



Welcome To Tech By WebCoder

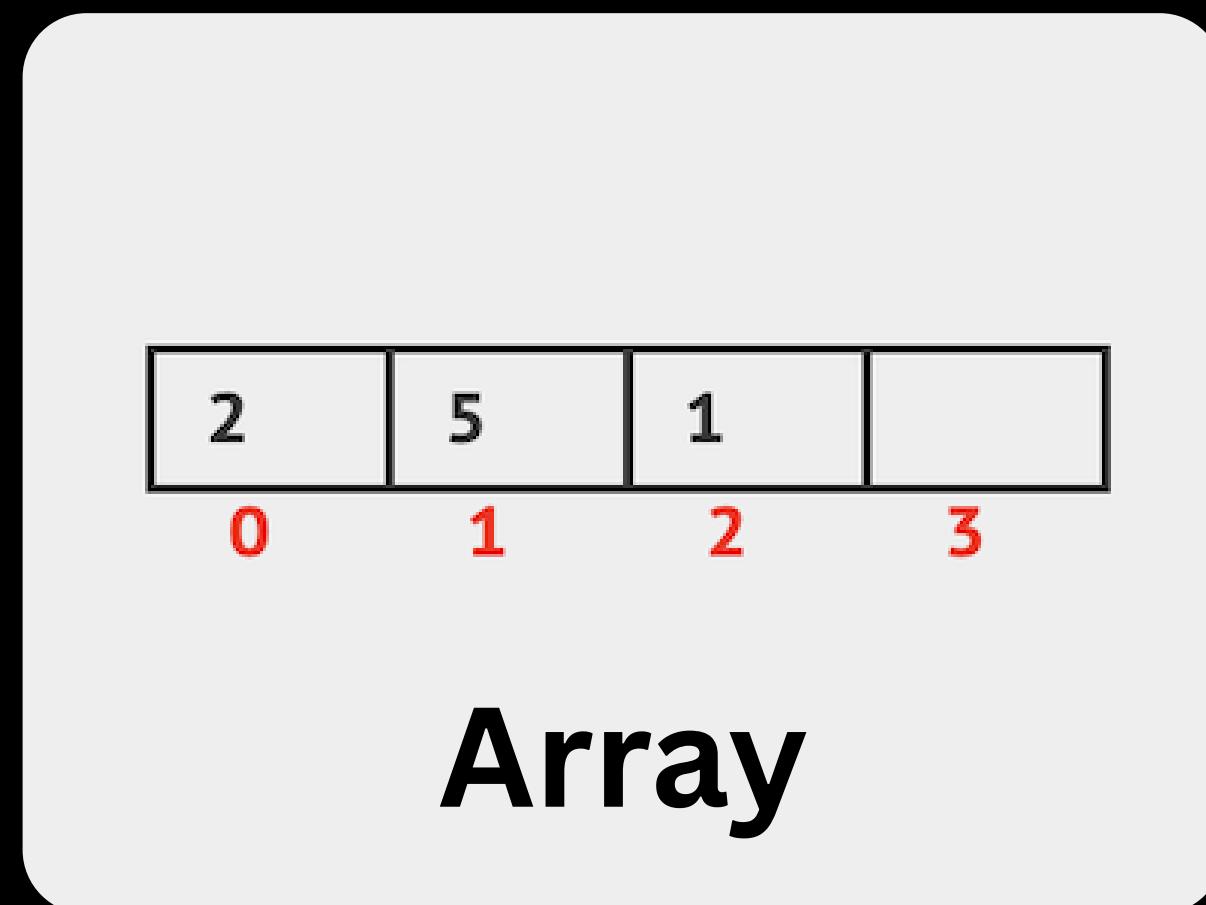
DSA Series



Topic : 1

Operation In Array

1. Insert
2. Remove
3. Select
4. Update





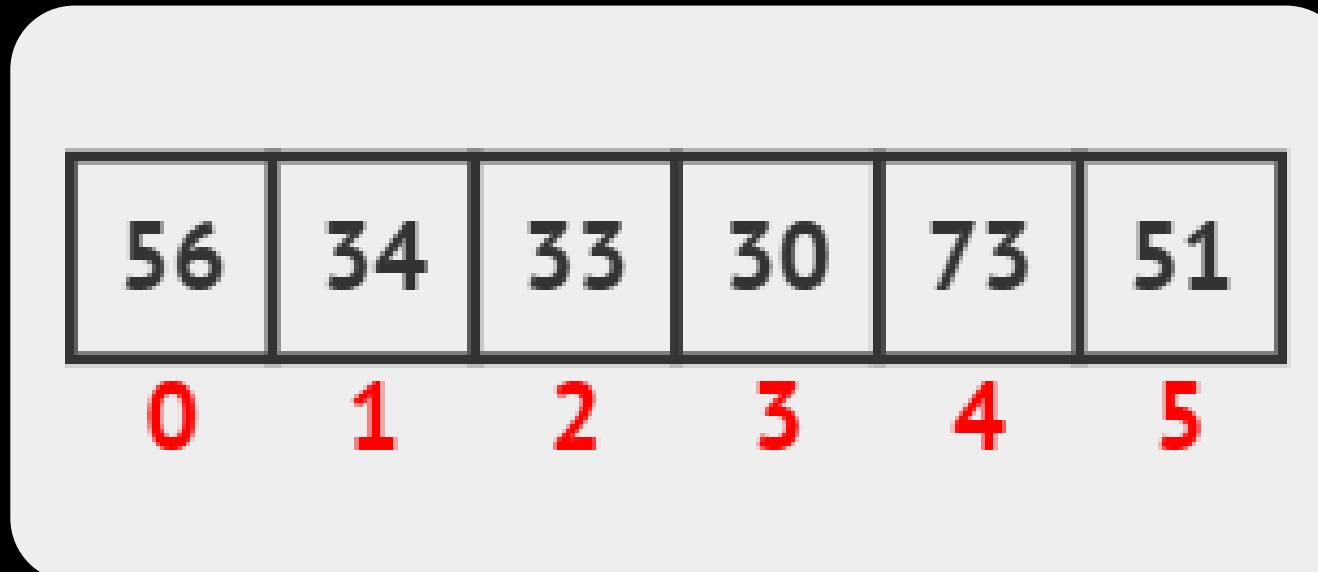
1.

INSERT Operation In Array

#techbywebcoder

INSERT Operation In Array

Algorithm



**Insert 30 to index
3 of array**

Step 1: Start

Step 2: Input value to insert

Step 3: Input position where the value should be inserted

Step 4: If position < 0 OR position > size OR array is full

Print "Invalid position" and stop

Step 5: For i = size-1 down to position

Move arr[i] to arr[i + 1]

Step 6: Insert value at arr[position]

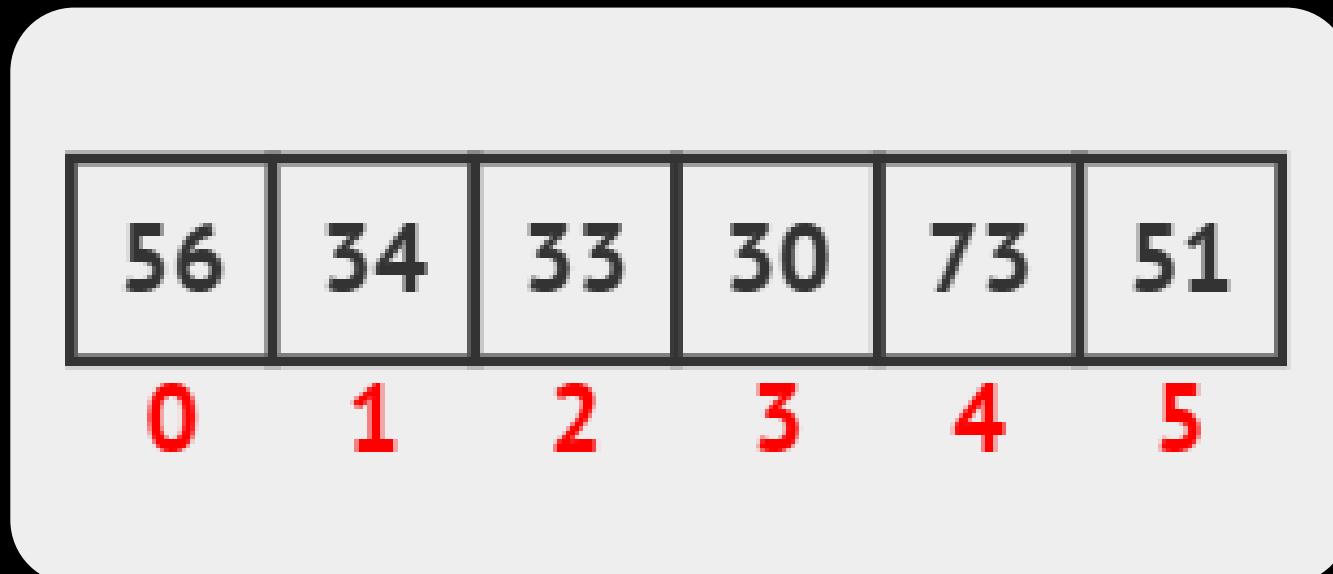
Step 7: Increase size by 1

Step 8: Stop

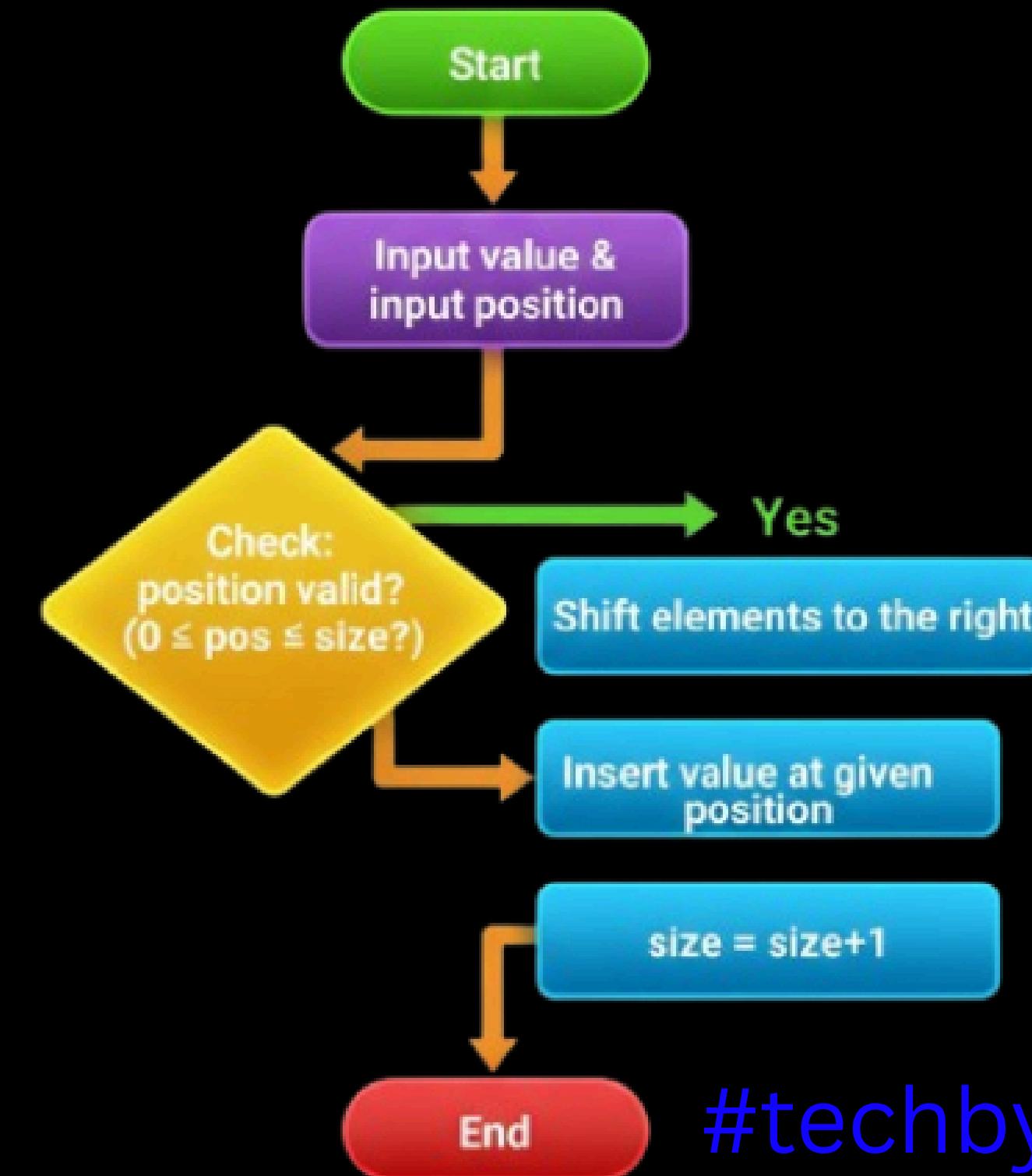
#techbywebcoder

INSERT Operation In Array

Flowchart

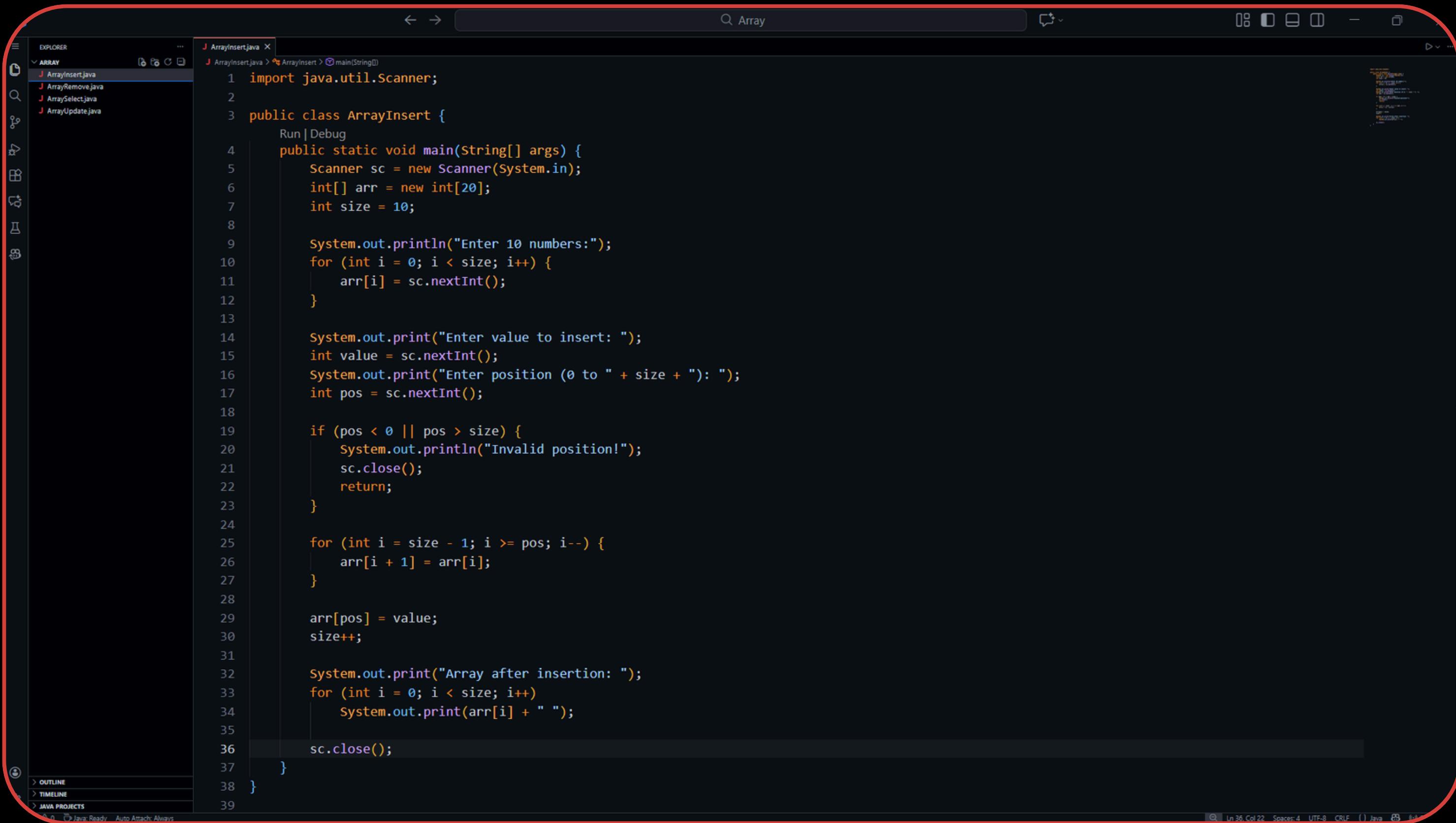


Insert 30 to index
3 of array



#techbywebcoder

Program



A screenshot of a Java IDE interface, specifically showing a code editor with a red rounded rectangle highlighting the main window. The title bar reads "Array". The code editor displays a Java program named "ArrayInsert.java". The code implements an array insertion operation. It uses a Scanner to read 10 integers into an array of size 10. It then prompts the user to enter a value to insert and its position. If the position is valid, it shifts elements to make space and inserts the value. Finally, it prints the modified array. The code is well-formatted with line numbers and color-coded syntax.

```
1 import java.util.Scanner;
2
3 public class ArrayInsert {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[20];
7         int size = 10;
8
9         System.out.println("Enter 10 numbers:");
10        for (int i = 0; i < size; i++) {
11            arr[i] = sc.nextInt();
12        }
13
14        System.out.print("Enter value to insert: ");
15        int value = sc.nextInt();
16        System.out.print("Enter position (0 to " + size + "): ");
17        int pos = sc.nextInt();
18
19        if (pos < 0 || pos > size) {
20            System.out.println("Invalid position!");
21            sc.close();
22            return;
23        }
24
25        for (int i = size - 1; i >= pos; i--) {
26            arr[i + 1] = arr[i];
27        }
28
29        arr[pos] = value;
30        size++;
31
32        System.out.print("Array after insertion: ");
33        for (int i = 0; i < size; i++)
34            System.out.print(arr[i] + " ");
35
36        sc.close();
37    }
38 }
39 }
```

The IDE's sidebar includes an "EXPLORER" view showing other files like "ArrayRemove.java", "ArraySelect.java", and "ArrayUpdate.java". The bottom status bar shows "Java: Ready" and "Auto Attach: Always". The bottom right corner shows the current file path as "C:\Users\user\Documents\Java\src\com\example\array\ArrayInsert.java".

Output

```
C:\Windows\System32\cmd.e × + ▾ - ⌂
```

```
D:\system\DSA\1. Array Operation\Array>javac ArrayInsert.java
D:\system\DSA\1. Array Operation\Array>java ArrayInsert
Enter The 10 Number
10
20
30
40
50
60
70
80
90
100
Enter Value To Insert : 7
Enter Position (0 to 10):1
Array After Inseration :10
7
20
30
40
50
60
70
80
90
100
D:\system\DSA\1. Array Operation\Array>
```



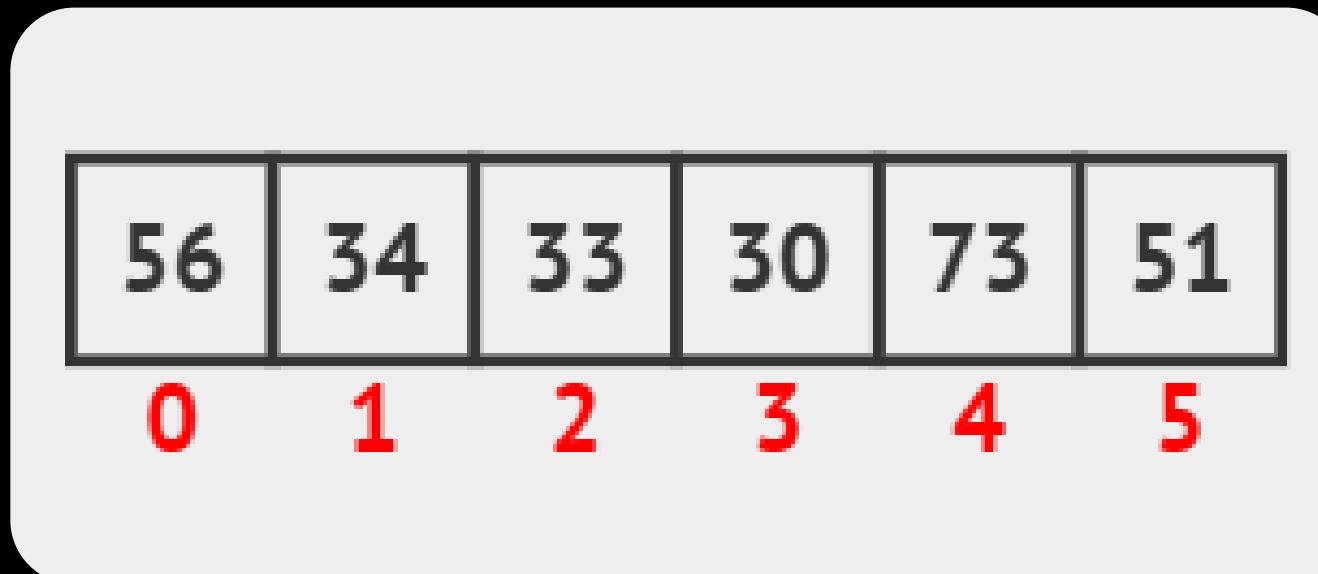
2.

REMOVE Operation In Array

#techbywebcoder

REMOVE Operation In Array

Algorithm



**Remove array[i]
from array**

Step 1: Start

Step 2: Input position to remove

Step 3: If position < 0 OR position ≥ size

Print "Invalid position" and stop

Step 4: For i = position to size-2

Move arr[i + 1] to arr[i]

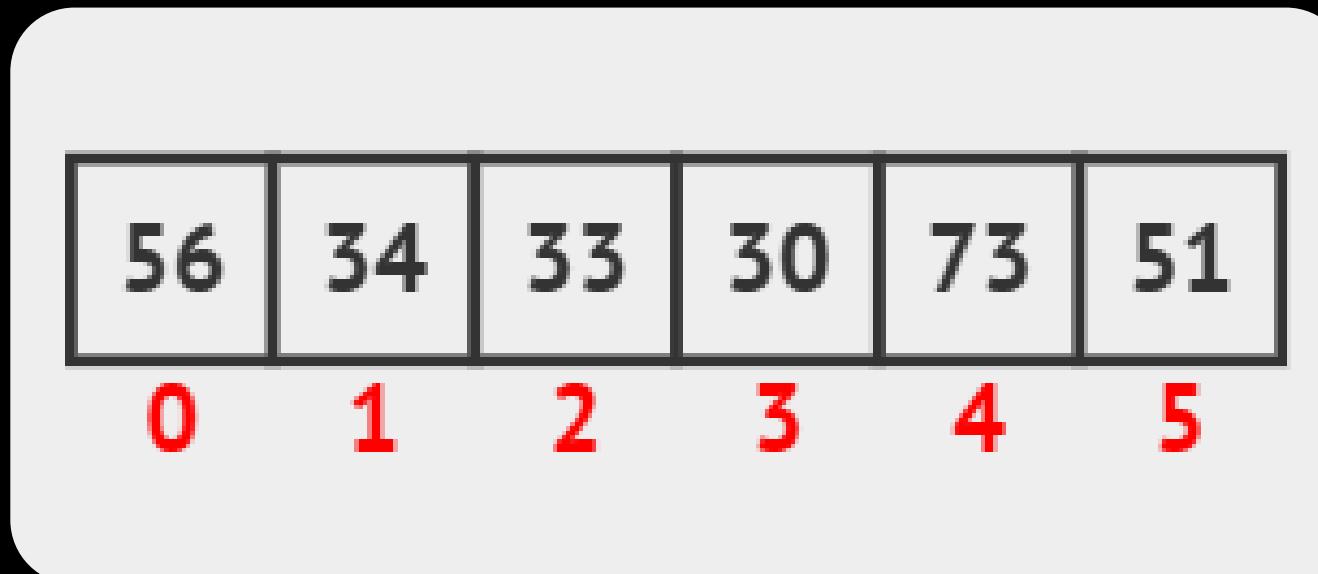
Step 5: Decrease size by 1

Step 6: Stop

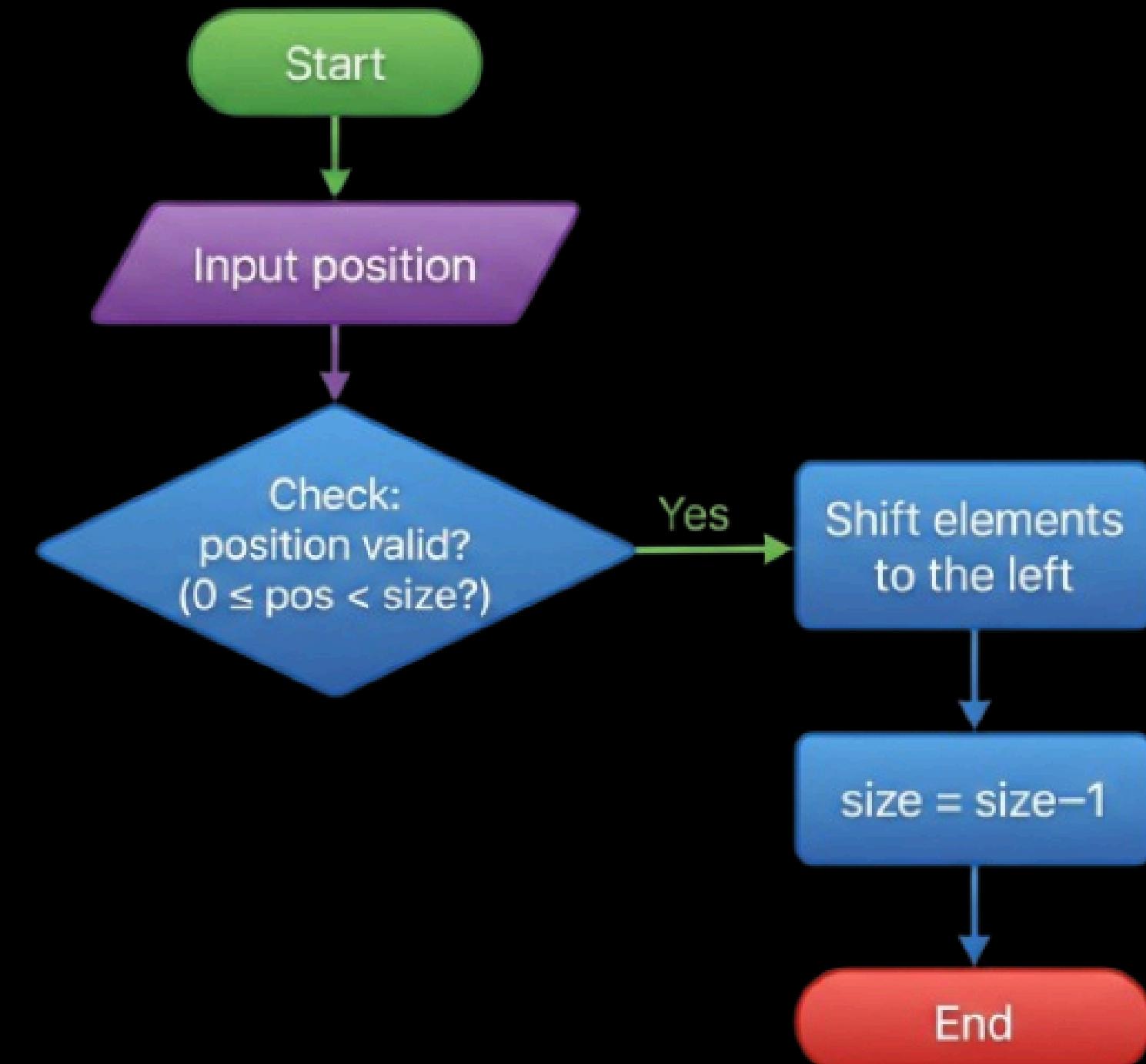
#techbywebcoder

REMOVE Operation In Array

Flowchart

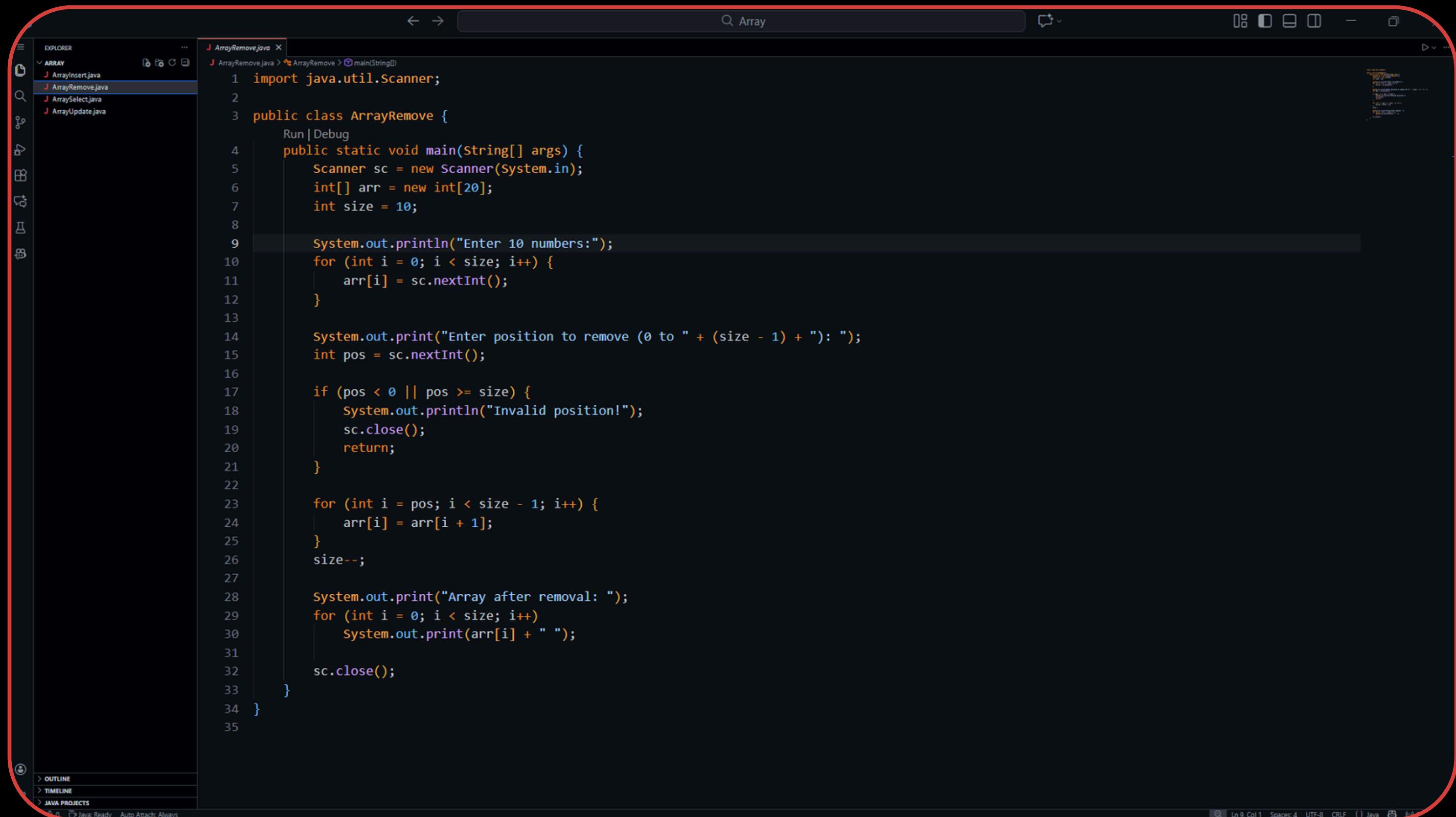


Remove array[i]
from array



#techbywebcoder

Program



The screenshot shows a Java development environment with a dark theme. A red rounded rectangle highlights the main window area. In the top bar, there is a search field with the text "Array". The left sidebar contains an "EXPLORER" panel with a tree view showing a project named "ARRAY" with four files: "ArrayInsert.java", "ArrayRemove.java" (which is selected), "ArraySelect.java", and "ArrayUpdate.java". The main editor area displays the "ArrayRemove.java" code:

```
1 import java.util.Scanner;
2
3 public class ArrayRemove {
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int[] arr = new int[20];
8         int size = 10;
9
10        System.out.println("Enter 10 numbers:");
11        for (int i = 0; i < size; i++) {
12            arr[i] = sc.nextInt();
13        }
14
15        System.out.print("Enter position to remove (0 to " + (size - 1) + "): ");
16        int pos = sc.nextInt();
17
18        if (pos < 0 || pos >= size) {
19            System.out.println("Invalid position!");
20            sc.close();
21            return;
22        }
23
24        for (int i = pos; i < size - 1; i++) {
25            arr[i] = arr[i + 1];
26        }
27        size--;
28
29        System.out.print("Array after removal: ");
30        for (int i = 0; i < size; i++)
31            System.out.print(arr[i] + " ");
32
33        sc.close();
34    }
35}
```

The status bar at the bottom shows "Java: Ready" and "Auto Attach: Always". The bottom right corner includes standard window controls (minimize, maximize, close) and a small "Java" icon.

Output

```
C:\Windows\System32\cmd.e × + ▾ - ☒

D:\system\DSA\1. Array Operation\Array>javac ArrayRemove.java

D:\system\DSA\1. Array Operation\Array>java ArrayRemove
Enter The 10 Number:
10
20
30
40
50
60
70
80
90
100
Enter The Position To remove
7
Array After Removal :10
20
30
40
50
60
70
90
100

D:\system\DSA\1. Array Operation\Array>
```



3.

SELECT Operation In Array

#techbywebcoder

SELECT Operation In Array

Algorithm



Step 1: Start

Step 2: Input value to search

Step 3: For $i = 0$ to size-1

If $\text{arr}[i] = \text{value}$

Print "Value found at index i" and stop

Step 4: If loop ends, print "Value not found"

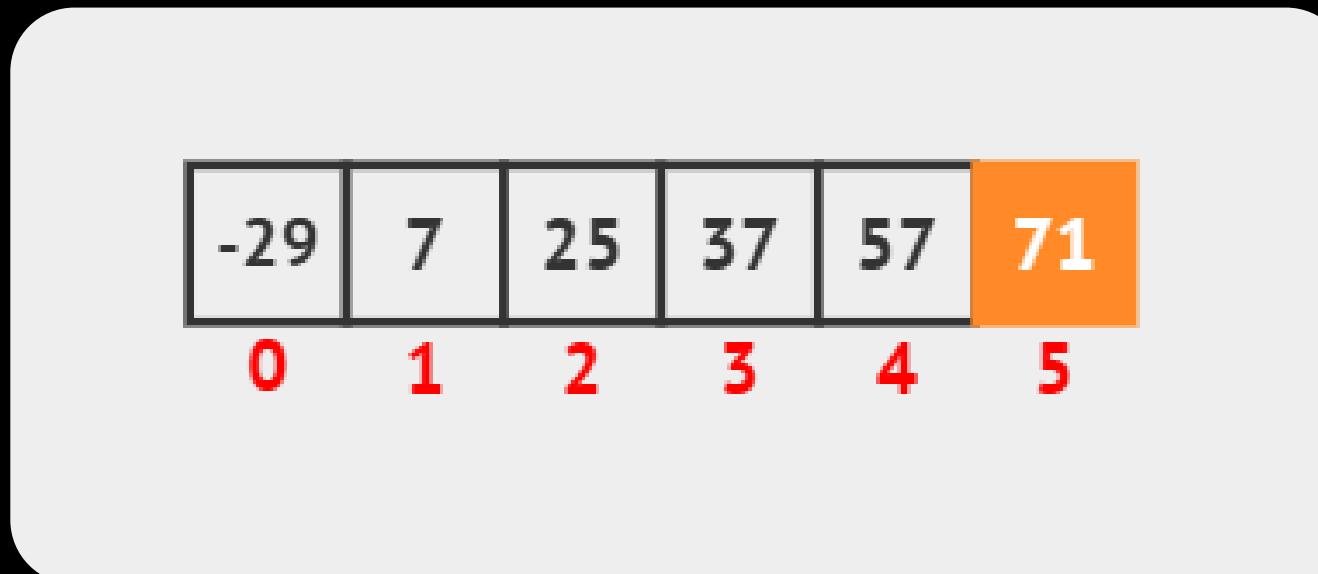
Step 5: Stop

Select 5 rank item
in sorted array

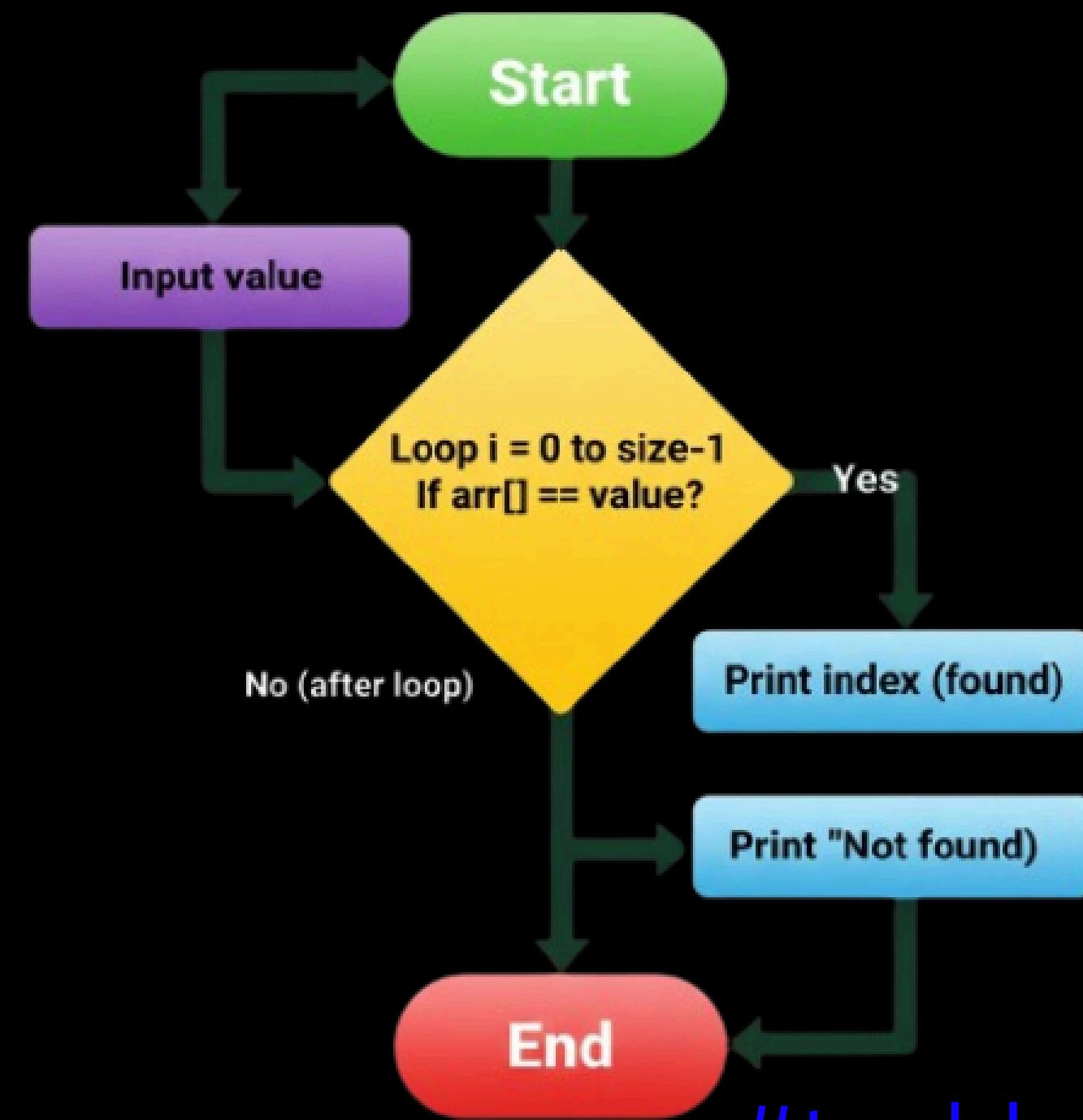
-29	7	25	37	57	71
0	1	2	3	4	5

SELECT Operation In Array

Flowchart

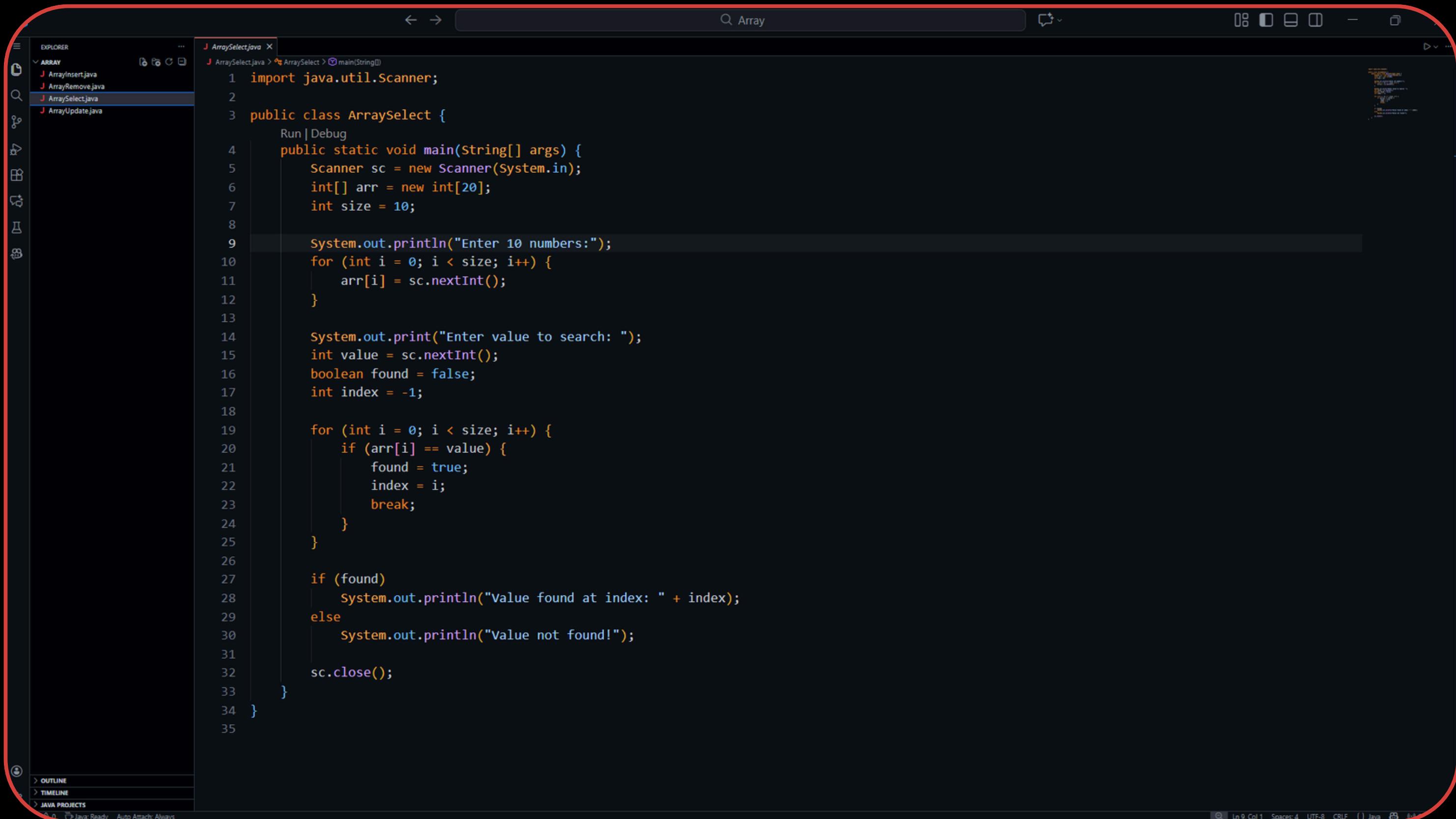


Select 5 rank item
in sorted array



#techbywebcoder

Program



The screenshot shows a Java development environment with a search results window titled "Array". The results list four files: "ArrayInsert.java", "ArrayRemove.java", "ArraySelect.java", and "ArrayUpdate.java". The "ArraySelect.java" file is selected and displayed in the main code editor. The code implements a search algorithm for an integer array using a Scanner to input values and System.out.println to output results.

```
import java.util.Scanner;

public class ArraySelect {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[20];
        int size = 10;

        System.out.println("Enter 10 numbers:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter value to search: ");
        int value = sc.nextInt();
        boolean found = false;
        int index = -1;

        for (int i = 0; i < size; i++) {
            if (arr[i] == value) {
                found = true;
                index = i;
                break;
            }
        }

        if (found)
            System.out.println("Value found at index: " + index);
        else
            System.out.println("Value not found!");

        sc.close();
    }
}
```

Output

```
C:\Windows\System32\cmd.e × + ▾ - ⓘ  
D:\system\DSA\1. Array Operation\Array>javac ArraySelect.java  
D:\system\DSA\1. Array Operation\Array>java ArraySelect  
Enter 10 Numbers:  
10  
20  
30  
40  
50  
60  
70  
80  
90  
100  
Enter Value To Search :40  
Value Found At Index:3  
D:\system\DSA\1. Array Operation\Array>
```



4.

UPDATE Operation In Array

#techbywebcoder

UPDATE Operation In Array



Algorithm

Step 1: Start

Step 2: Input position to update

Step 3: If position < 0 OR position ≥ size

 Print "Invalid position" and stop

Step 4: Input new value

Step 5: Set arr[position] = new value

Step 6: Stop

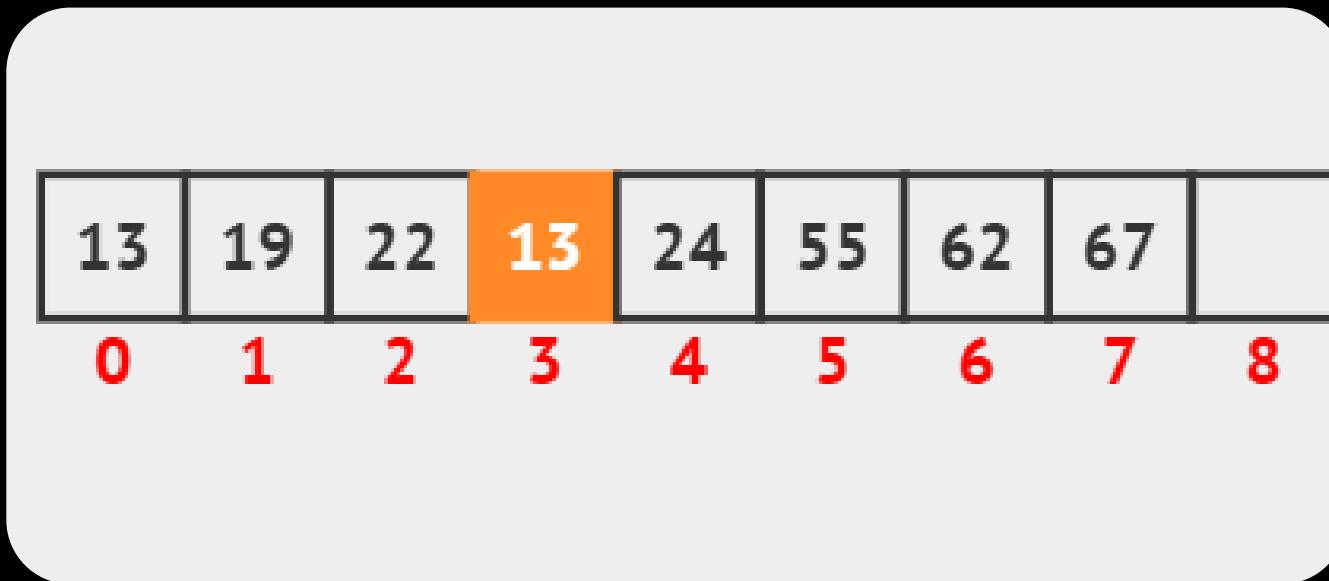
Update array
element at chosen
index

13	19	22	13	24	55	62	67	
0	1	2	3	4	5	6	7	8

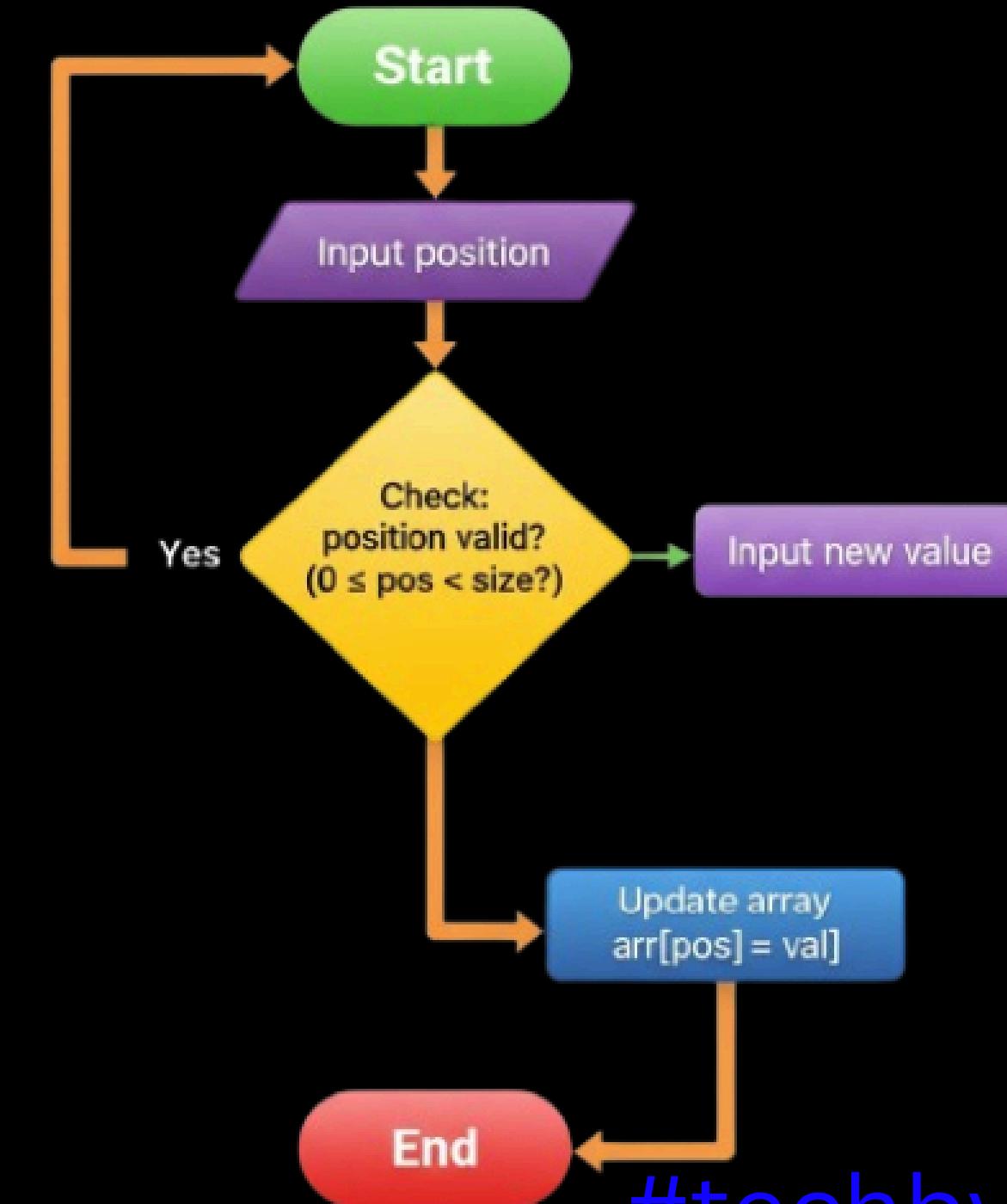
#techbywebcoder

UPDATE Operation In Array

Flowchart

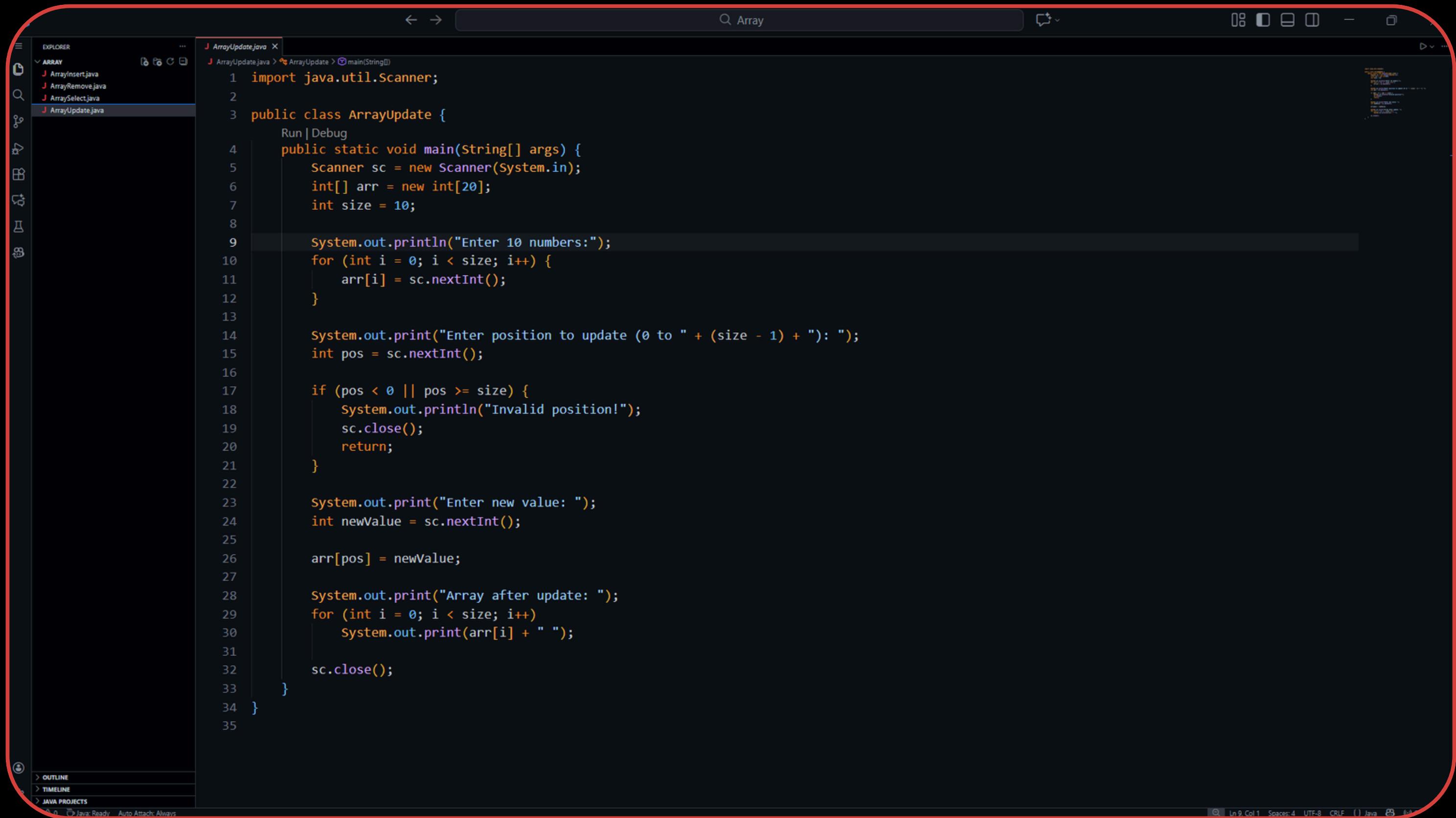


Update array
element at chosen
index



#techbywebcoder

Program



A screenshot of a Java IDE interface, specifically showing the code for an array update program. The code is written in Java and uses the Scanner class to interact with the user.

```
import java.util.Scanner;
public class ArrayUpdate {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[20];
        int size = 10;

        System.out.println("Enter 10 numbers:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter position to update (0 to " + (size - 1) + "): ");
        int pos = sc.nextInt();

        if (pos < 0 || pos >= size) {
            System.out.println("Invalid position!");
            sc.close();
            return;
        }

        System.out.print("Enter new value: ");
        int newValue = sc.nextInt();

        arr[pos] = newValue;

        System.out.print("Array after update: ");
        for (int i = 0; i < size; i++)
            System.out.print(arr[i] + " ");
        sc.close();
    }
}
```

The code defines a class named `ArrayUpdate` with a `main` method. It uses a `Scanner` object to read input from the console. The program prompts the user to enter 10 integers and stores them in an array of size 20. Then, it asks for a position to update and a new value. If the position is invalid, it prints an error message and exits. Otherwise, it updates the array at the specified index and prints the updated array.

Output

```
C:\Windows\System32\cmd.e × + ▾ - ☒

D:\system\DSA\1. Array Operation\Array>javac ArrayUpdate.java

D:\system\DSA\1. Array Operation\Array>java ArrayUpdate
Enter 10 Numbers:
10
20
30
40
50
60
70
80
90
100
Enter Position To Update (0 to 9): 6
Enter New Value: 500
Array After Update:
10 20 30 40 50 60 500 80 90 100
D:\system\DSA\1. Array Operation\Array>
```

Thank you



The banner features a colorful gradient background transitioning from purple to orange. On the left is a cartoon character with black hair and blue eyes wearing a headband with code symbols. In the center, the text "TECH BY WEBCODER" is written in large, bold, yellow letters. Below it, the tagline "REFACTORING IS LIKE DENTAL HYGIENE FOR SOFTWARE ENGINEERS" is displayed in white. To the right of the text are three icons: a blue square with "S", a red square with "3", and an orange square with "JS". The overall design is modern and tech-oriented.

Tech By WebCoder

@TechByWebCoder · 1.85k subscribers · 224 videos

Hello Guys..... [...more](#)

instagram.com/tech_bywebcoder?igsh=YXByM3kwbmZqOGkx and 5 more links

 Subscribed

Subscribe For More !