 0:
0 1 2 3 4 5


Process finished with exit code 0
```

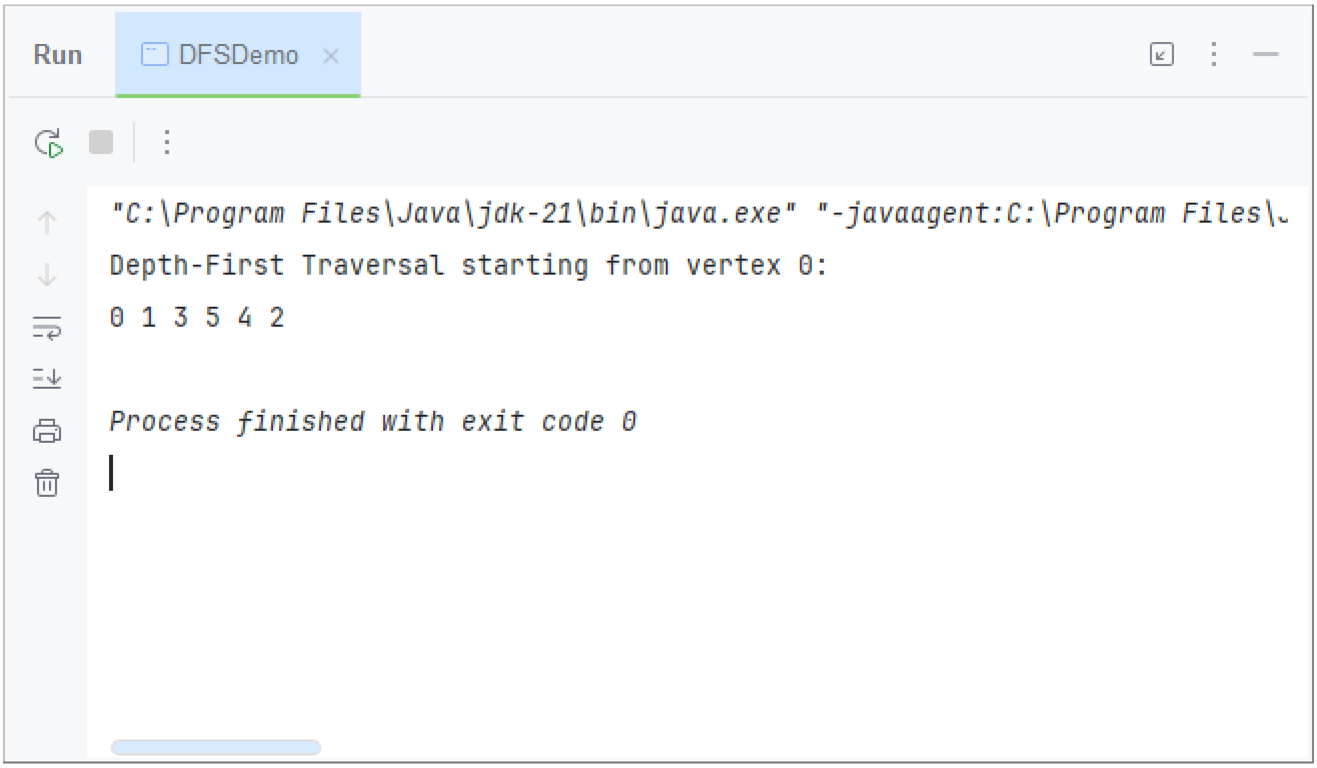
Program 13

A program to find out depth-first traversals of a given graph.

Code

```
 class Graph {

 private int vertices;
 private LinkedList<Integer>[] adjacencyList;

 public Graph(int vertices) {
  this.vertices = vertices;
  adjacencyList = new LinkedList[vertices];
  for (int i = 0; i < vertices; i++) {
   adjacencyList[i] = new LinkedList<>();
  }
 }

 public void addEdge(int source, int destination) {
  adjacencyList[source].add(destination);
 }

 public void DFS(int startVertex) {
  boolean[] visited = new boolean[vertices];
  System.out.println("Depth-First Traversal starting from
vertex " + startVertex + ":");
  dfsHelper(startVertex, visited);
  System.out.println();
 }

 private void dfsHelper(int vertex, boolean[] visited) {
  visited[vertex] = true;
  System.out.print(vertex + " ");

  for (int neighbor : adjacencyList[vertex]) {
   if (!visited[neighbor]) {
    dfsHelper(neighbor, visited);
   }
  }
 }
}

public class DFSDemo {
  public static void main(String[] args) {
    Graph graph = new Graph(6);

    // Add edges to the graph
    graph.addEdge(0, 1);
    graph.addEdge(0, 2);
    graph.addEdge(1, 3);
    graph.addEdge(1, 4);
    graph.addEdge(2, 4);
    graph.addEdge(3, 5);
    graph.addEdge(4, 5);

    // Perform DFS starting from vertex 0
    graph.DFS(0);
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\.
Depth-First Traversal starting from vertex 0:
0 1 3 5 4 2

Process finished with exit code 0
```

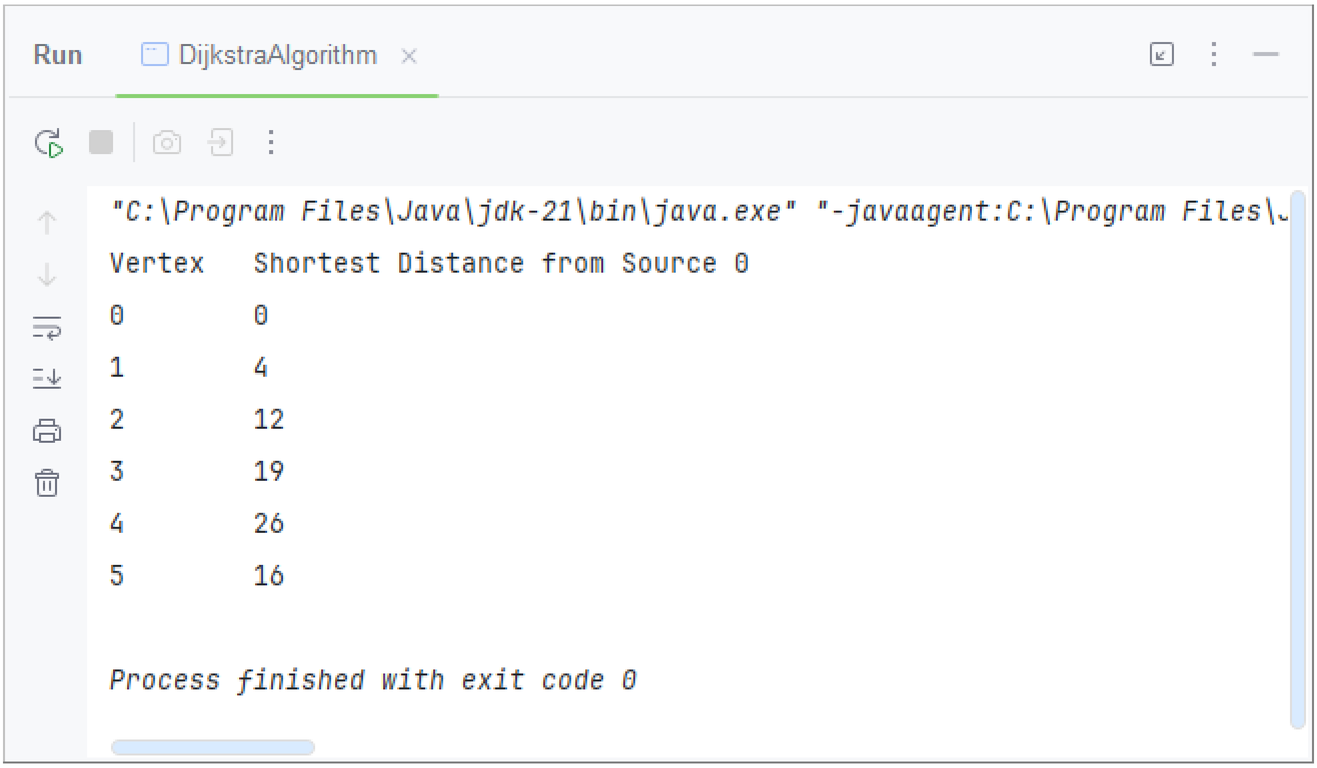
Program 14

A program to find out Dijkstra’s shortest path algorithm using greedy approach.

Code

```
  class DijkstraAlgorithm {

  private static int getMinDistanceVertex(int[] distances,
boolean[] visited, int vertices) {
    int minDistance = Integer.MAX_VALUE;
    int minIndex = -1;

    for (int i = 0; i < vertices; i++) {
      if (!visited[i] && distances[i] < minDistance) {
        minDistance = distances[i];
        minIndex = i;
      }
    }

    return minIndex;
  }

  public static void dijkstra(int[][] graph, int source) {
    int vertices = graph.length;
    int[] distances = new int[vertices];
    boolean[] visited = new boolean[vertices];
    Arrays.fill(distances, Integer.MAX_VALUE);
    distances[source] = 0;

    for (int i = 0; i < vertices - 1; i++) {
      int currentVertex = getMinDistanceVertex(distances,
visited, vertices);
      visited[currentVertex] = true;


      for (int neighbor = 0; neighbor < vertices; neighbor++) {
        if (!visited[neighbor] &&
          graph[currentVertex][neighbor] != 0 &&
          distances[currentVertex] != Integer.MAX_VALUE &&
          distances[currentVertex] + graph[currentVertex]
[neighbor] < distances[neighbor]) {
          distances[neighbor] = distances[currentVertex] +
graph[currentVertex][neighbor];
        }
      }
    }

    // Print shortest distances
    System.out.println("Vertex \t Shortest Distance from Source "
+ source);
    for (int i = 0; i < vertices; i++) {
      System.out.println(i + " \t\t " + distances[i]);
    }
  }

  public static void main(String[] args) {
    int[][] graph = {
      {0, 4, 0, 0, 0, 0},
      {4, 0, 8, 0, 0, 0},
      {0, 8, 0, 7, 0, 4},
      {0, 0, 7, 0, 9, 14},
      {0, 0, 0, 9, 0, 10},
      {0, 0, 4, 14, 10, 0}
    };

    int source = 0; // Starting vertex
    dijkstra(graph, source);
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\J
Vertex   Shortest Distance from Source 0
0        0
1        4
2        12
3        19
4        26
5        16

Process finished with exit code 0
```

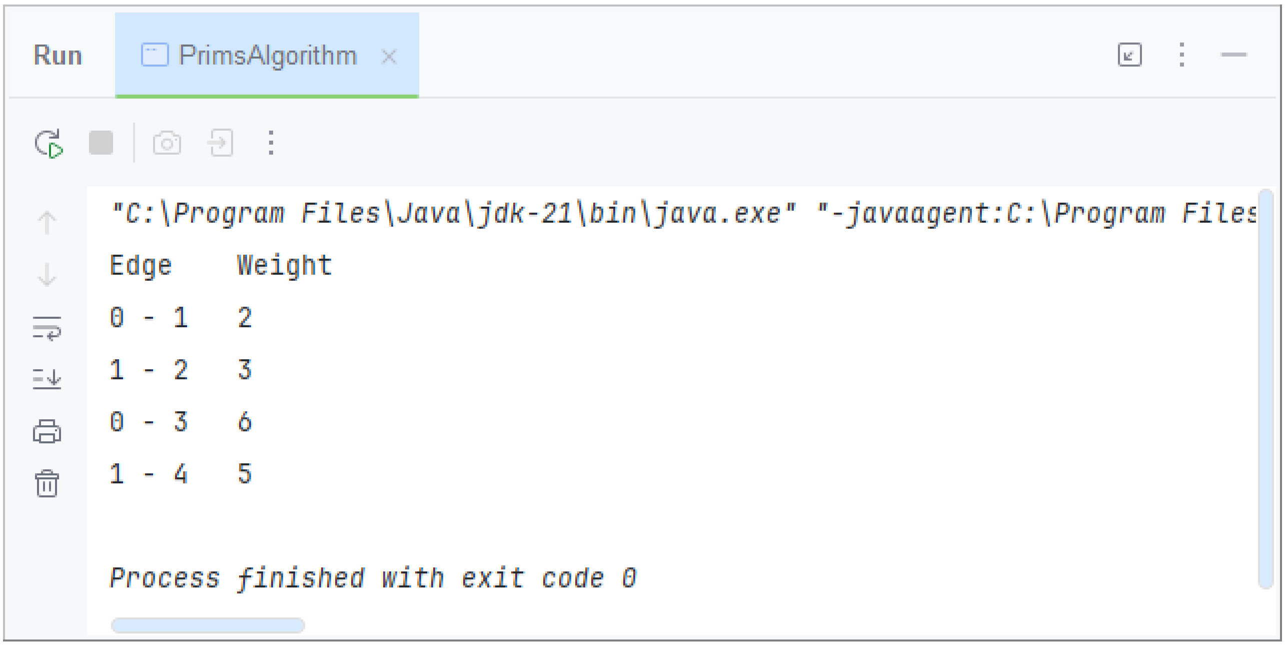
Program 15

A program to find out Prim’s minimum spanning algorithm using greedy approach.

Code

```
 class PrimsAlgorithm {

  private static int getMinKeyVertex(int[] key, boolean[]
included, int vertices) {
    int minValue = Integer.MAX_VALUE;
    int minIndex = -1;

    for (int i = 0; i < vertices; i++) {
      if (!included[i] && key[i] < minValue) {
        minValue = key[i];
        minIndex = i;
      }
    }

    return minIndex;
  }
  public static void primMST(int[][] graph) {
    int vertices = graph.length;
    int[] parent = new int[vertices];
    int[] key = new int[vertices];
    boolean[] included = new boolean[vertices];
    Arrays.fill(key, Integer.MAX_VALUE);
    key[0] = 0;
    parent[0] = -1;

    for (int i = 0; i < vertices - 1; i++) {
      int u = getMinKeyVertex(key, included, vertices);
      included[u] = true;
      for (int v = 0; v < vertices; v++) {
        if (graph[u][v] != 0 && !included[v] && graph[u][v] <
key[v]) {
          parent[v] = u;
          key[v] = graph[u][v];
        }
      }
    }
    printMST(parent, graph);
  }
  private static void printMST(int[] parent, int[][] graph) {
    System.out.println("Edge \tWeight");
    for (int i = 1; i < graph.length; i++) {
      System.out.println(parent[i] + " - " + i + "\t" + graph[i]
[parent[i]]);
    }
  }

  public static void main(String[] args) {
    int[][] graph = {
      {0, 2, 0, 6, 0},
      {2, 0, 3, 8, 5},
      {0, 3, 0, 0, 7},
      {6, 8, 0, 0, 9},
      {0, 5, 7, 9, 0}
    };

    primMST(graph);
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files
Edge    Weight
0 - 1   2
1 - 2   3
0 - 3   6
1 - 4   5

Process finished with exit code 0
```

Program 16

A program to implement min head and max head.

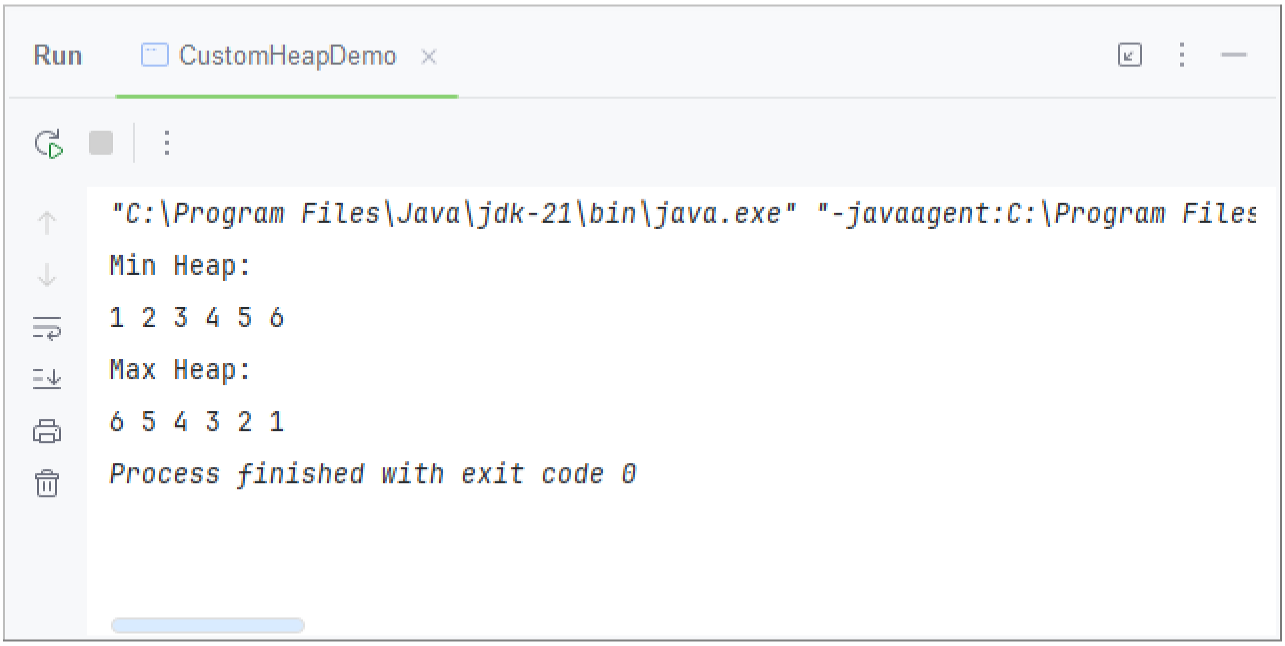
Code

```
  class CustomHeap {

  private List<Integer> heap;
  private boolean isMinHeap;
  public CustomHeap(boolean isMinHeap) {
    this.heap = new ArrayList<>();
    this.isMinHeap = isMinHeap;
  }
  public void add(int value) {
    heap.add(value);
    heapifyUp(heap.size() - 1);
  }
  public int poll() {
    if (heap.isEmpty()) throw new NoSuchElementException("Heap is
empty");
    int top = heap.get(0);
    heap.set(0, heap.get(heap.size() - 1));
    heap.remove(heap.size() - 1);
    heapifyDown(0);
    return top;
  }
  private void heapifyUp(int index) {
    int parentIndex = (index - 1) / 2;
    if (index > 0 && compare(heap.get(index),
heap.get(parentIndex))) {
      Collections.swap(heap, index, parentIndex);
      heapifyUp(parentIndex);
    }
  }
  private void heapifyDown(int index) {
    int leftChild = 2 * index + 1;
    int rightChild = 2 * index + 2;
    int extremum = index;
```

```java
    if (leftChild < heap.size() && compare(heap.get(leftChild),
heap.get(extremum))) {
      extremum = leftChild;
    }
    if (rightChild < heap.size() && compare(heap.get(rightChild),
heap.get(extremum))) {
      extremum = rightChild;
    }

    if (extremum != index) {
      Collections.swap(heap, index, extremum);
      heapifyDown(extremum);
    }
  }
  private boolean compare(int child, int parent) {
    return isMinHeap ? child < parent : child > parent;
  }
  public boolean isEmpty() {
    return heap.isEmpty();
  }
  public void display() {
    System.out.println(heap);
  }
}

public class CustomHeapDemo {
  public static void main(String[] args) {
    // Min Heap
    CustomHeap minHeap = new CustomHeap(true);
    System.out.println("Min Heap:");
    minHeap.add(3);
    minHeap.add(1);
    minHeap.add(6);
    minHeap.add(5);
    minHeap.add(2);
    minHeap.add(4);
    while (!minHeap.isEmpty()) {
      System.out.print(minHeap.poll() + " ");
    }
    System.out.println();

    // Max Heap
    CustomHeap maxHeap = new CustomHeap(false);
```

```
        System.out.println("Max Heap:");
        maxHeap.add(3);
        maxHeap.add(1);
        maxHeap.add(6);
        maxHeap.add(5);
        maxHeap.add(2);
        maxHeap.add(4);
        while (!maxHeap.isEmpty()) {
          System.out.print(maxHeap.poll() + " ");
        }
      }
    }
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files
Min Heap:
1 2 3 4 5 6
Max Heap:
6 5 4 3 2 1
Process finished with exit code 0
```

Program 1

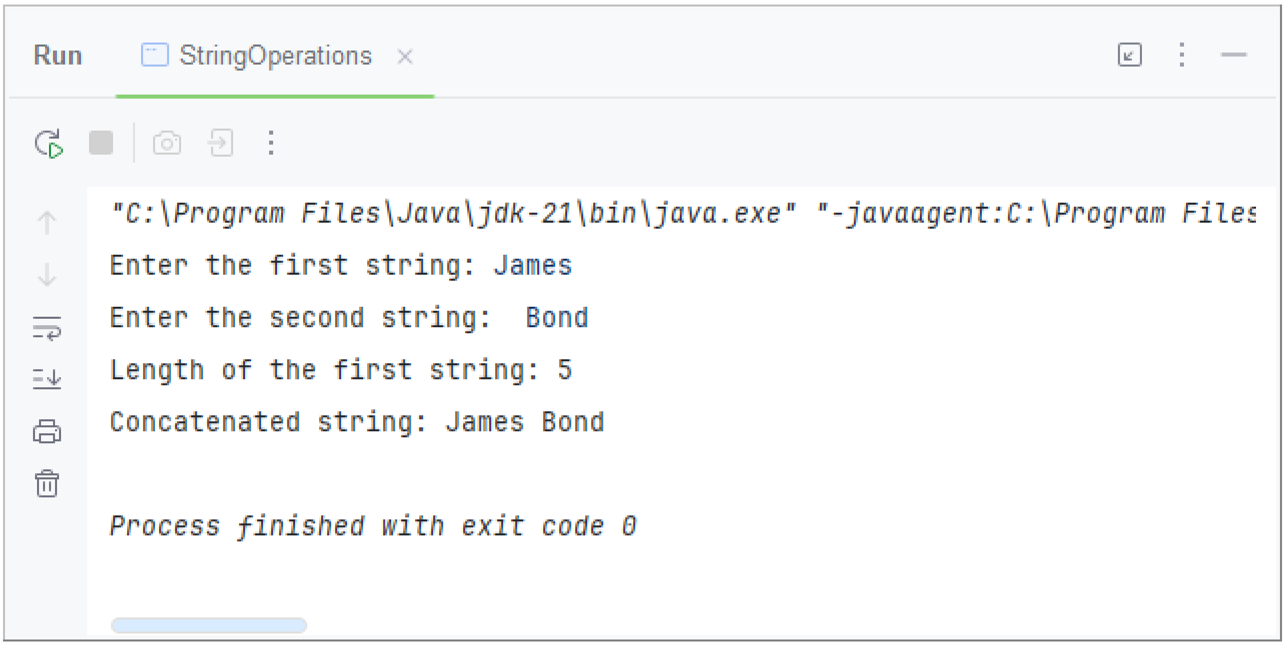
A program to find length of string, concatenation of two strings without using inbuilt function

Code

```
public class StringOperations {

 public static int findLength(String str) {
   int length = 0;
   try {
     while (true) {
       char c = str.charAt(length);
       length++;
     }
   } catch (IndexOutOfBoundsException e) {
   }
   return length;
 }

 public static String concatenateStrings(String str1, String
str2) {
   char[] result = new char[str1.length() + str2.length()];
   int i = 0;
   for (i = 0; i < str1.length(); i++) {
     result[i] = str1.charAt(i);
   }
   for (int j = 0; j < str2.length(); j++) {
     result[str1.length() + j] = str2.charAt(j);
   }
   return new String(result);
 }

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

   System.out.print("Enter the first string: ");
   String str1 = scanner.nextLine();
   System.out.print("Enter the second string: ");
```

```
    String str2 = scanner.nextLine();
    int length = findLength(str1);
    System.out.println("Length of the first string: " + length);
    String concatenatedString = concatenateStrings(str1, str2);
    System.out.println("Concatenated string: " +
concatenatedString);
    scanner.close();
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files
Enter the first string: James
Enter the second string:  Bond
Length of the first string: 5
Concatenated string: James Bond


Process finished with exit code 0
```

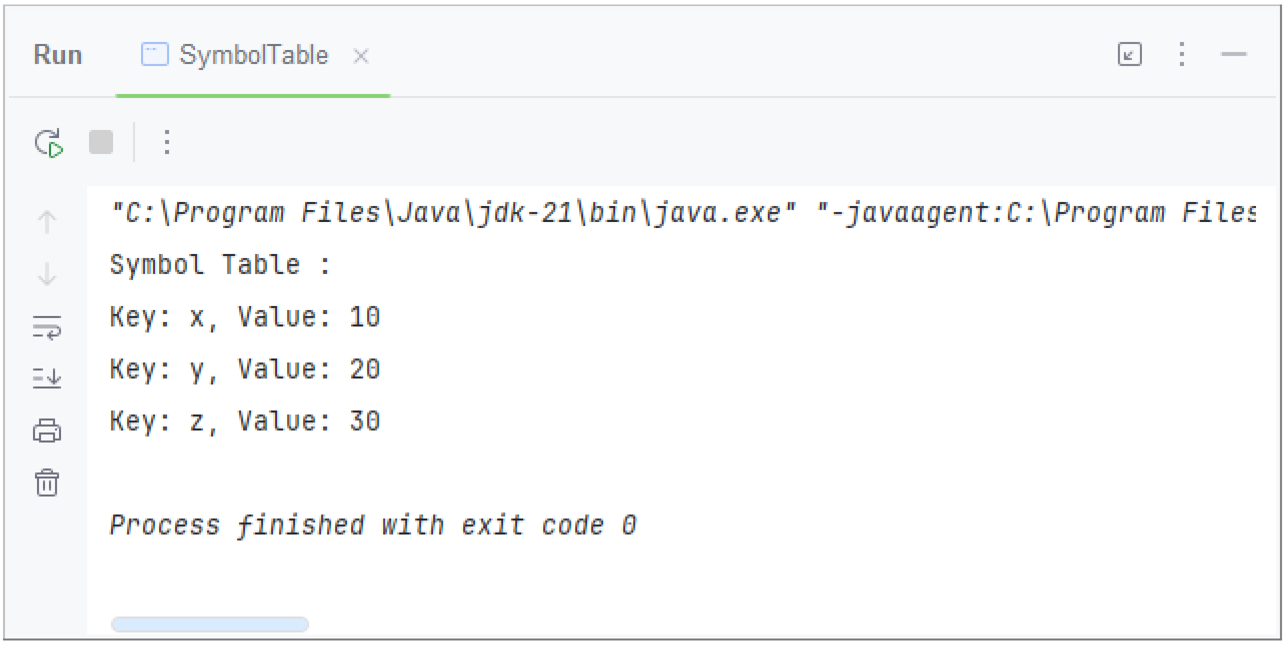
Program 2

A program to print a Symbol Table

Code

```
public class SymbolTable {
  private Map<String, String> table;

  public SymbolTable() {
    table = new HashMap<>();
  }

  public void addSymbol(String key, String value) {
    table.put(key, value);
  }

  public void printSymbolTable() {
    System.out.println("Symbol Table :");
    for (Map.Entry<String, String> entry : table.entrySet()) {
      System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());
    }
  }
  public static void main(String[] args) {
    SymbolTable symbolTable = new SymbolTable();
    symbolTable.addSymbol("x", "10");
    symbolTable.addSymbol("y", "20");
    symbolTable.addSymbol("z", "30");
    symbolTable.printSymbolTable();
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files
Symbol Table :
Key: x, Value: 10
Key: y, Value: 20
Key: z, Value: 30

Process finished with exit code 0
```

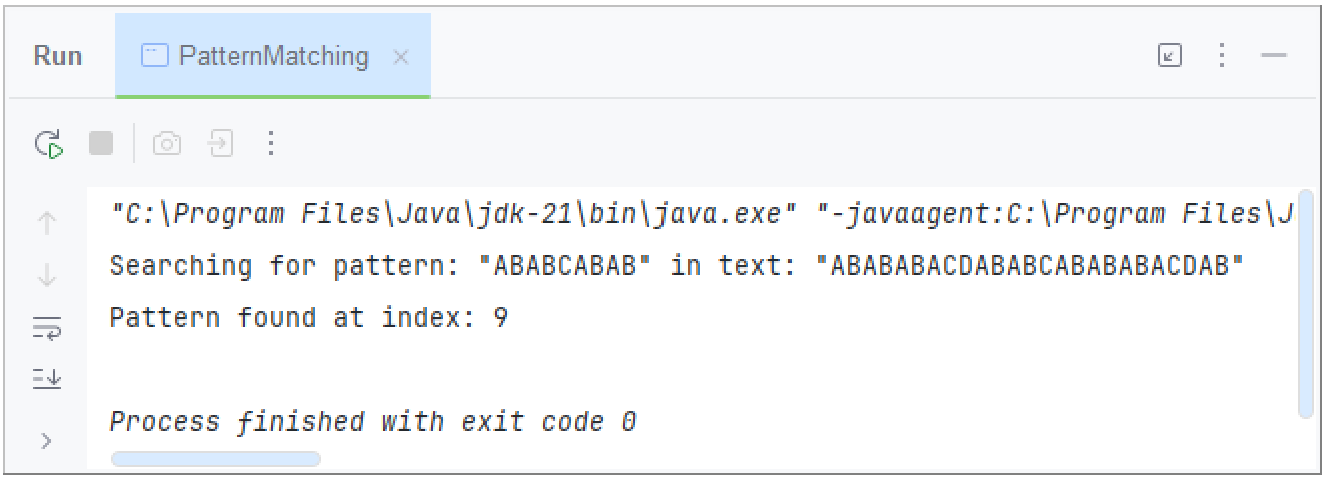
Program 3

A program to implement Pattern Matching

Code

```
 public class PatternMatching {

   public static void naivePatternMatch(String text, String
pattern) {
    int n = text.length();
    int m = pattern.length();
  for (int i = 0; i <= n - m; i++) {
      int j = 0;
  while (j < m && text.charAt(i + j) == pattern.charAt(j)) {
       j++;  }
     if (j == m) {
       System.out.println("Pattern found at index: " + i);
     }
    }
  }
public static void main(String[] args) {
    String text = "ABABABACDABABCABABABACDAB";
    String pattern = "ABABCABAB";
 System.out.println("Searching for pattern: \"" + pattern + "\"
in text: \"" + text + "\"");
 naivePatternMatch(text, pattern);
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\J
Searching for pattern: "ABABCABAB" in text: "ABABABACDABABCABABABACDAB"
Pattern found at index: 9


Process finished with exit code 0
```

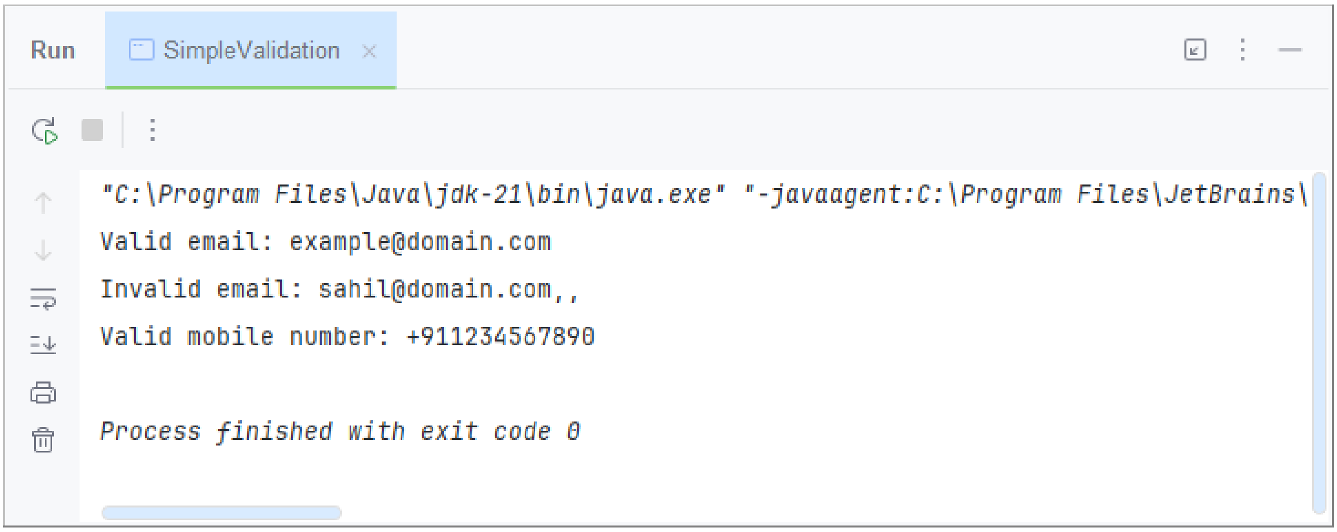
Program 4

A program to validate an email ID/Mobile Number

Code

```java
public class SimpleValidation {

  public static boolean validateEmail(String email) {
    String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\.[a-zA-Z0-9_+&*-]+)*@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$";
    return email.matches(emailRegex);
  }
  public static boolean validateMobile(String mobile) {
    String mobileRegex = "^(\\+\\d{1,3})?\\d{10}$";
    return mobile.matches(mobileRegex);
  }
  public static void main(String[] args) {
    String email = "example@domain.com";
    String inemail = "sahil@domain.com,,";
    String mobile = "+911234567890";
    if (validateEmail(email)) {
      System.out.println("Valid email: " + email);
    } else {
      System.out.println("Invalid email: " + email);
    }
    if (validateEmail(inemail)) {
      System.out.println("Valid email: " + inemail);
    } else {
      System.out.println("Invalid email: " + inemail);
    }
    if (validateMobile(mobile)) {
      System.out.println("Valid mobile number: " + mobile);
    } else {
      System.out.println("Invalid mobile number: " + mobile);
    }
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\
Valid email: example@domain.com
Invalid email: sahil@domain.com,,
Valid mobile number: +911234567890

Process finished with exit code 0
```

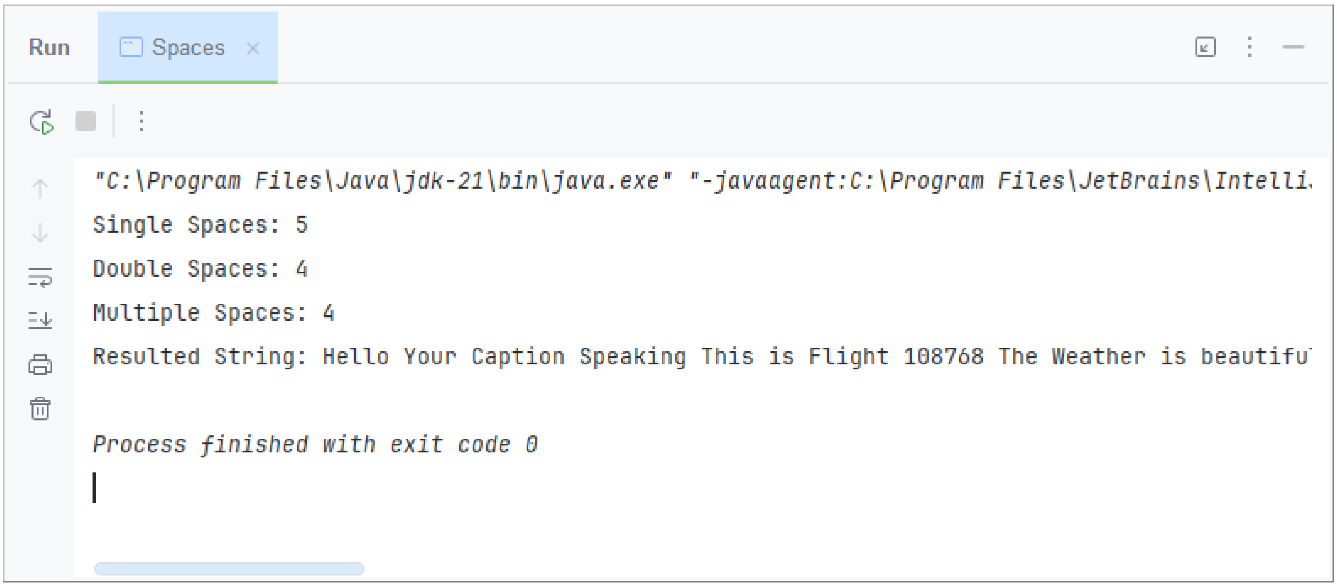
Program 5

A program to check number of single spaces, double spaces and multiple spaces in a given string. Print the resultant string after replacing double/multiple spaces with single space.

Code

```
 public class Spaces {
    public static void main(String[] args) {
        String input = "Hello Your Caption   Speaking 
This  is  Flight   108768 The Weather   is  beautiful 
Today ";
        int singleSpaceCount = 0;
        int doubleSpaceCount = 0;
        int multiSpaceCount = 0;

        for (int i = 0; i < input.length(); i++) {
            if (input.charAt(i) == ' ') {
                int spaceCount = 1;

                while (i + 1 < input.length() && 
input.charAt(i + 1) == ' ') {
                    spaceCount++;
                    i++;
                }
                if (spaceCount == 1) {
                    singleSpaceCount++;
                } else if (spaceCount == 2) {
                    doubleSpaceCount++;
                } else if (spaceCount > 2) {
                    multiSpaceCount++;
                }
            }
        }
        String result = input.replaceAll("\\s{2,}", "
").trim();
        System.out.println("Single Spaces: " +
singleSpaceCount);
        System.out.println("Double Spaces: " +
doubleSpaceCount);
        System.out.println("Multiple Spaces: " +
multiSpaceCount);
        System.out.println("Resulted String: " + result);
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intelli
Single Spaces: 5
Double Spaces: 4
Multiple Spaces: 4
Resulted String: Hello Your Caption Speaking This is Flight 108768 The Weather is beautifu

Process finished with exit code 0
```

Compiler Construction

S no	Program Description	Remarks
1	Program to find length of string, concatenation of two strings without using inbuilt function	
2	Program to print a Symbol Table	
3	Program to implement Pattern Matching	
4	Program to validate an email ID/Mobile Number.	
5	Program to check number of single spaces, double spaces and multiple spaces in a given string. Print the resultant string after replacing double/multiple spaces with single space.	

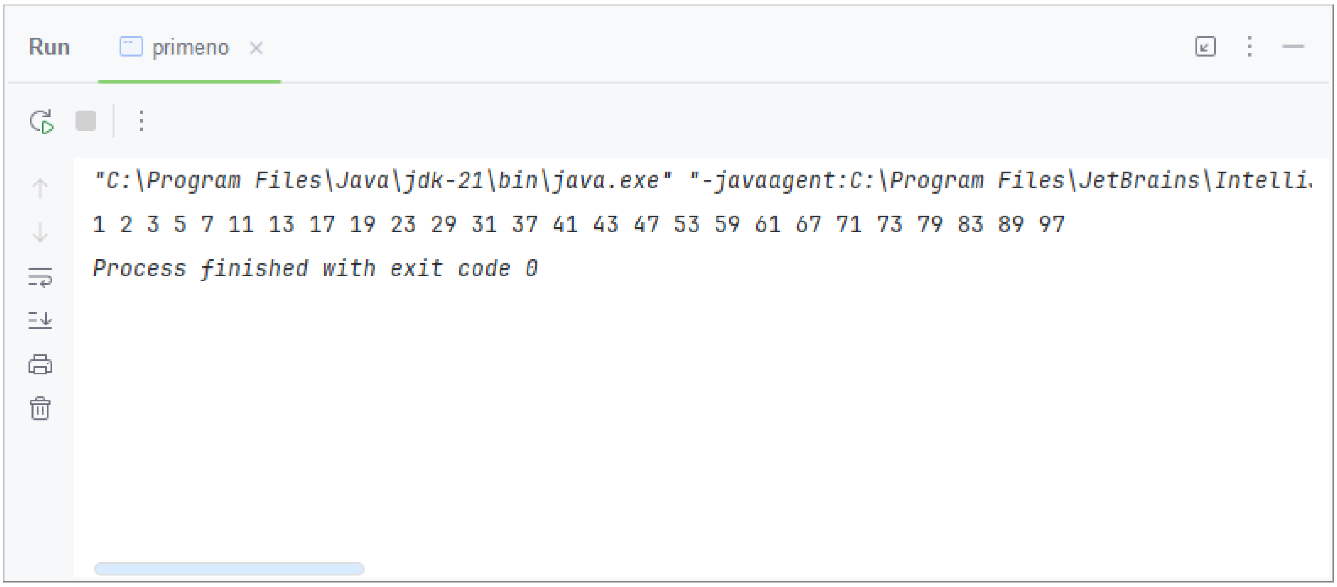
Program 1

Program in JAVA to implement print Prime Numbers up to 100.

Code

```java
public class primeno {

  public static void firsthundredprimeno() {
    Set<Integer> set = new HashSet<>();
    int i = 1;
    int k = 1;
    System.out.print(k + " ");
    while (k != 100) {
      k++;
      boolean m = false;
      for (int j : set) {
        if (k % j == 0) {
          m = true;
          break;
        }
      }
      if (m) {
        continue;
      }
      set.add(k);
      System.out.print(k + " ");
      i++;

    }
  }

  public static void main(String[] args) {
    firsthundredprimeno();
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intelli
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
Process finished with exit code 0
```

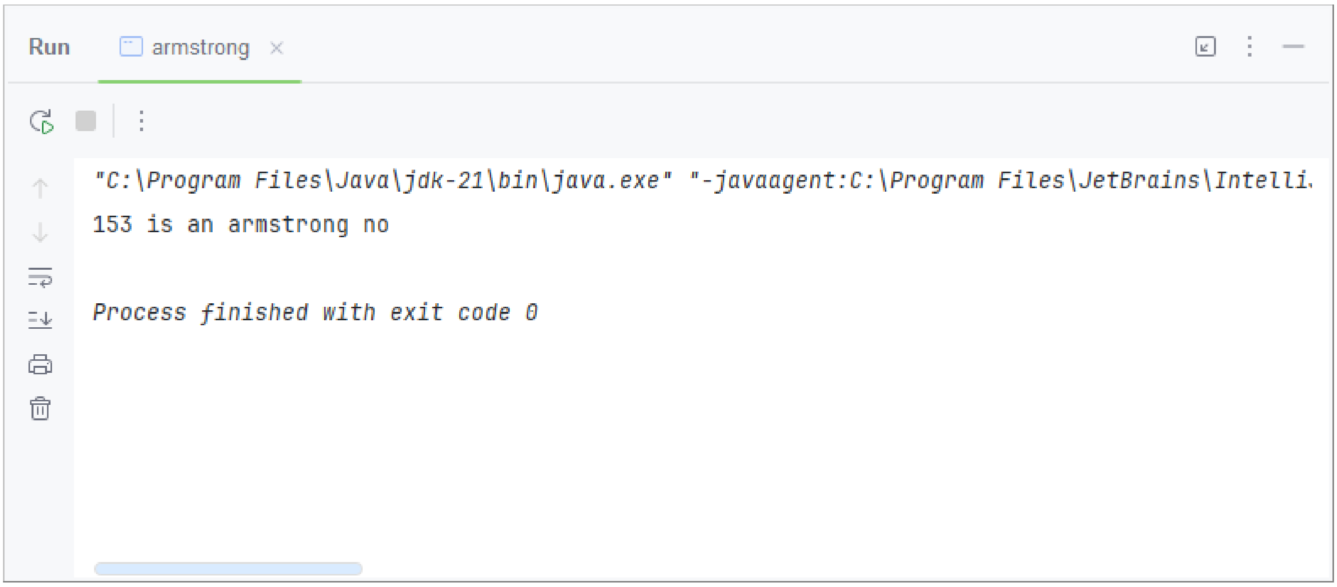
Program 2

Program in JAVA to implement finding whether a number is Armstrong or not.

Code

```java
public class armstrong {
  public static boolean armstrongornot(int n){
    int pow = (int)Math.log10(n)+1;
    int z=n;
    int ans =0;
    while(z!=0){
      int k = 1;
      for (int i = 0; i <pow ; i++) {
        k*=z%10;
      }
      ans+=k;
      z=z/10;
    }
    return n==ans;
  }

  public static void main(String[] args) {

    int n = 153;
    if(armstrongornot(n)){
      System.out.println(n+" is an armstrong no ");
    }else{
      System.out.println(n+" is not an armstrong no ");
    }
  }
}


}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intelli
153 is an armstrong no

Process finished with exit code 0
```

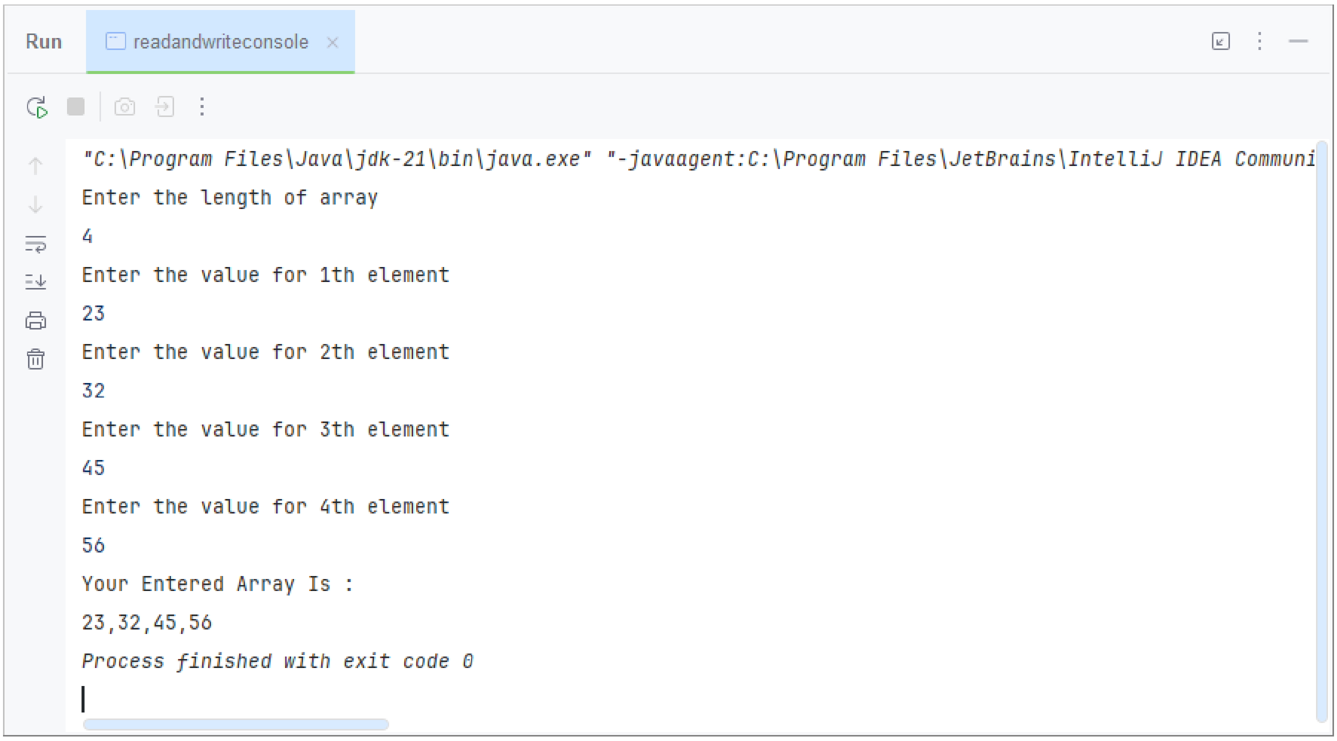
Program 3

Program in Java to implement array reading and writing on/from console.

Code

```
 public class readandwriteconsole {

 public static void main(String[] args) {
  Scanner r =  new Scanner(System.in);
  System.out.println("Enter the length of array ");
  int n = r.nextInt();
  int[] nums = new int[n];
  for (int i = 0; i <n ; i++) {
   System.out.println("Enter the value for " +(i+1)+ "th
element");
   int k = r.nextInt();
   nums[i]=k;
  }
  System.out.println("Your Entered Array Is :");
  for (int i = 0; i <n ; i++) {
   if(i==n-1){System.out.print(nums[i]);}
   else{
    System.out.print(nums[i]+",");
   }
  }
 }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Enter the length of array
4
Enter the value for 1th element
23
Enter the value for 2th element
32
Enter the value for 3th element
45
Enter the value for 4th element
56
Your Entered Array Is :
23,32,45,56
Process finished with exit code 0
```

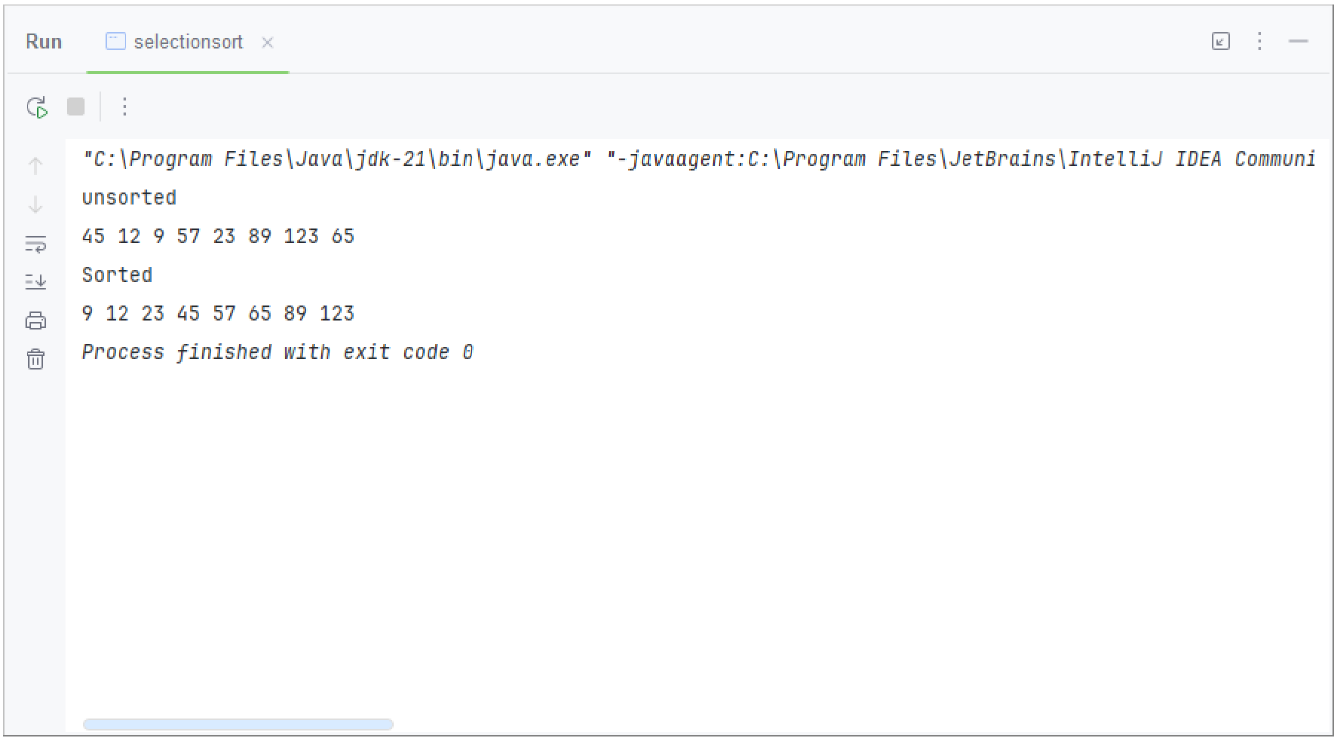
Program 4

Program in Java to implement selection sort using functions

Code

```
 public class selectionsort {

  public static void selection(int[] nums){
    int n = nums.length;
    for (int i = 0; i <n; i++) {
      for (int j = i+1; j <n ; j++) {
        if(nums[i]>nums[j]){
          int temp = nums[i];
          nums[i]=nums[j];
          nums[j]=temp;
        }
      }
    }
  }

  public static void main(String[] args) {
    int[] nums = new int[]{45,12,9,57,23,89,123,65};
    System.out.println("unsorted");
    for(int l:nums){
      System.out.print(l+" ");
    }
    selection(nums);
    System.out.println();
    System.out.println("Sorted");
    for(int l:nums){
      System.out.print(l+" ");
    }
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
unsorted
45 12 9 57 23 89 123 65
Sorted
9 12 23 45 57 65 89 123
Process finished with exit code 0
```

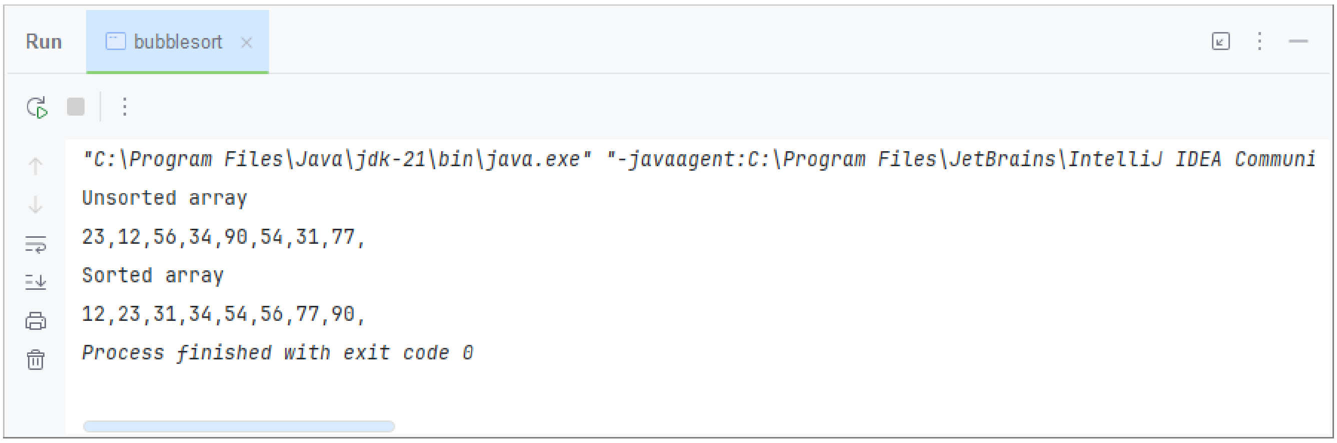
Program 5

Program in Java to implement Bubble Sort using Functions

Code

```
public static void bubble(int[] nums) {
  if (nums.length < 2) {
    return;
  }
  for (int i = nums.length - 1; i >= 0; i--) {
    for (int j = 0; j < i; j++) {
      if (nums[j + 1] < nums[j]) {
        int temp = nums[j];
        nums[j] = nums[j + 1];
        nums[j + 1] = temp;
      }
    }
  }
}

public static void main(String[] args) {
  int[] nums = new int[]{23, 12, 56, 34, 90, 54, 31, 77};
  System.out.println("Unsorted array");
  for (int n : nums) {
    System.out.print(n + ",");
  }
  System.out.println();
  bubble(nums);
  System.out.println("Sorted array");
  for (int n : nums) {
    System.out.print(n + ",");
  }
}}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Unsorted array
23,12,56,34,90,54,31,77,
Sorted array
12,23,31,34,54,56,77,90,
Process finished with exit code 0
```

Program 6

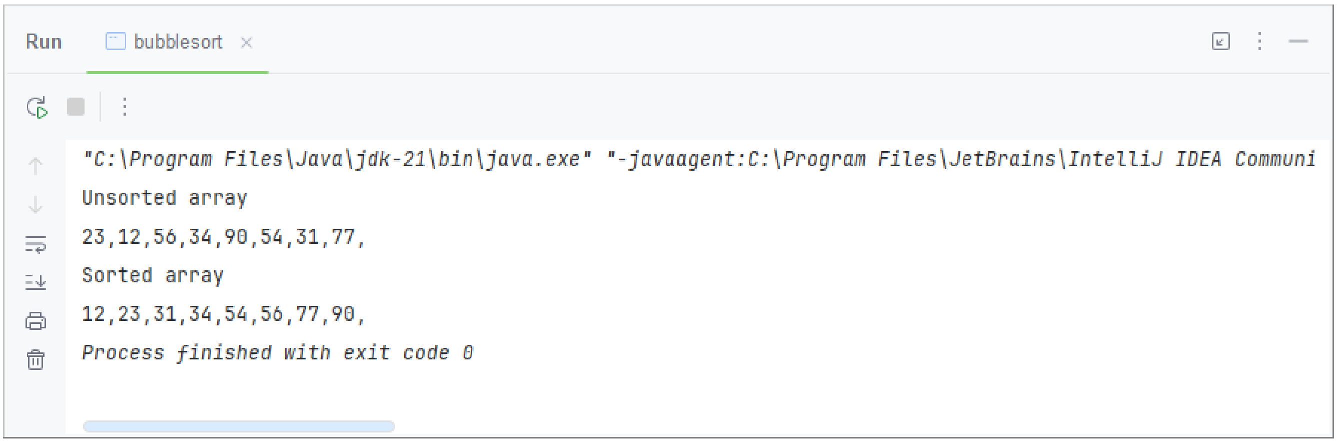
Program in Java to implement Insertion sort using functions.

Code

```
public class insertionsort {

  public static void inserted(int[] nums){

   for (int i = 1; i < nums.length ; i++) {
     for (int j = i-1; j >=0 ; j--) {
       if (nums[j+1]<nums[j]){
         int temp = nums[j+1];
         nums[j+1]=nums[j];
         nums[j]=temp;
       }else{break;}
     }
   }
 }

 public static void main(String[] args) {
   int[] nums = new int[]{23, 12, 56, 34, 90, 54, 31, 77};
   System.out.println("Unsorted array");
   for (int n : nums) {
     System.out.print(n + ",");
```

```
    }
    System.out.println();
    inserted(nums);
    System.out.println("Sorted array");
    for (int n : nums) {
      System.out.print(n + ",");
    }
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Unsorted array
23,12,56,34,90,54,31,77,
Sorted array
12,23,31,34,54,56,77,90,
Process finished with exit code 0
```

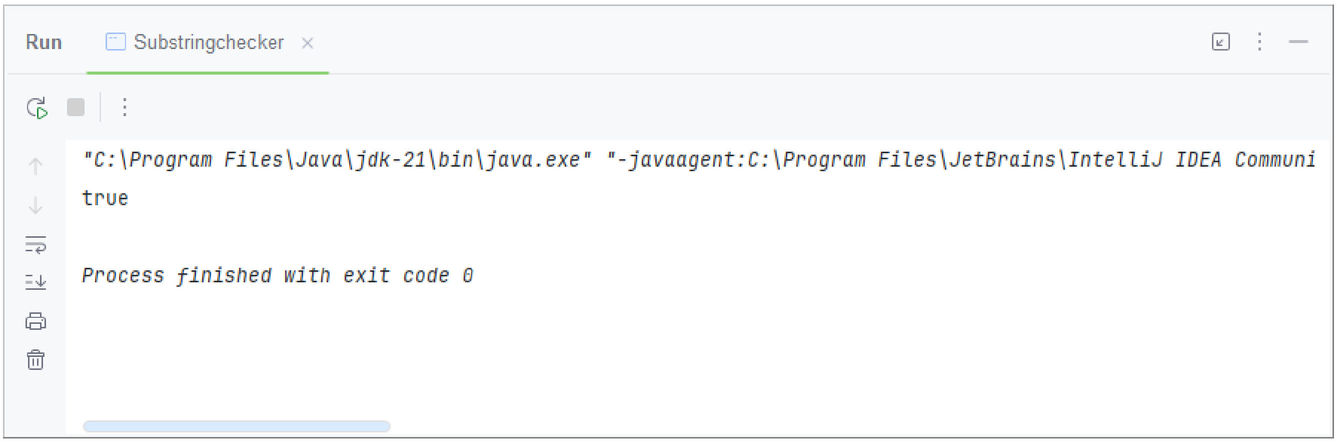
Program 7

Program in Java to implement Matching a Particular Substring in a String without using inbuilt matching function(s).

Code

```java
 public class Substringchecker {

  public static boolean substring(String a ,String b){
    if(a.equals(" ")&&b.equals(" ")){return true;}
    if(a.equals(" ")||b.equals(" ")){return false;}
    int b1=0;
    for (int i = 0; i <a.length()&&b1<b.length(); i++) {
      if (a.charAt(i)==b.charAt(b1)){b1++;}
      else{b1=0;}
    }
    return b1==b.length();
  }
  public static void main(String[] args) {
    System.out.println(substring("ABCDDDEFCKLOCK","LOCK"));
  }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
true

Process finished with exit code 0
```

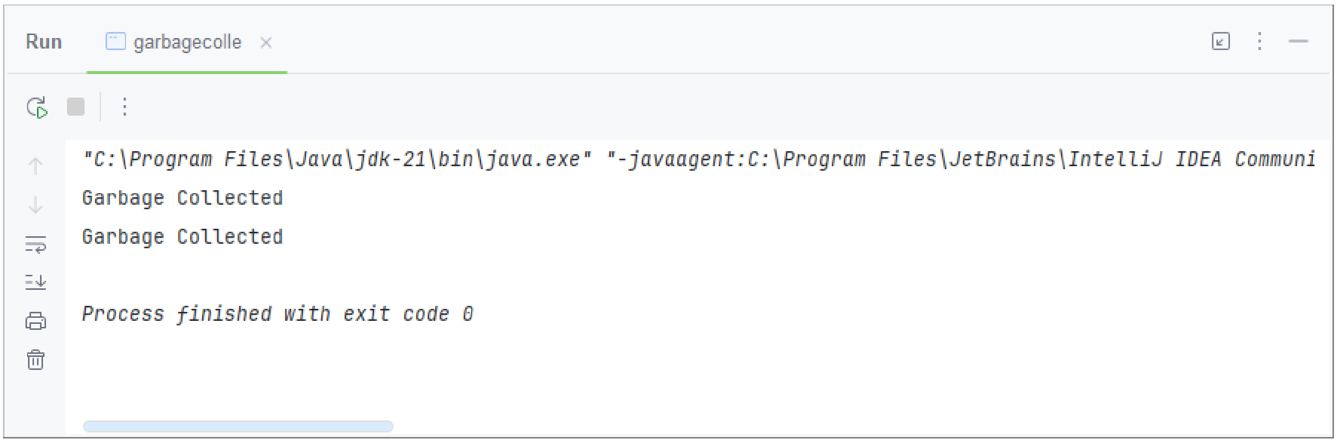
Program 8

Program in Java to implement Garbage Collection Usage in Java.

Code

```
public class garbagecolle {
    @Override
    protected void finalize() throws Throwable {
        super.finalize();
        System.out.println("Garbage Collected");
    }

    public static void main(String[] args) {
        garbagecolle g1 = new garbagecolle();
        garbagecolle g2 = new garbagecolle();
        g1=g2=null;
        System.gc();

    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Garbage Collected
Garbage Collected

Process finished with exit code 0
```

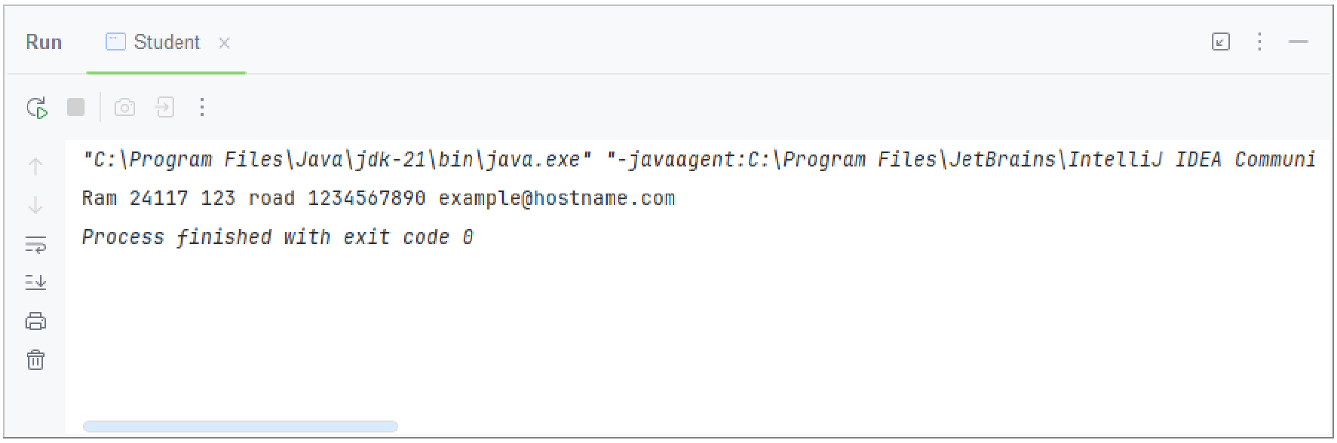
Program 9

Program in Java to implement array of objects and create a student record with details of name, address, contact number and email ID

Code

```
public class Student {
        int rollno;
        String name;
        String address;
        String phno;
        String emailid;
        Student(int a , String b , String c, String d,
String e){
            this.rollno=a;
            this.name=b;
            this.address =c;
            this.phno=d;
            this.emailid=e;
        }
        public static void main(String[] args) {
         Student[] array = new Student[1];
         Student a = new Student(24117,"Ram","123
road","1234567890","example@hostname.com");
         array[0]=a;
            for (Student z :array){
                System.out.print(z.name+" "+z.rollno+"
"+z.address+" "+z.phno+" "+z.emailid);
            }
        }}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
Ram 24117 123 road 1234567890 example@hostname.com
Process finished with exit code 0
```

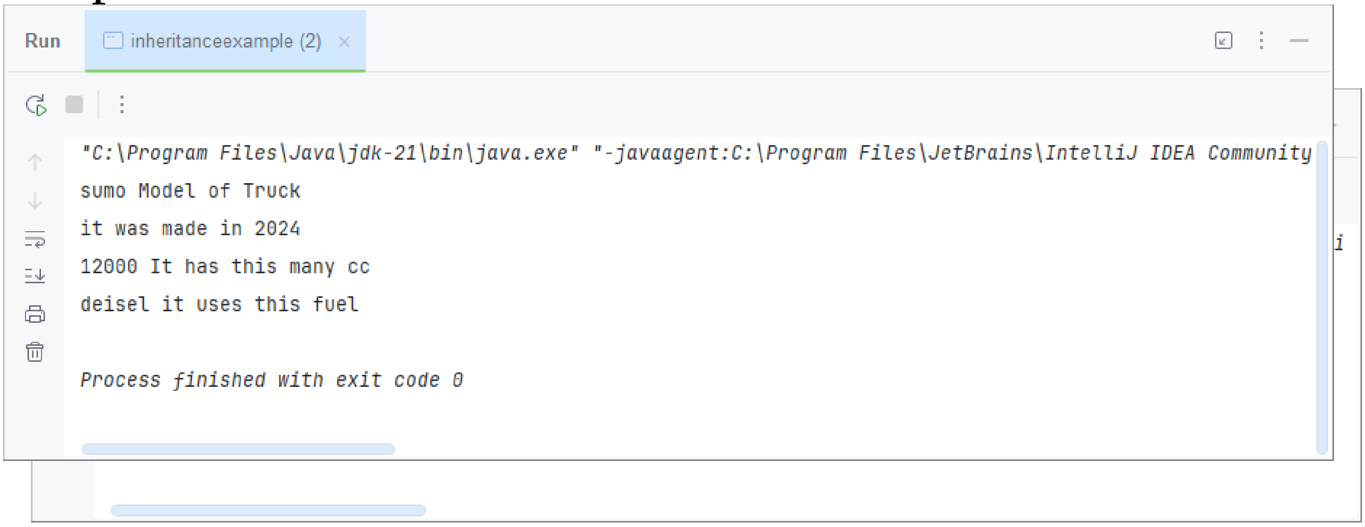
Program 10

Program in Java to implement Single & Multi-level inheritance.

Code

```
class Engine{
    int cc;
    String fueltype;
    Engine(int a,String b){
        this.cc=a;
        this.fueltype=b;
    }
}
class Car extends Engine{
    int yearofmfg;
    Car(int a,int b,String c){
        super(b,c);
        this.yearofmfg=a;
    }

}
class bike extends Engine{
    int model;
    bike(int a,int b,String c){
        super(b,c);
        this.model=a;
    }
}
class Truck extends Car{
    String model;
    Truck(String a,int b,int c,String d ){
        super(b,c,d);
        this.model=a;
    }
}
public class inheritanceexample {
    public static void main(String[] args) {
        Truck chotahathi = new
Truck("sumo",2024,12000,"deisel");
        System.out.println(chotahathi.model+" Model of Truck");
        System.out.println("it was made in
"+chotahathi.yearofmfg);
        System.out.println(chotahathi.cc+" It has this many
cc");
        System.out.println(chotahathi.fueltype+" it uses this
fuel");
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
sumo Model of Truck
it was made in 2024
12000 It has this many cc
deisel it uses this fuel

Process finished with exit code 0
```

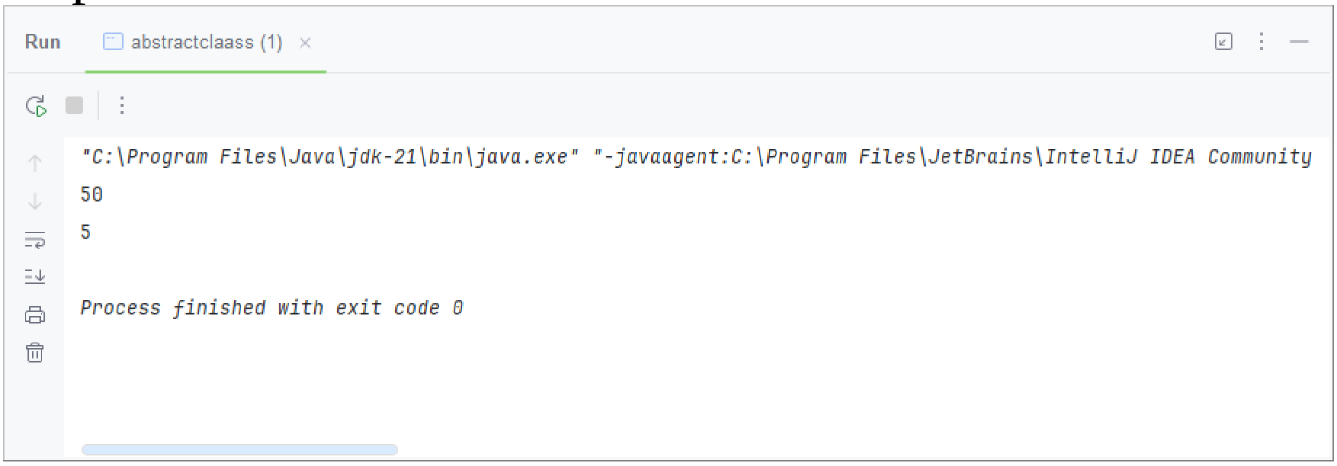
Program 11

Program in Java to implement Abstract Class Usage

Code

```
public class abstractclaass extends test{
    @Override
    public int sum(int a, int b) {
        return (a+b)*10;
    }

    public static void main(String[] args) {
        abstractclaass a = new abstractclaass();
        System.out.println(a.sum(2,3));
        System.out.println(a.sum(2,3,0));
    }
}
class test2 extends test{
    @Override
    public int sum(int a, int b) {
        return Math.abs(a-b);
    }
}
abstract class test{
    int a ;
    int b ;
    abstract public int sum(int a ,int b);
    public int sum(int a,int b,int c){
        return a+b+c;
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
50
5

Process finished with exit code 0
```

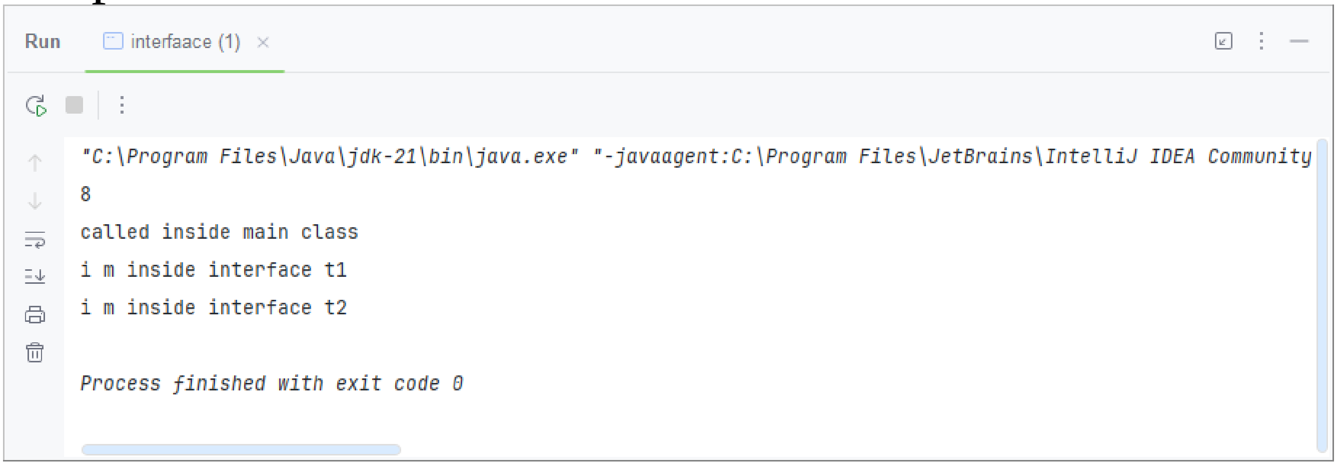
Program 12

Program in Java to implement Interface Usage to implement multiple inheritance.

Code

```
 interface t1{
     int sd =8;
    static void call(){
         System.out.println("i m inside interface t1");
    }
     void called();
}
interface t2{
     static void call(){
        System.out.println("i m inside interface t2");
    }
}
class t3{
    public  void call(){
        System.out.println("i m inside class t3");
    }
}
public class interfaace extends t3 implements t1,t2 {
    public static void main(String[] args) {
        interfaace  t = new interfaace();
        System.out.println(sd);
        t.called();
        t1.call();
        t2.call();
    }

    @Override
    public void called() {
        System.out.println("called inside main class");
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
8
called inside main class
i m inside interface t1
i m inside interface t2

Process finished with exit code 0
```

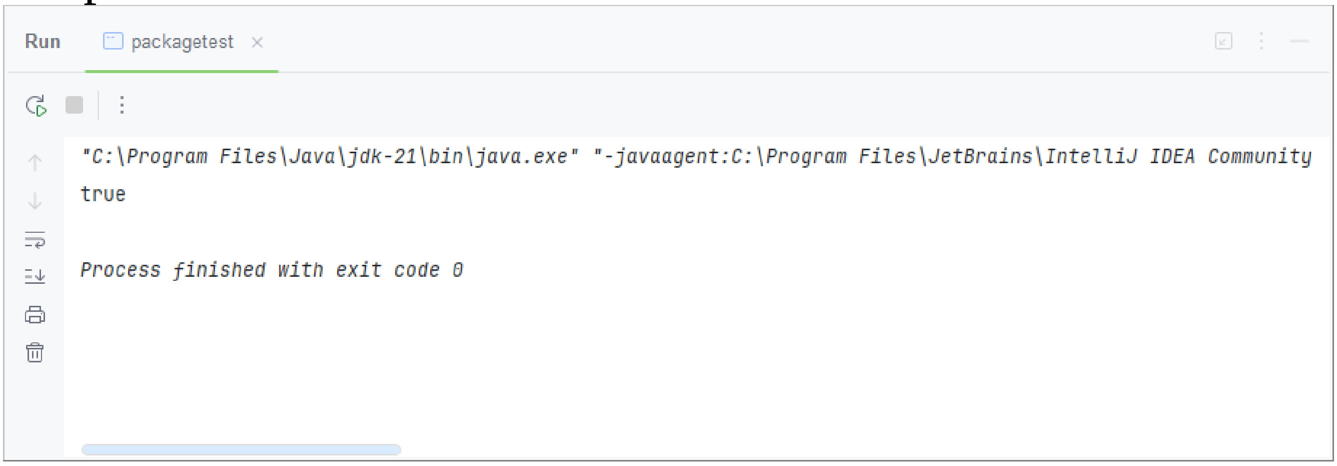
Program 13

Program in Java to implement Packages usage to use one function of a class in another.

Code

```
package labreport;
import leetcode.onethreefive;
public class packagetest {
    public static void main(String[] args) {

System.out.println(onethreefive.isPalindrome("naman"));
    }
}



public class onethreefive {

    public static boolean isPalindrome(String s) {
        //  48-57
        // 65-90
        // 97-122
        int left =0;
        int right = s.length()-1;
        while(right>left){
            int r =(int)s.charAt(right);
            int l =(int)s.charAt(left);
            if(l==0){left++;}
            if(r==0){right--;}
            if(l>=65&&l<=90){l=l+32;}
            if(r>=65&&r<=90){r=r+32;}
            if(!((l>=48&&l<=57)||(l>=97&&l<=122))){left+
+;continue;}
            if(!((r>=48&&r<=57)||(r>=97&&r<=122)))
{right--;continue;}
            if(r!=l){return false;}
            else{left++;right--;}
        }
        return true;
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
true

Process finished with exit code 0
```

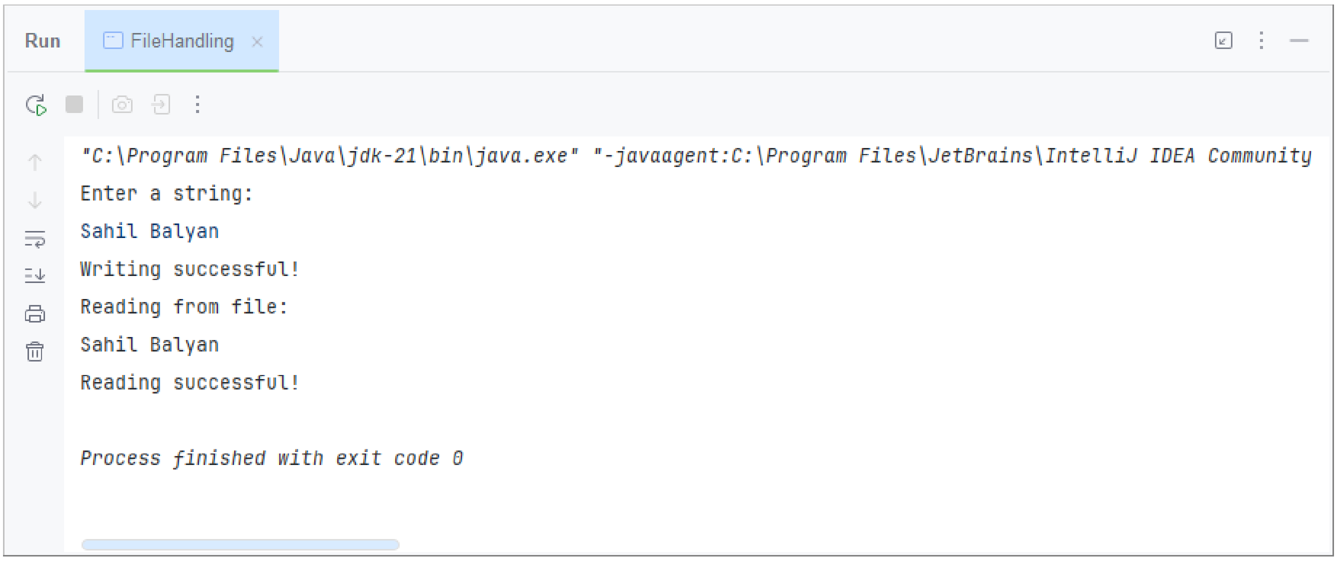
Program 14

Program in Java to implement I/O and file handling

Code

```
public class FileHandling {
    public static void main(String[] args) throws
IOException {
        FileWriter fw = new FileWriter("sahil.txt");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String str = sc.nextLine();
        for (int i = 0; i < str.length(); i++) {
            fw.write(str.charAt(i));
        }
        System.out.println("Writing successful!");
        fw.close();
        FileReader fr = new FileReader("sahil.txt");
        int i;
        System.out.println("Reading from file:");
        while ((i = fr.read()) != -1) {
            System.out.print((char) i);
        }
        System.out.println("\nReading successful!");
        fr.close();
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Enter a string:
Sahil Balyan
Writing successful!
Reading from file:
Sahil Balyan
Reading successful!

Process finished with exit code 0
```

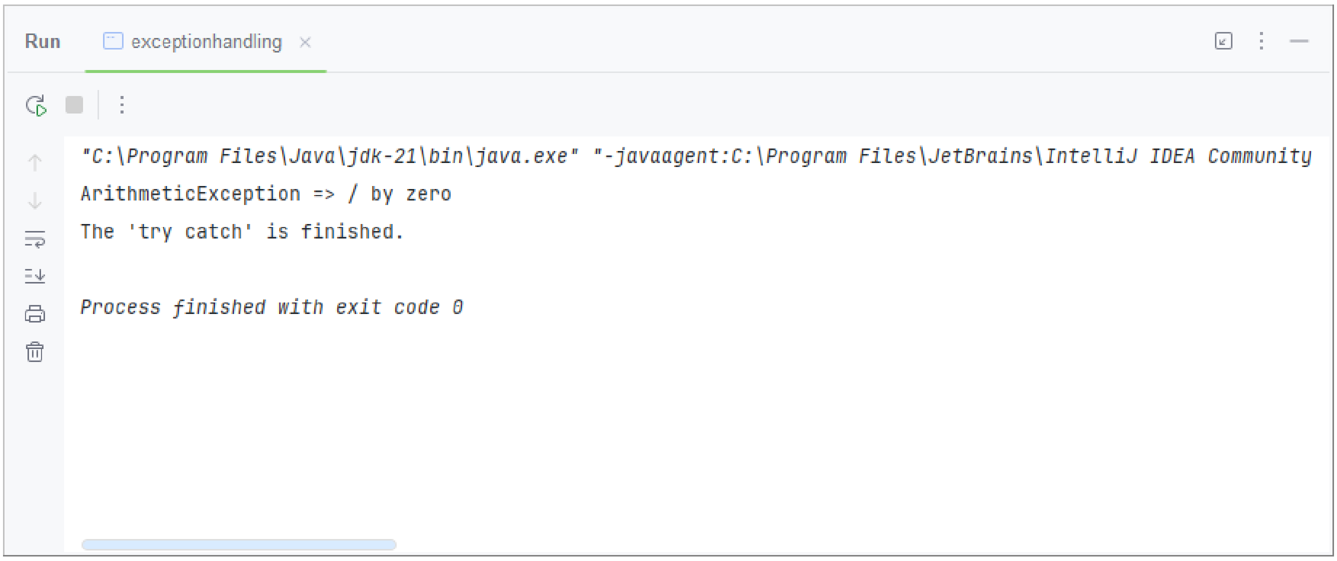
Program 15

Program in Java to implement Exceptions.

Code

```
  class exceptionhandling {
    public static void main(String[] args) {

        try {

            // code that generate exception
            int divideByZero = 5 / 0;
            System.out.println("Rest of code in try 
block");
        } catch (ArithmeticException e) {
            System.out.println("ArithmeticException => " + 
e.getMessage());
        } finally {
            System.out.println("The 'try catch' is 
finished.");
        }
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
ArithmeticException => / by zero
The 'try catch' is finished.

Process finished with exit code 0
```

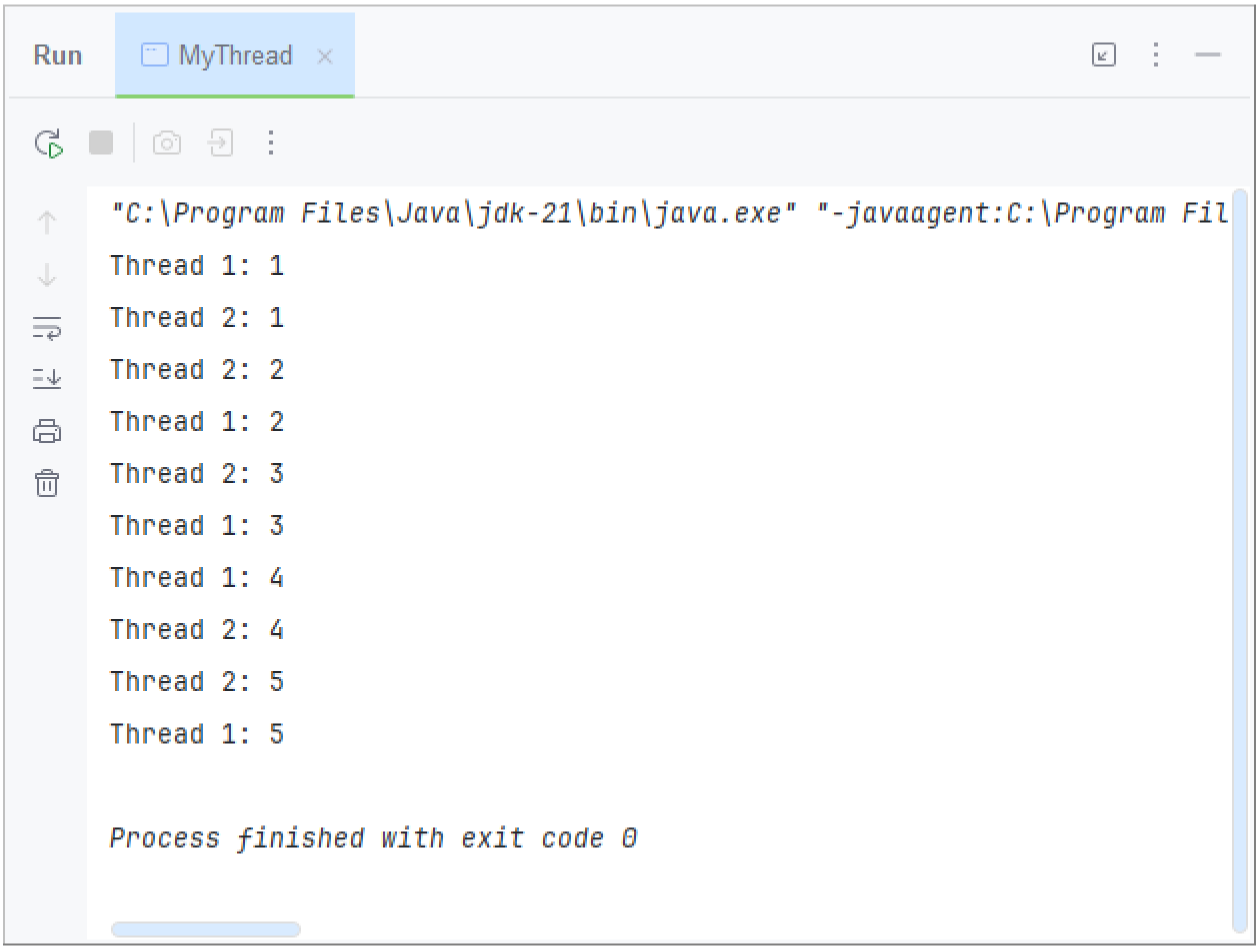
Program 16

Program in Java to implement Multiple Threads

Code

```
 class MyThread extends Thread {
    private String threadName;
   public MyThread(String name)
{ threadName = name; }
 public void run() {
        try {  for (int i = 1; i <= 5; i++) {
                  System.out.println(threadName + ": " +
i);
                  Thread.sleep(500);
            }
        }      catch (InterruptedException e)
            {System.out.println(threadName + "
interrupted."); }
    }
 public static void main(String[] args) {
        MyThread thread1 = new MyThread("Thread 1");
        MyThread thread2 = new MyThread("Thread 2");
        thread1.start();
        thread2.start();
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Fil
Thread 1: 1
Thread 2: 1
Thread 2: 2
Thread 1: 2
Thread 2: 3
Thread 1: 3
Thread 1: 4
Thread 2: 4
Thread 2: 5
Thread 1: 5

Process finished with exit code 0
```

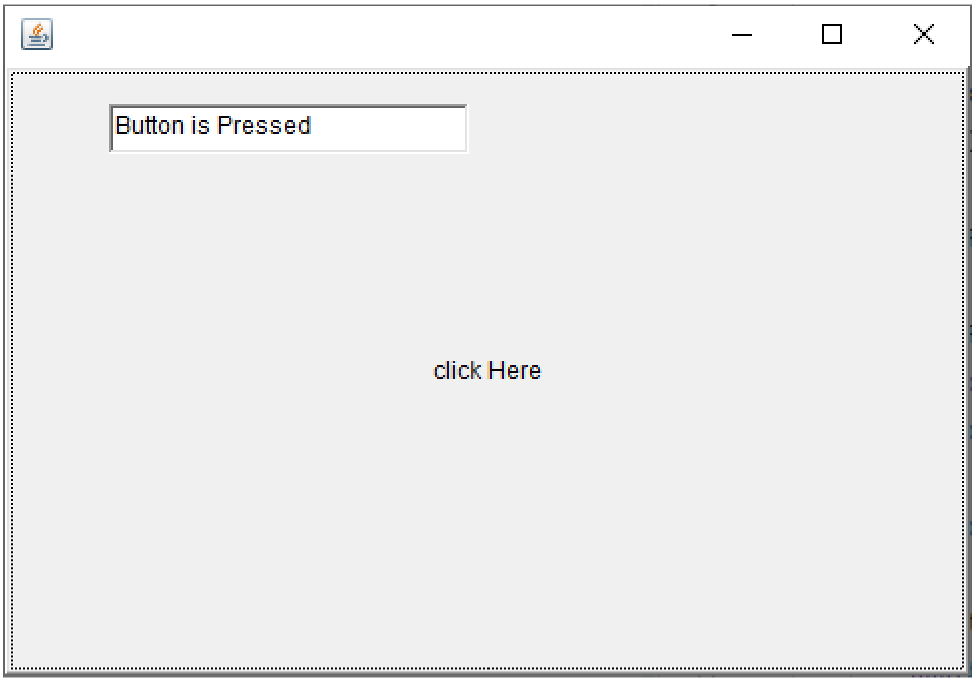
Program 17

Program in Java to implement Event Handlers

Code

```
 import java.awt.*;
import java.awt.event.*;

class AEvent extends Frame implements ActionListener {
TextField textField;
     AEvent()
    {
        textField = new TextField();

        textField.setBounds(60, 50, 180, 25);
        Button button = new Button("click Here");
        button.setBounds(100, 120, 80, 30);

        button.addActionListener(this);

        add(textField);
        add(button);
        setVisible(true);

    }

    public void actionPerformed(ActionEvent e)
    {
        textField.setText("Button is Pressed ");
    }

    public static void main(String[] args)
    {
        AEvent s = new AEvent();
    }
}
```

Output



Button is Pressed

Program 18

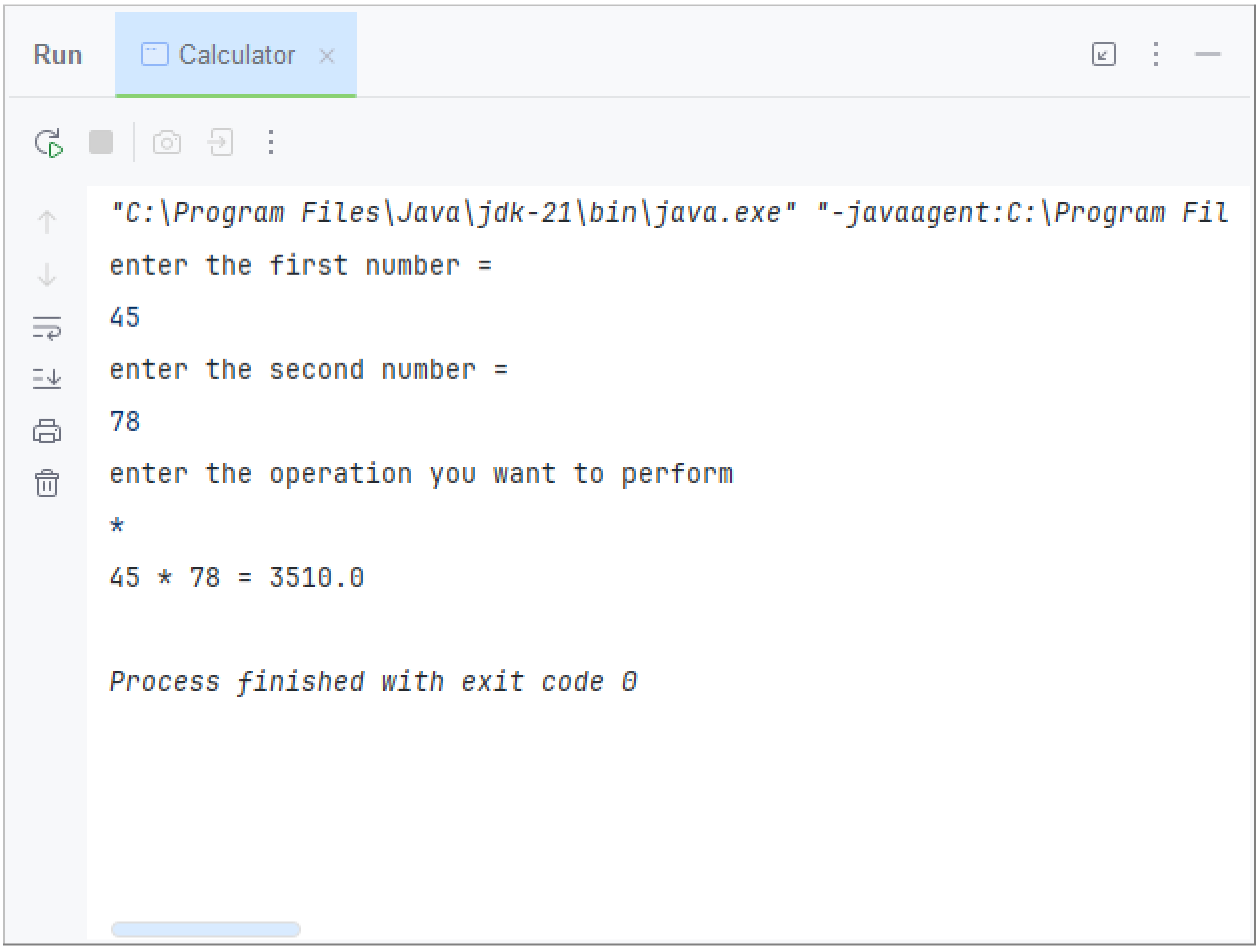
Program in Java to implement Mini Calculator

Code

```
public class Calculator {
  public static void main(String[] args)
 {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the first number =");
        int x = sc.nextInt();
        System.out.println("enter the second number =");
        int y = sc.nextInt();
        Scanner op = new Scanner(System.in);
        System.out.println("enter the operation you want to 
perform");
        char operation = op.next().charAt(0);

        double result;
        switch (operation)
```

```
      {
            case '+':
                result = x + y;
                System.out.println(x + " + " + y + " = " +
result);
                break;

            case '-':
                result = x - y;
                System.out.println(x + " - " + y + " = " +
result);
                break;

            case '*':
                result = x * y;
                System.out.println(x + " * " + y + " = " +
result);
                break;

            case '/':
                result = x / y;
                System.out.println(x + " / " + y + " = " +
result);
                break;

            default:
                System.out.println("Invalid operator!");
                break;
        }
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Fil
enter the first number =
45
enter the second number =
78
enter the operation you want to perform
*
45 * 78 = 3510.0

Process finished with exit code 0
```

Program 19

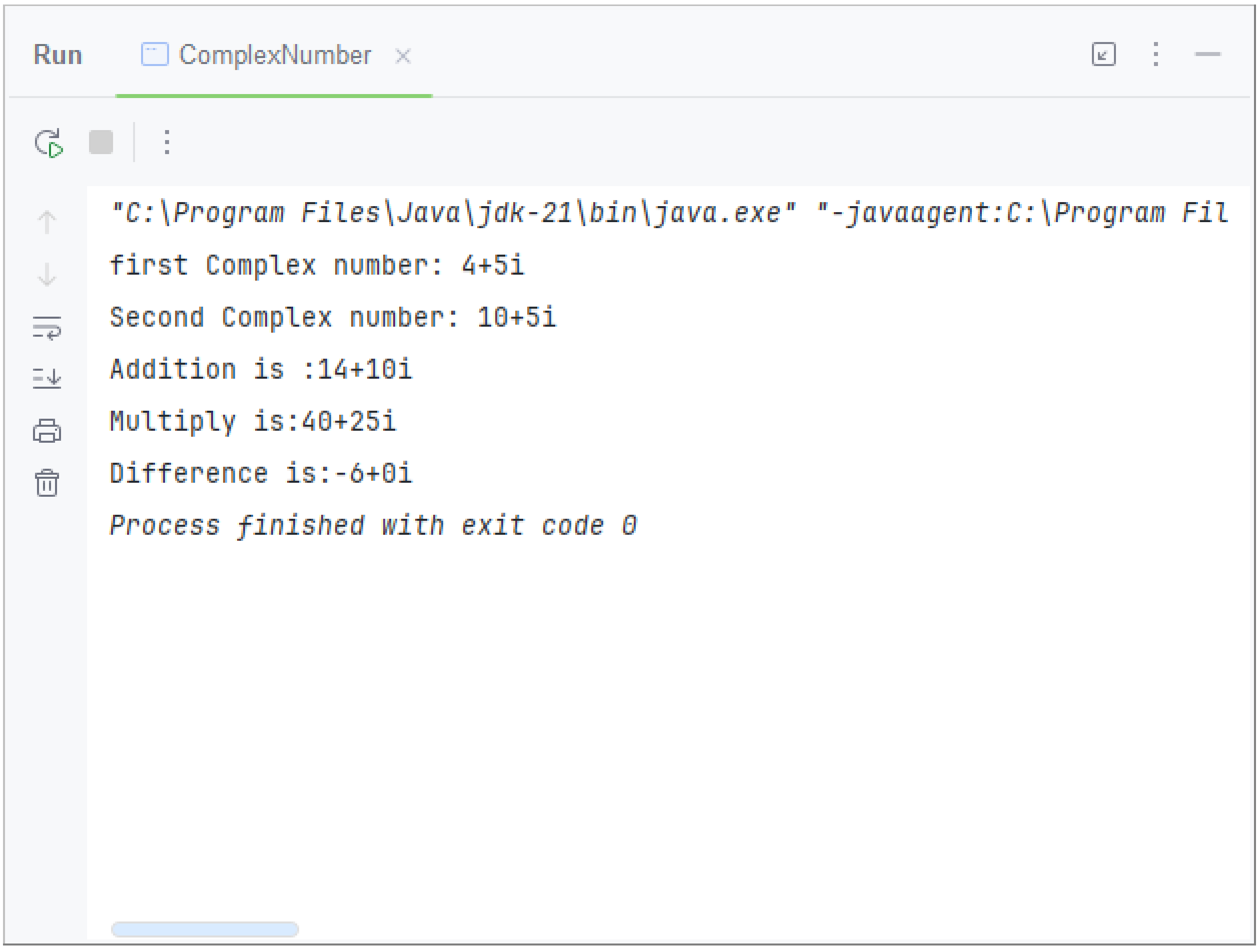
Program in Java to define two complex numbers and do the addition and multiplication and print the results

Code

```
 public ComplexNumber(int r, int i) {
         this.real = r;
         this.image = i;
    }

    public void showC() {
         System.out.print(this.real + "+" + this.image +
"i");
```

```
    }

    public static ComplexNumber add(ComplexNumber n1,
ComplexNumber n2) {   //ADDITION FUNCTION
        ComplexNumber res = new ComplexNumber(0, 0);
        res.real = n1.real + n2.real;
        res.image = n1.image + n2.image;
        return res;
    }

    public static ComplexNumber multiply(ComplexNumber n1,
ComplexNumber n2) {
        ComplexNumber rest = new ComplexNumber(0, 0);
        rest.real = n1.real * n2.real;
        rest.image = n1.image * n2.image;
        return rest;
    }

    public static ComplexNumber difference(ComplexNumber
n1, ComplexNumber n2) {
        ComplexNumber rest1 = new ComplexNumber(0, 0);
        rest1.real = n1.real - n2.real;
        rest1.image = n1.image - n2.image;
        return rest1;
    }

    public static void main(String arg[]) {
        ComplexNumber c1 = new ComplexNumber(4, 5);
        ComplexNumber c2 = new ComplexNumber(10, 5);
        System.out.print("first Complex number: ");
        c1.showC();
        System.out.print("\nSecond Complex number: ");
        c2.showC();
        ComplexNumber res = add(c1, c2);
        ComplexNumber rest = multiply(c1, c2);
        ComplexNumber rest1 = difference(c1, c2);

        System.out.print("\nAddition is :");
        res.showC();
        System.out.print("\nMultiply is:");
        rest.showC();
        System.out.print("\nDifference is:");
        rest1.showC();
    }
}
```

Output



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Fil
first Complex number: 4+5i
Second Complex number: 10+5i
Addition is :14+10i
Multiply is:40+25i
Difference is:-6+0i
Process finished with exit code 0
```

Program 1

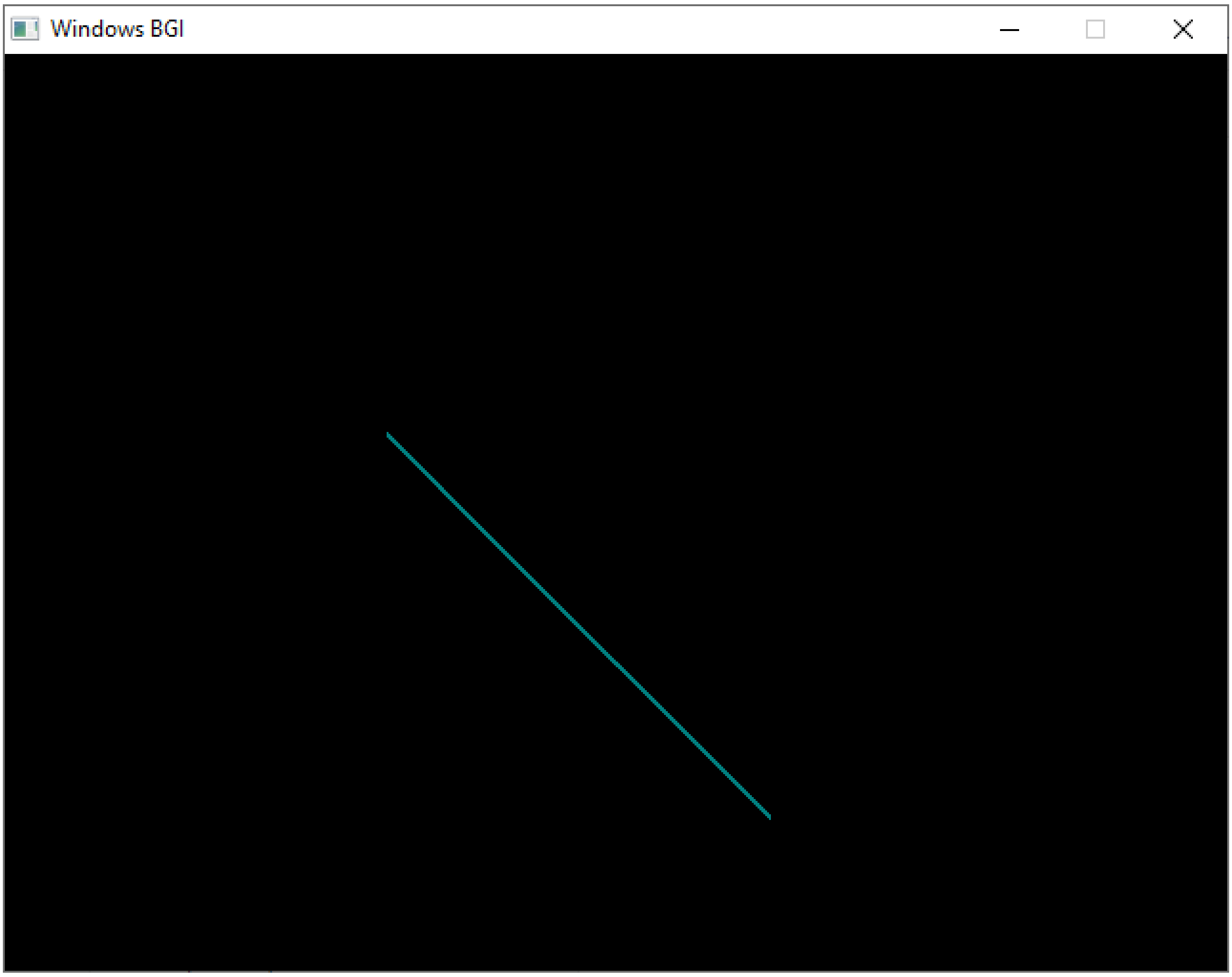
Program in C to implement DDA line drawing algorithm.

Code

```
 #include <graphics.h>

#include <math.h>


void ddaLine(int x1, int y1, int x2, int y2) {

  int dx = x2 - x1;
  int dy = y2 - y1;

  int steps = (abs(dx) > abs(dy)) ? abs(dx) : abs(dy);


  float xIncrement = dx / (float)steps;
  float yIncrement = dy / (float)steps;

  float x = x1;
  float y = y1;

  for (int i = 0; i <= steps; i++) {
    putpixel(round(x), round(y), CYAN);
    putpixel(round(x), round(y)-1, CYAN);//to make it more
visible
    putpixel(round(x), round(y)-2, CYAN);

    x += xIncrement;
    y += yIncrement;
    delay(20);
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");
  int x1 = 200;
```

```
    int x2 = 400;
    int y1 = 200;
    int y2 = 400;
    ddaLine(x1, y1, x2, y2);
    delay(10000);
    return 0;
}
```

Output



Program 2

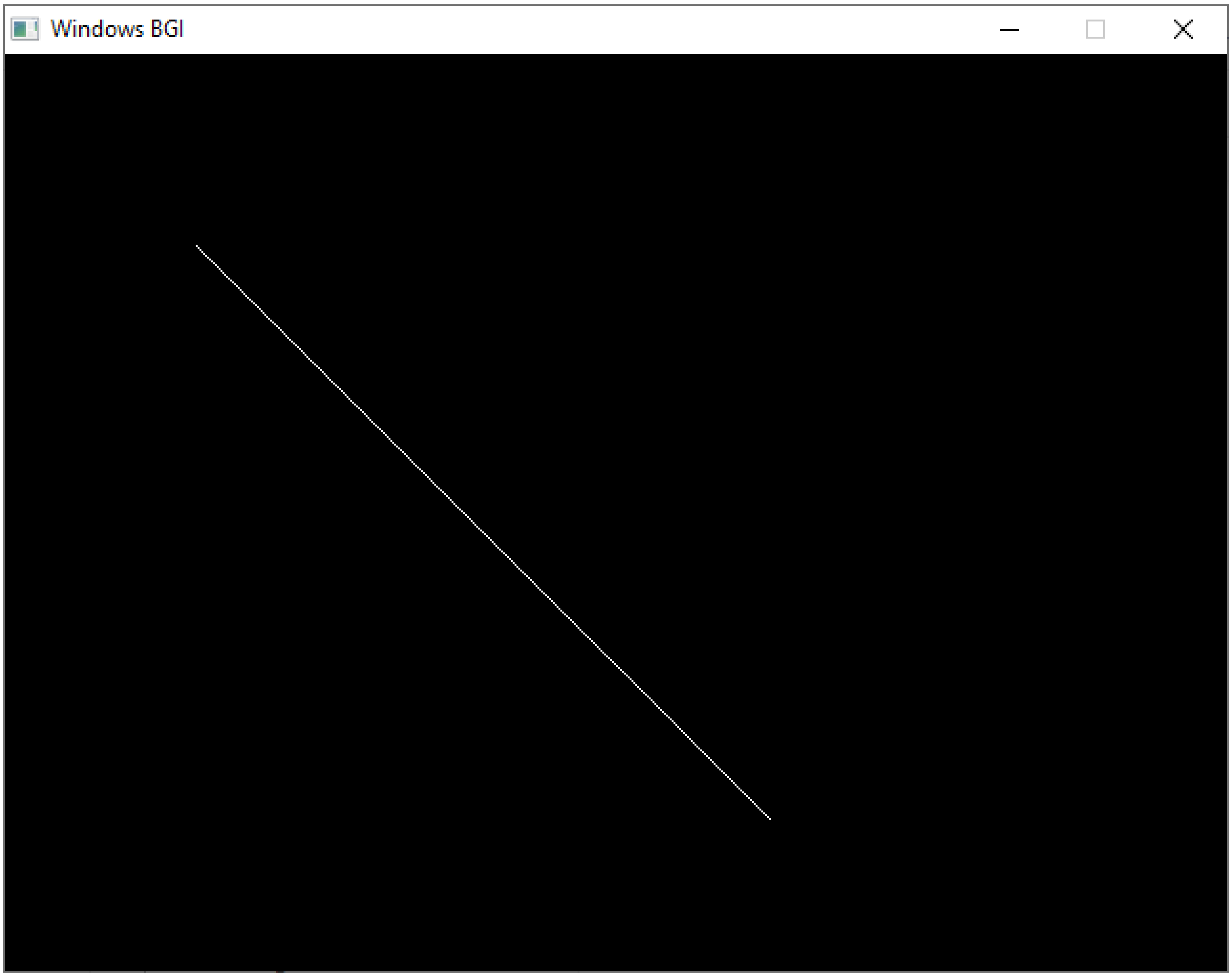
Program in C to implement Bresenham’s line drawing algorithm.

Code

```
 #include <graphics.h>

#include <stdio.h>
#include <stdlib.h>

void drawLine(int x1, int y1, int x2, int y2) {
  int dx = abs(x2 - x1);
  int dy = abs(y2 - y1);
  int sx = (x2 >= x1) ? 1 : -1;
  int sy = (y2 >= y1) ? 1 : -1;
  int err = dx - dy;
  int e2;

  while (1) {
    putpixel(x1, y1, WHITE);
    if (x1 == x2 && y1 == y2) {
      break;
    }

    e2 = 2 * err;
    if (e2 > -dy) {
      err -= dy;
      x1 += sx;
    }
    if (e2 < dx) {
      err += dx;
      y1 += sy;
    }
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");
```

```
    int x1, y1, x2, y2;
    x1=100;
    y1=100;
    x2=400;
    y2=400;
    drawLine(x1, y1, x2, y2);
    getch();
    closegraph();

    return 0;
}
```

Output



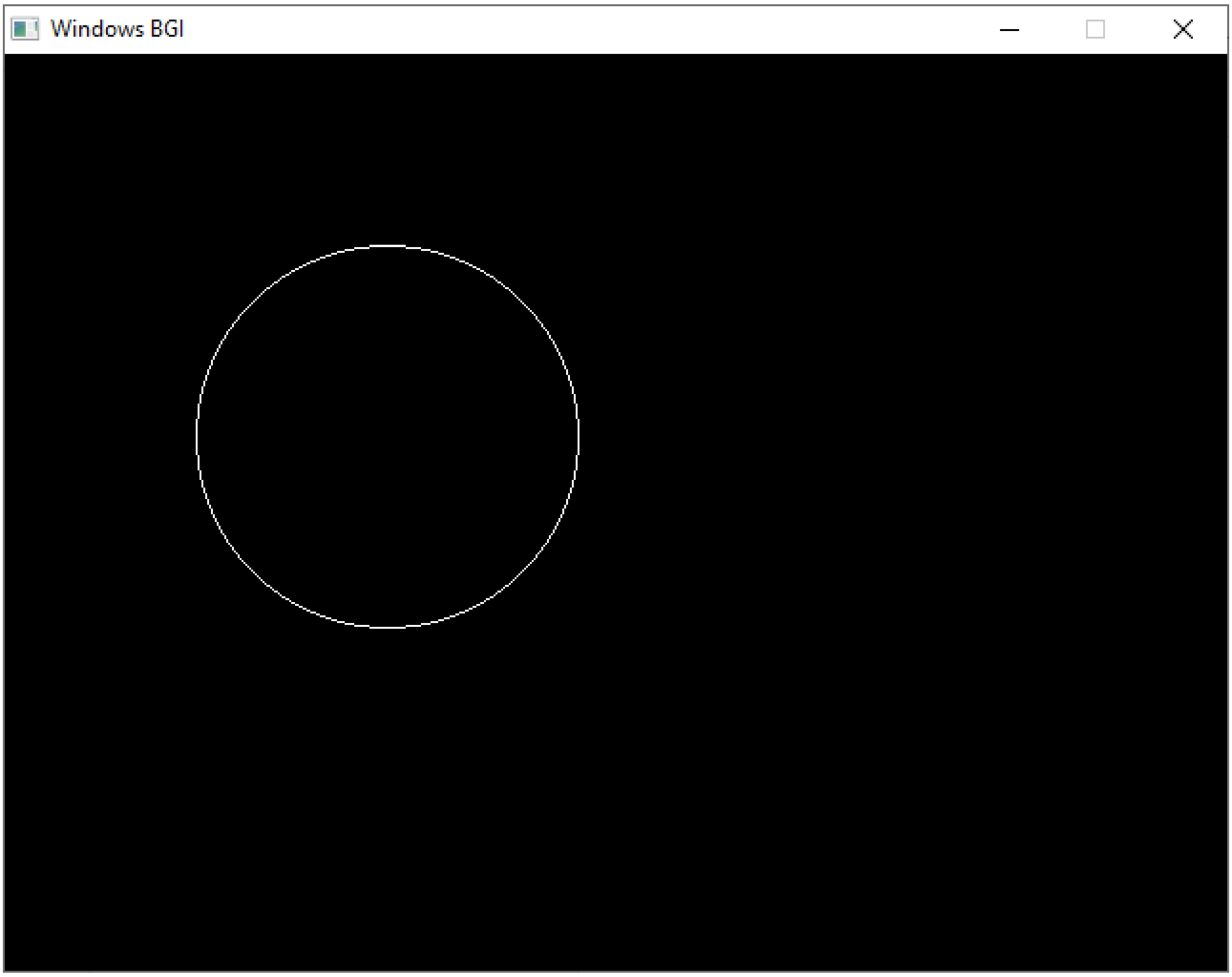
Program 3

Program in C to implement Circle drawing using polynomial approach.

Code

```c
#include <graphics.h>
#include <stdio.h>
#include <math.h>

void drawCircle(int xc, int yc, int r) {
  int x = 0;
  float y = r;

  while (x <= y) {
    putpixel(xc + x, yc + round(y), WHITE);
    putpixel(xc - x, yc + round(y), WHITE);
    putpixel(xc + x, yc - round(y), WHITE);
    putpixel(xc - x, yc - round(y), WHITE);
    putpixel(xc + round(y), yc + x, WHITE);
    putpixel(xc - round(y), yc + x, WHITE);
    putpixel(xc + round(y), yc - x, WHITE);
    putpixel(xc - round(y), yc - x, WHITE);

    x++;
    y = sqrt(r * r - x * x);
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  int xc, yc, r;
  xc=200;
  yc=200;
  r=100;

  drawCircle(xc, yc, r);

  getch();
  closegraph();

  return 0;}
```

Output



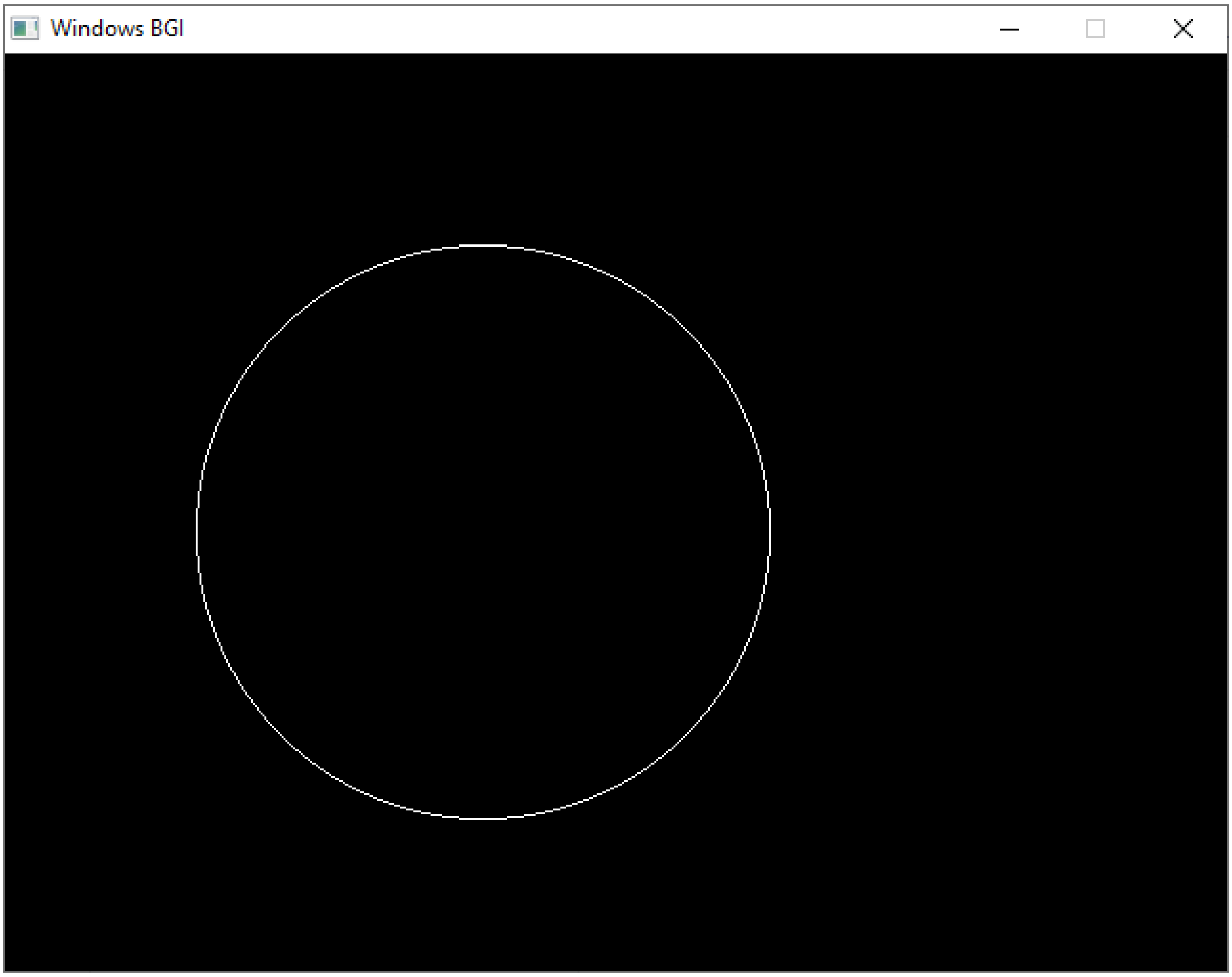
Program 4

Program in C to implement Circle drawing using Bresenham’s approach.

Code

```c
#include <graphics.h>
#include <stdio.h>

void drawCircle(int xc, int yc, int r) {
  int x = 0, y = r, d = 3 - 2 * r;
  while (x <= y) {
    putpixel(xc + x, yc + y, WHITE);
    putpixel(xc - x, yc + y, WHITE);
    putpixel(xc + x, yc - y, WHITE);
    putpixel(xc - x, yc - y, WHITE);
    putpixel(xc + y, yc + x, WHITE);
    putpixel(xc - y, yc + x, WHITE);
    putpixel(xc + y, yc - x, WHITE);
    putpixel(xc - y, yc - x, WHITE);

    if (d < 0)
      d += 4 * x + 6;
    else {
      d += 4 * (x - y) + 10;
      y--;
    }
    x++;
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  int xc = 250, yc = 250, radius = 150;
  drawCircle(xc, yc, radius);

  getch();
  closegraph();
  return 0;
}
```

Output



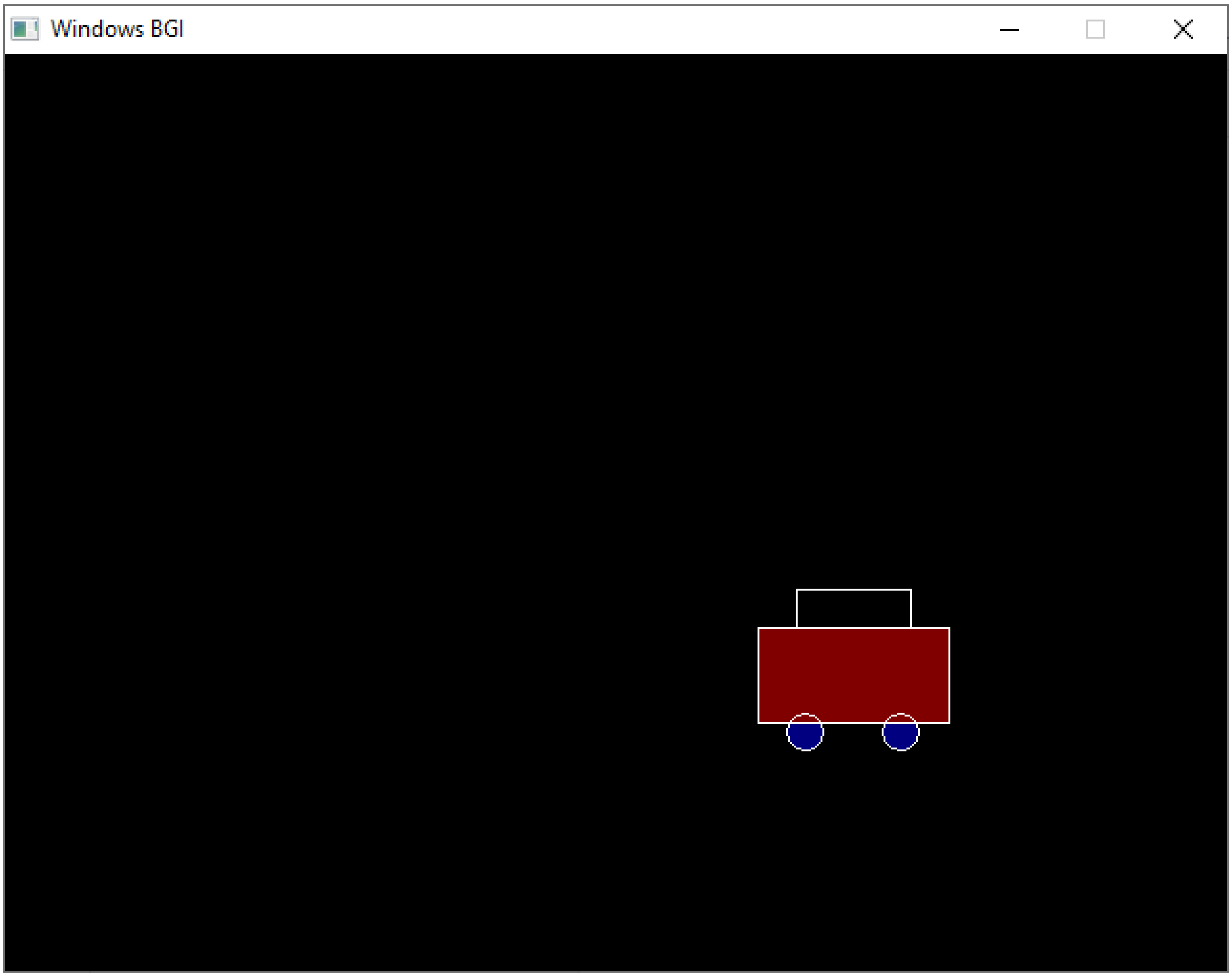
Program 5

Program in C to implement a moving object like car / fan / moving man

Code

```
 #include <graphics.h>

#include <conio.h>

void drawCar(int x, int y) {
  // Car body
  rectangle(x, y, x + 100, y + 50);
  rectangle(x + 20, y - 20, x + 80, y); // Car top
  setfillstyle(SOLID_FILL, RED);
  floodfill(x + 1, y + 1, WHITE);

  // Wheels
  circle(x + 25, y + 55, 10);
  circle(x + 75, y + 55, 10);
  setfillstyle(SOLID_FILL, BLUE);
  floodfill(x + 25, y + 55, WHITE);
  floodfill(x + 75, y + 55, WHITE);
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  int x = 0, y = 300;

  while (x < getmaxx()) {
    cleardevice();
    drawCar(x, y);
    delay(20);
    x += 2;
  }

  getch();
  closegraph();
  return 0;}
```

Output



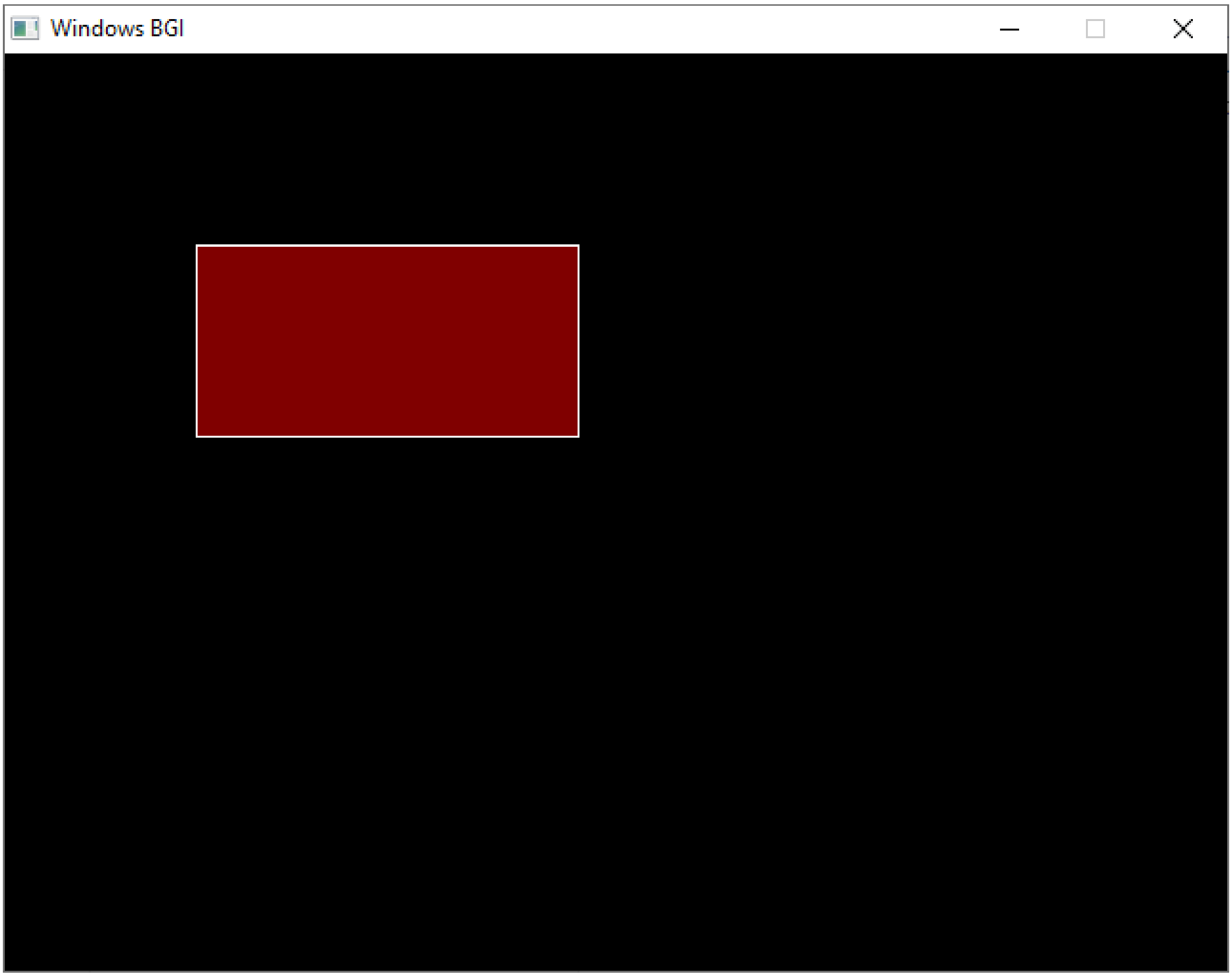
Program 6

Program in C to implement color filling in a closed object

Code

```
 #include <graphics.h>

#include <conio.h>

void boundaryFill(int x, int y, int fillColor, int
boundaryColor) {
  int currentColor = getpixel(x, y);
  if (currentColor != fillColor && currentColor !=
boundaryColor) {
    putpixel(x, y, fillColor);
    delay(1/3);
    boundaryFill(x + 1, y, fillColor, boundaryColor);
    boundaryFill(x - 1, y, fillColor, boundaryColor);
    boundaryFill(x, y + 1, fillColor, boundaryColor);
    boundaryFill(x, y - 1, fillColor, boundaryColor);
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");
  rectangle(100, 100, 300, 200);
  setcolor(WHITE);
  int x = 150, y = 150;
  int fillColor = RED;
  int boundaryColor = WHITE;
  boundaryFill(x, y, fillColor, boundaryColor);
  getch();
  closegraph();
  return 0;
}
```

Output



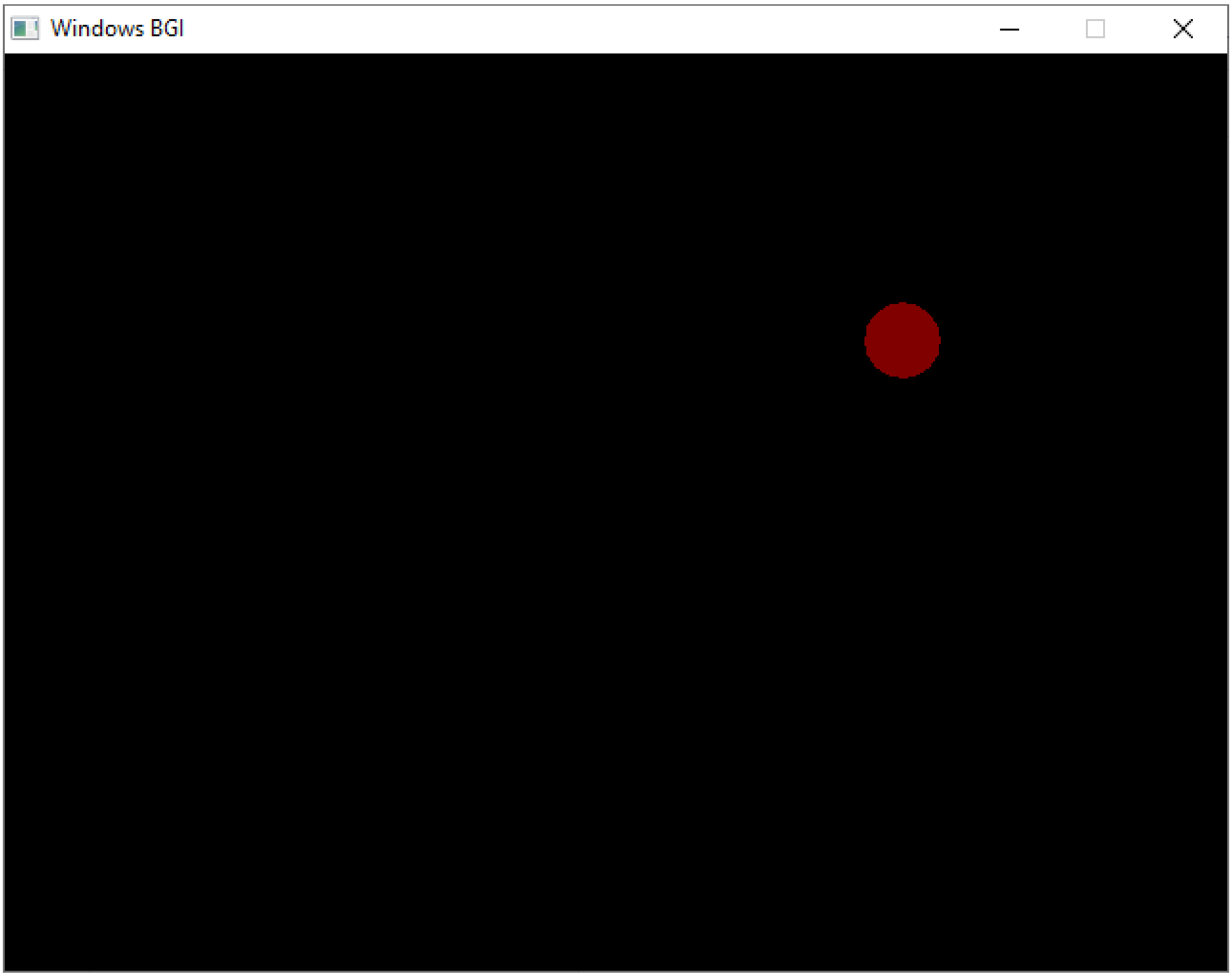
Program 7

Program in C to implement bouncing ball

Code

```c
#include <graphics.h>
#include <conio.h>

void bouncingBall() {
  int x = 200, y = 200;
  int radius = 20;
  int dx = 5, dy = 5;

  int maxX = getmaxx();
  int maxY = getmaxy();

  while (!kbhit()) {
    cleardevice();
    setcolor(RED);
    setfillstyle(SOLID_FILL, RED);
    circle(x, y, radius);
    floodfill(x, y, RED);

    delay(20);
    x += dx;
    y += dy;

    if (x + radius >= maxX || x - radius <= 0) {
      dx = -dx;
    }
    if (y + radius >= maxY || y - radius <= 0) {
      dy = -dy;
    }
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  bouncingBall();
  getch();
  closegraph();
  return 0;
}
```

Output



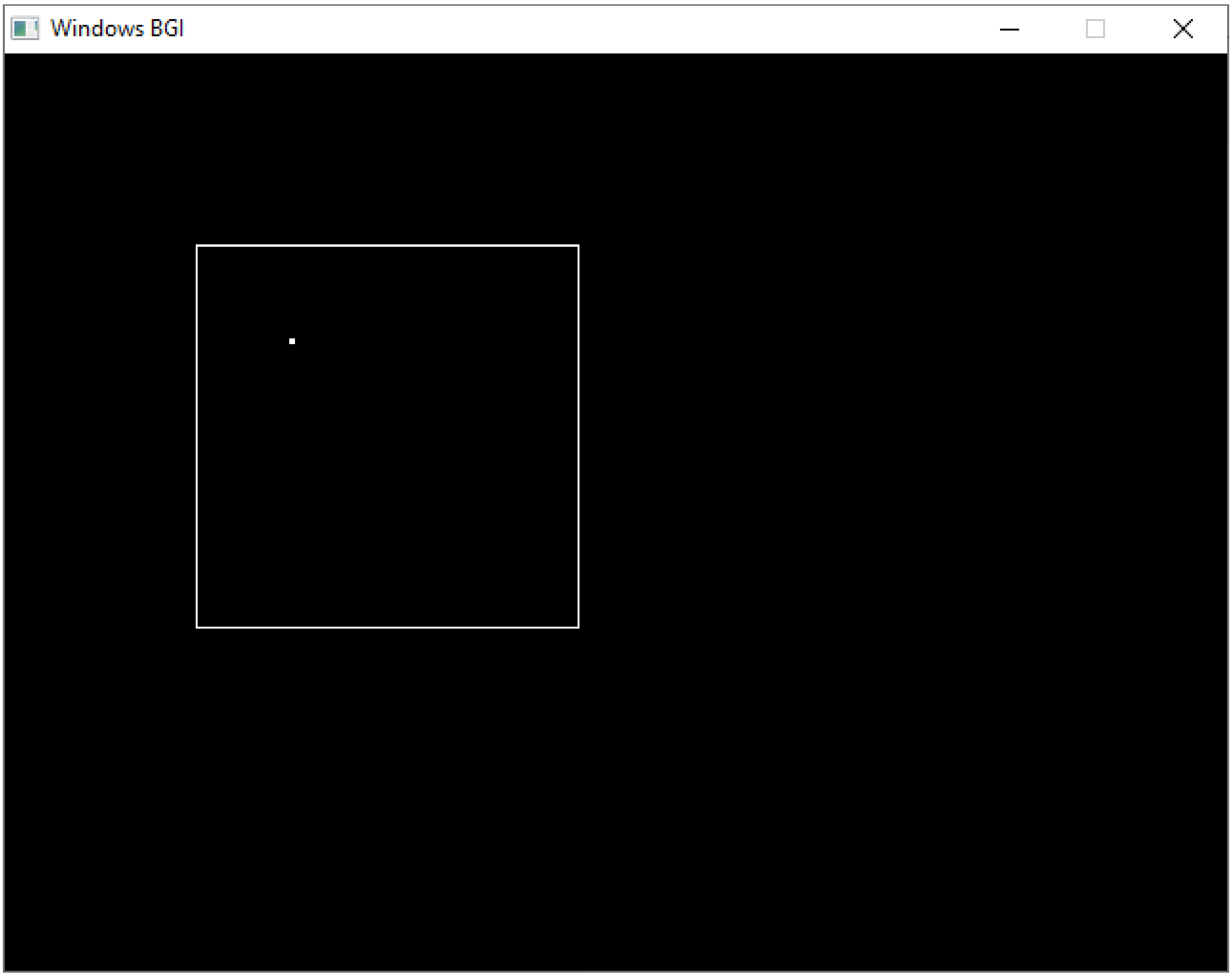
Program 8

Program in C to implement point clipping/line clipping method.

Code

```
#include <graphics.h>

#include <stdio.h>

void pointClipping(int x, int y, int xMin, int xMax, int yMin,
int yMax) {
  if (x >= xMin && x <= xMax && y >= yMin && y <= yMax) {
    putpixel(x, y, WHITE);
    putpixel(x - 1, y - 1, WHITE);
    putpixel(x + 1, y + 1, WHITE);
    putpixel(x - 1, y + 1, WHITE); //to make it more visible
    putpixel(x + 1, y - 1, WHITE);
    putpixel(x, y - 1, WHITE);
    putpixel(x, y + 1, WHITE);
    putpixel(x - 1, y, WHITE);
    putpixel(x + 1, y, WHITE);
    printf("Point (%d, %d) is inside the clipping window.\n", x,
y);
  } else {
    printf("Point (%d, %d) is outside the clipping window.\n", x,
y);
  }
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  int xMin = 100, yMin = 100, xMax = 300, yMax = 300;
  rectangle(xMin, yMin, xMax, yMax);

  pointClipping(150, 150, xMin, xMax, yMin, yMax);
  pointClipping(50, 50, xMin, xMax, yMin, yMax);

  getch();
  closegraph();
  return 0;
}
```

Output



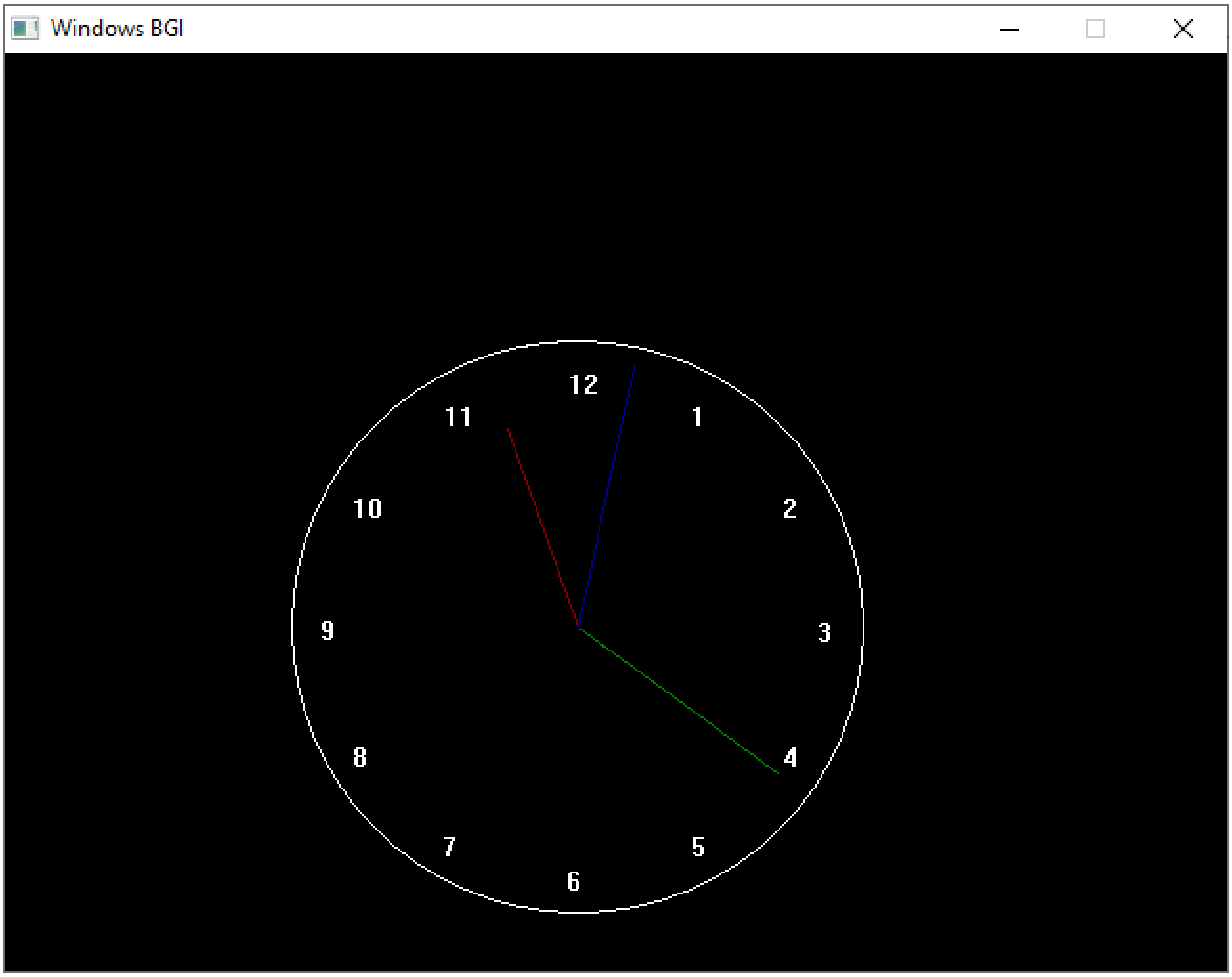
Program 9

Program in C to implement analog clock

Code

```
#include <graphics.h>
#include <time.h>
#include <math.h>

#define PI 3.14159265358979323846

float degToRad(float degrees) {
  return (degrees * PI) / 180.0;
}

void drawHand(int xCenter, int yCenter, int length, float angle,
int color) {
  int xEnd = xCenter + length * cos(angle);
  int yEnd = yCenter - length * sin(angle);
  setcolor(color);
  line(xCenter, yCenter, xEnd, yEnd);
}

void drawClockFace(int xCenter, int yCenter, int radius) {
  setcolor(WHITE);
  circle(xCenter, yCenter, radius);

  for (int i = 1; i <= 12; i++) {
    float angle = degToRad(90 - i * 30);
    int x = xCenter + (radius - 20) * cos(angle);
    int y = yCenter - (radius - 20) * sin(angle);
    char label[3];
    sprintf(label, "%d", i);
    outtextxy(x - 5, y - 5, label);
  }
}

void drawAnalogClock() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  int xCenter = 300, yCenter = 300, radius = 150;
  while (!kbhit()) {
    cleardevice();

    drawClockFace(xCenter, yCenter, radius);

    time_t currentTime = time(NULL);
    struct tm *localTime = localtime(&currentTime);

    int hours = localTime->tm_hour % 12;
    int minutes = localTime->tm_min;
    int seconds = localTime->tm_sec;

    float secondAngle = degToRad(90 - seconds * 6);
    float minuteAngle = degToRad(90 - (minutes * 6 + seconds *
0.1));
    float hourAngle = degToRad(90 - (hours * 30 + minutes *
0.5));

    drawHand(xCenter, yCenter, radius - 40, hourAngle, RED);
    drawHand(xCenter, yCenter, radius - 20, minuteAngle, GREEN);
    drawHand(xCenter, yCenter, radius - 10, secondAngle, BLUE);

    delay(1000);
  }

  closegraph();
}

int main() {
  drawAnalogClock();
  return 0;
}
```

Output



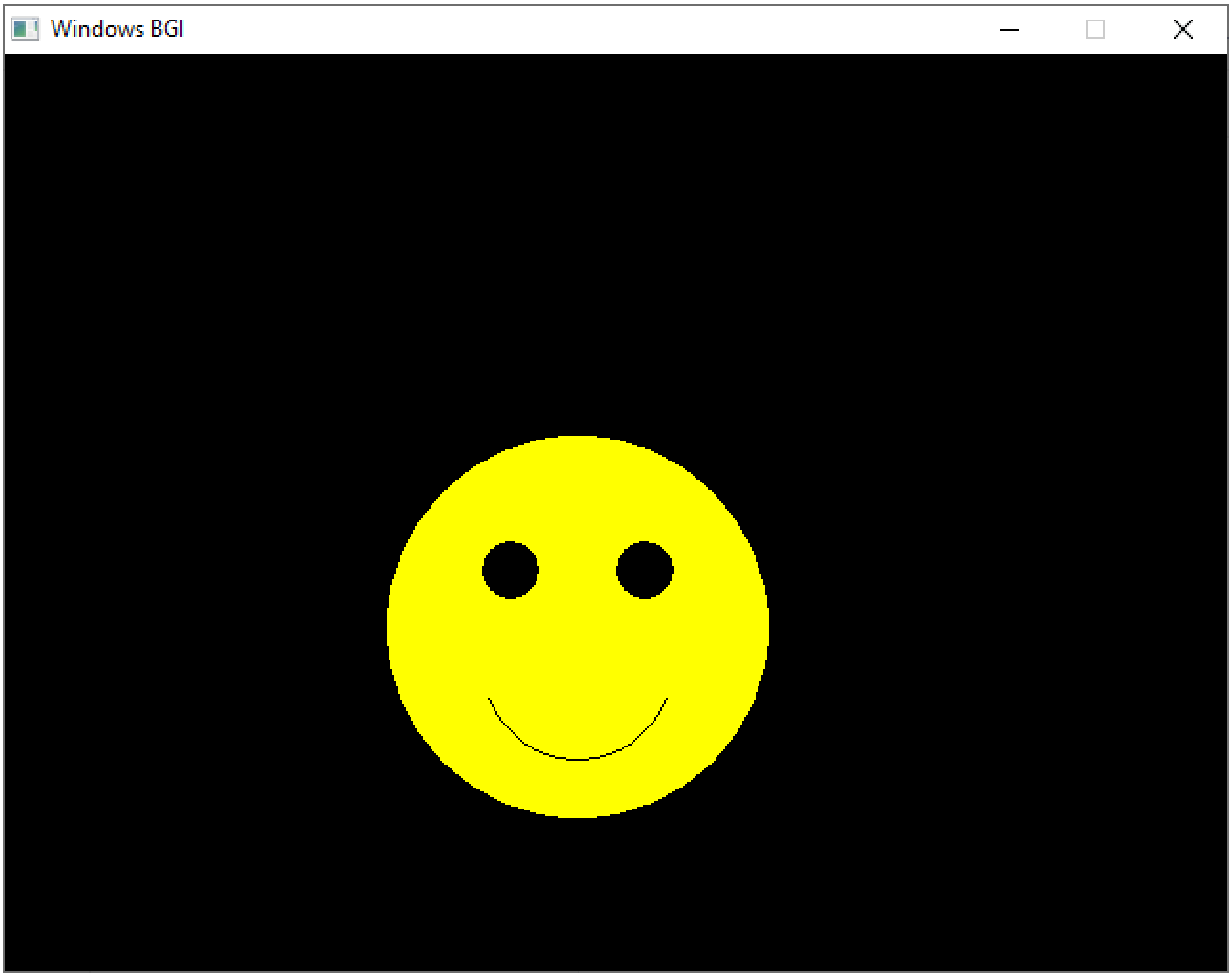
Program 10

Program in C to implement a three dimensional object / smiley

Code

```
 #include <graphics.h>

#include <conio.h>
#include <stdio.h>

void drawSmiley() {
  int x = 300, y = 300;
  int faceRadius = 100;
  int eyeRadius = 15;

  setcolor(YELLOW);
  setfillstyle(SOLID_FILL, YELLOW);
  circle(x, y, faceRadius);
  floodfill(x, y, YELLOW);

  setcolor(BLACK);
  setfillstyle(SOLID_FILL, BLACK);
  circle(x - 35, y - 30, eyeRadius);
  floodfill(x - 35, y - 30, BLACK);
  circle(x + 35, y - 30, eyeRadius);
  floodfill(x + 35, y - 30, BLACK);

  setcolor(BLACK);
  arc(x, y + 20, 200, 340, 50);
}

int main() {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "");

  drawSmiley();

  getch();
  closegraph();
  return 0;}
```

Output



Computer Graphics

S No.	Program	Remarks
1.	Program to implement DDA line Drawing Algorithm.	
2.	Program to implement Bresenham’s Line Drawing Algorithm.	
3.	Program to implement Circle Drawing using Polynomial Approach.	
4.	Program to implement Circle drawing using Bresenham’s Approach.	
5.	Program to implement a moving object like car/fan/moving man.	
6.	Program to implement colour filling in a closed object.	
7.	Program to implement Bouncing Ball.	
8.	Program to implement Point Clipping/ Line Clipping Method.	
9.	Program to implement Analogue Clock.	
10.	Program to implement a 3D (Three Dimensional) Object / Smiley.	