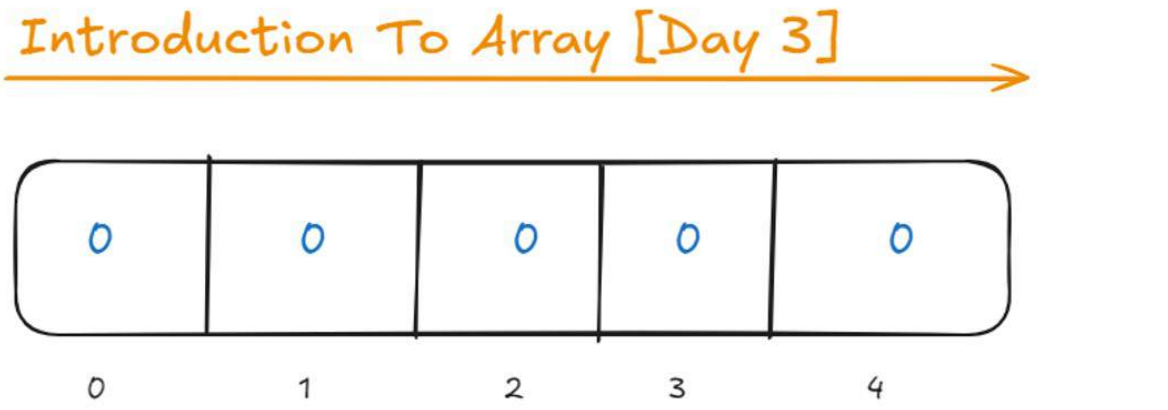
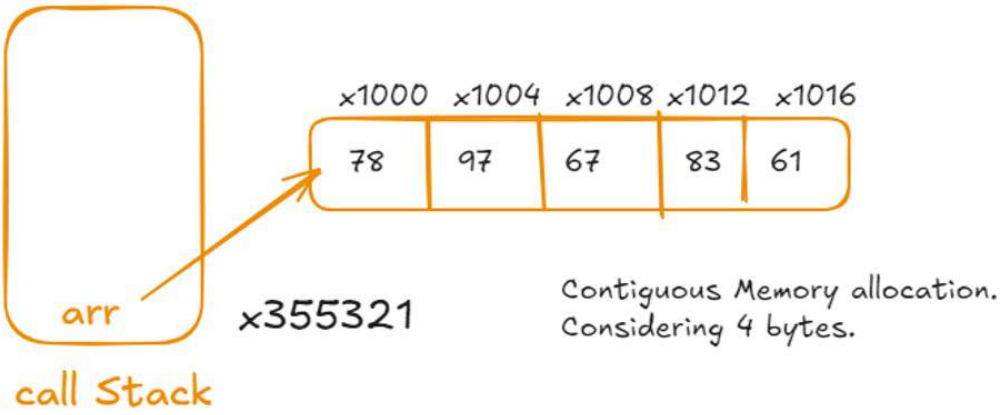
Introduction To Array [Day 3]

04 September 2024 17:11



- Indexing starts from 0.
- Fixed Size Array.
- Stores Homogeneous Element.

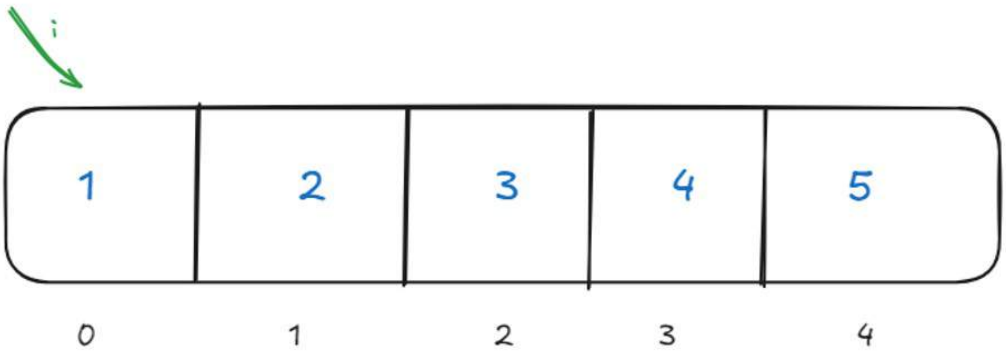
```
type[]  arrayName = new type[size];
int arr = new int[5];
```




```
// Initialize with Value:
   int[] arr = {1,2,3,4,5};

   arr[3] = 4 // Accessing the array

   arr[0] = 10 ;  // Modifying the array.

   Modified Array : [10 , 2 , 3, 4 ,5];
```

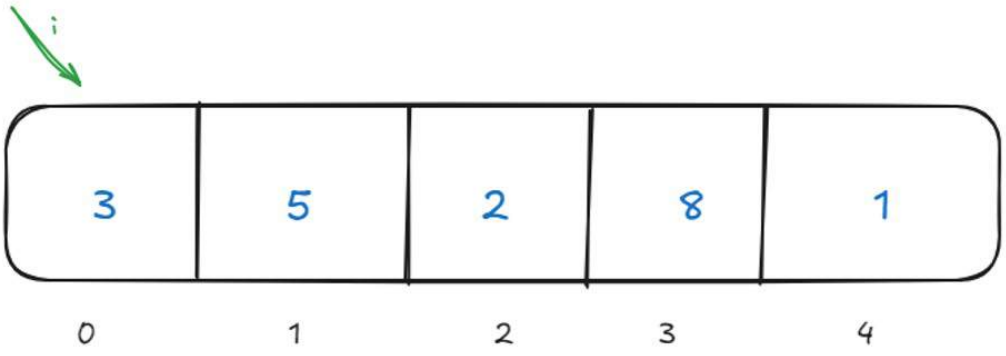
Modified Array : [10 , 2 , 3, 4 ,5];

Write a program to traverse the array.
int numbers = {1,2,3,4,5}.




```
public class ArrayTraversal {
    public static void main(String[] args) {
        int[] numbers = {1,2,3,4,5};
        for(int i = 0; i< numbers.length ; i++){
            System.out.println(numbers[i]);
        }
    }
}
```

Write an program to find largest element in an array.



Pseudo :

start max = arr[0];
keep iterating through the array.

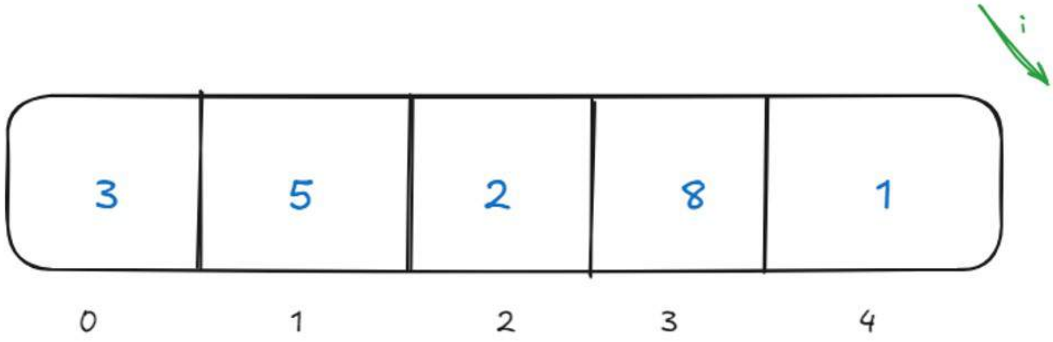
keep iterating through the array.
if(arr[i] > max):
update the max;
after loops end , Display max.

```
public class MaxElement {
    public static void main(String[] args) {
        int[] arr = {3,5,2,8,1};
        int max = arr[0];
        for(int i = 1 ; i < arr.length; i++){
            if(arr[i] > max){
                max = arr[i];
            }
        }
        System.out.println("The Maximum Element is: " + max);
    }
}
```

Write a program to sum all elements in an array.

Eg = arr = [1,2,3,4,5]

output : 15.



sum = 19

sum += arr[i];

```
public class SumArray {
    public static void main(String[] args) {
        int[] arr = {1,2,3,4,5};
        int sum =0;
        for(int i = 0; i<arr.length ; i++){
```

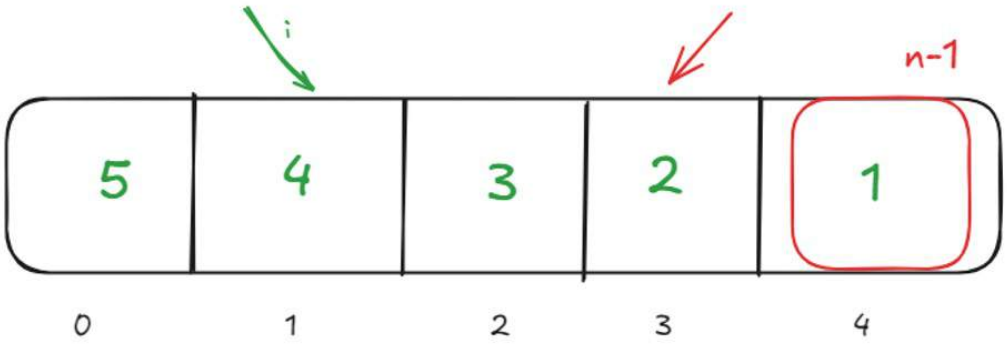
```
        int[] arr = {1,2,3,4,5};
        int sum =0;
        for(int i = 0; i<arr.length ; i++){
            sum += arr[i];
        }
        System.out.println("Sum of all elements is : " + sum);
    }
}
```

Write a program To Reverse an Array.

arr = [1,2,3,4,5]

output = [5,4,3,2,1]

pseudo -
Iterate through an array till n/2;
swap arr[i] with arr[n-1-i]



0th iteration - i = 0
- 1 is swapped with 5.

1st iteration i = 1;

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

        int[] arr = {1,2,3,4,5};
        int n = arr.length;
        for(int i = 0 ; i< n/2; i++){
            int temp = arr[i];
            arr[i] = arr[n-1-i];
            arr[n-1-i] = temp;
        }

        // Traverse the reverse array to print
```

→ O(LogN)

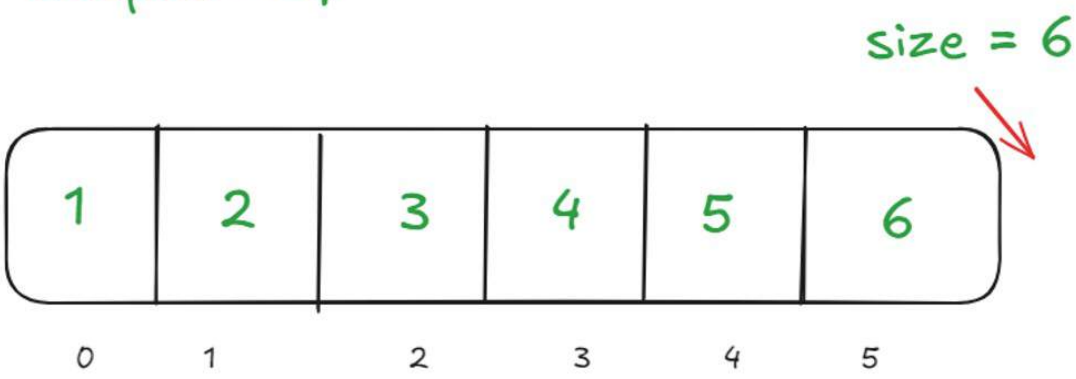
```
        // Traverse the reverse array to print
          the elements.
        for(int i =0 ; i<n; i++){
            System.out.print(arr[i] + " ");
        }
    }
}
```

O(N)

Time Complexity = O(N + LogN);
Space Complexity - O(N)

Write a program to Count Even Numbers in an array.

arr = [1,2,3,4,5,6]
output = 3;



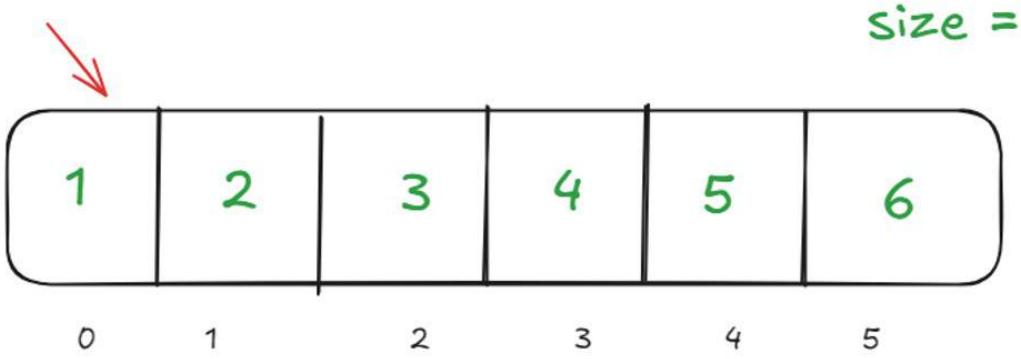
count = 3;

```
public class CountEvens {
    public static void main(String[] args) {
        int[] arr = {1,2,3,4,5,6};
        int count = 0;
        for(int i = 0 ; i<arr.length; i++){
            if(arr[i] % 2 == 0){
                count++;
            }
        }
        System.out.println("Number of Even Numbers : " + count);
    }
}
```

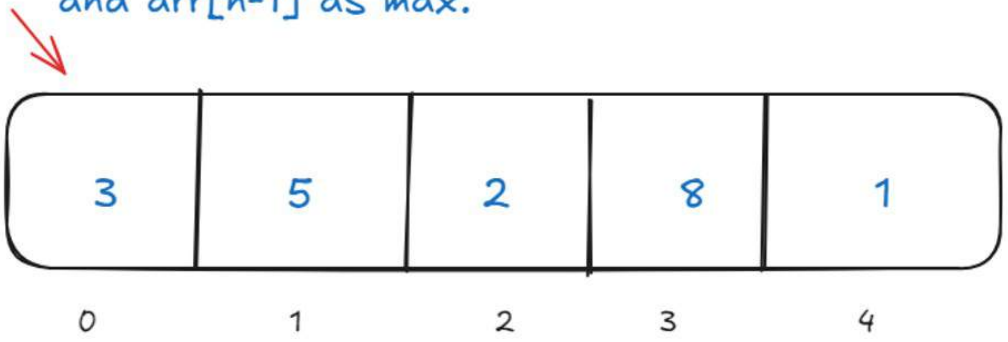
Write a program to find the smallest element in the array

Write a program to find the smallest element in the array.

size = 6



if it is sorted in asscending return arr[0] as min.
and arr[n-1] as max.



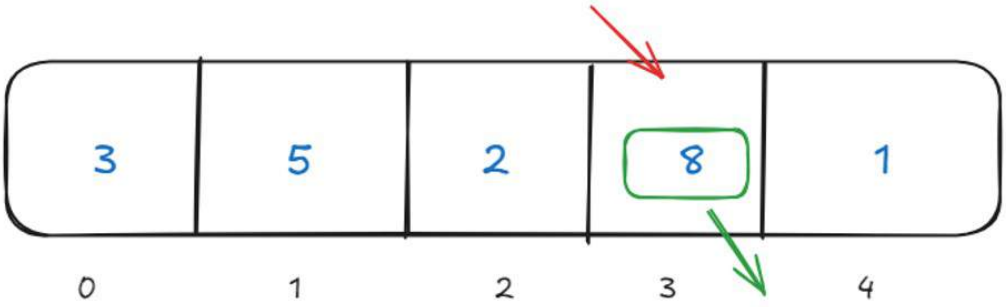
int num = arr[0];

run a loop till n - 1
and compare with arr[i].
if found any min value other than that.
update min .
After loop end Display min.

```
public class MinElement {
    public static void main(String[] args) {
        int[] arr = {3,5,2,8,1};

        int min = arr[0];
        for(int i = 1; i<arr.length ; i++){
            if(arr[i] < min){
                min = arr[i];
            }
        }
        System.out.println("Min Element in an array : " + min);
    }
```

```
        System.out.println("Min Element in an array : " + min);
    }
}
```

Write a program to check Specific Value from an array.
[3,5,2,8,1];
valueToSearch = 8;
if not found , print a message - Value Not Found.



boolean flag = false;

flag = true;

```
public class FindValue {
    public static void main(String[] args) {
        int[] arr = {3,5,2,8,1};
        int value = 8;
        boolean found = false;

        for(int i =0; i< arr.length ; i++){
            if(arr[i] == value){
                found = true;
                break;
            }
        }
        if(found){
            System.out.println("Value found");
        }else{
            System.out.println("Value not found");
        }
    }
}
```

Problem 1: Contains Duplicate

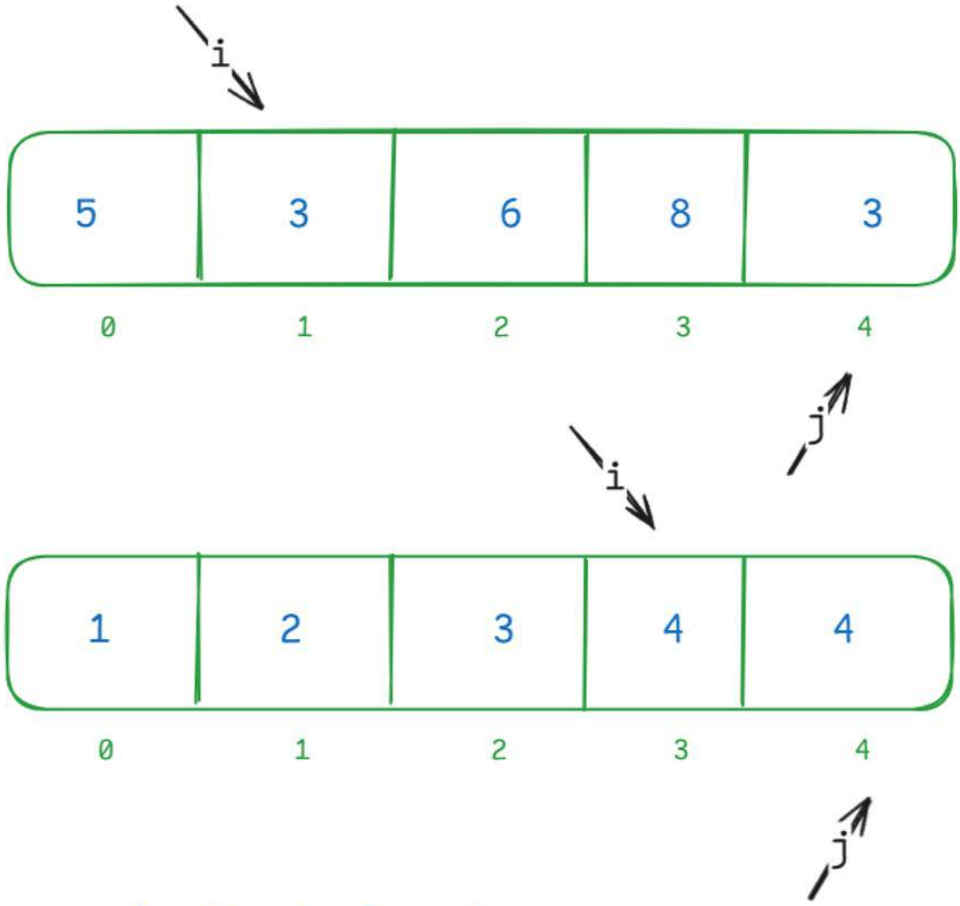
Problem 1: Contains Duplicate

Problem Statement:

Write a program to check if an array contains any duplicate elements. If duplicates are found, print "Duplicates found," otherwise print "No duplicates."

Example:

- Input: `5, 3, 6, 8, 3`
- Output: `Duplicates found`
- Input: `1, 2, 3, 4, 5`
- Output: `No duplicates`




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

        System.out.println("Enter the size of an array: " );
```

```
        System.out.println("Enter the size of an array: " );

        int n = scn.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the Element of an array:");
        for(int i =0 ; i<n ; i++){
            arr[i] = scn.nextInt();
        }

        // check for duplicate.
        boolean foundDuplicate = false;
        for(int i=0 ; i< n - 1; i++){
            for(int j = i+1; j<n ; j++){
                if(arr[i] == arr[j]){
                    foundDuplicate = true;
                    break;
                }
            }
        }

        if(foundDuplicate){
            System.out.println("Duplicates Found");
        }else{
            System.out.println("No Duplicates");
        }
    }
}
```

O(N)

O(N^2)

O(1)

Time Complexity = O(N^2) + O(N) + O(1)

Space Complexity - O(N)

Problem 2: Array Operations

Problem Statement:

Write a program to perform basic operations on an array such as finding the sum, product, and average of all elements.

Example:

- Input: `1, 2, 3, 4, 5`
- Output:
  Sum = 15
  Product = 120
  Average = 3.0

```
Average = 3.0
```

```java
import java.util.Scanner;
public class ArrayOpearions {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the size of an array: ");
        int n = scn.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of an array:");
        for(int i=0 ; i <n; i++){
            arr[i] = scn.nextInt();
        }
        int sum =0;
        int product = 1;
        for(int i=0; i<n; i++){
            sum += arr[i];
            product *= arr[i];
        }
        double average = (double) sum / n;

        System.out.println(sum);
        System.out.println(product);
        System.out.println(average);
    }
}
```

O(N)

O(1)

O(N)

O(1)

Time Complexity : O(N);
Space Complexity : O(N)

Problem 3: Second Largest Element From Array

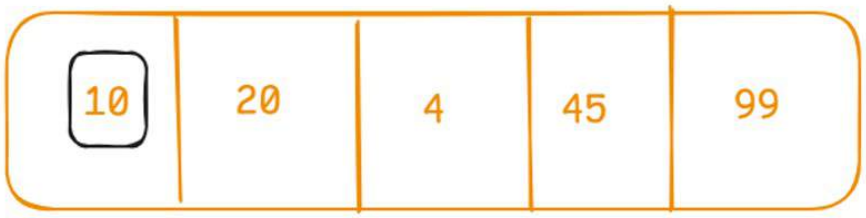
Problem Statement:

Write a program to find the second largest element in an array.

Example:

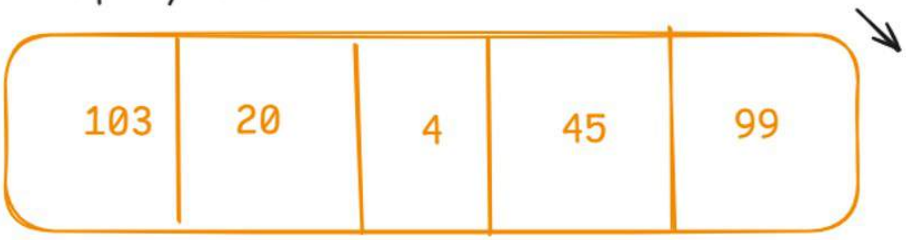
- Input: `10, 20, 4, 45, 99`
- Output: `45`

- Output: `45`



firstmax = 90
secondmax = 45

```
if(arr[i] > firstmax){
    secondmax = firstmax;
    firstmax = arr[i];
}else if(arr[i] > secondmax && arr[i] < firstmax){
    secondmax = arr[i];
}
display secondmax
```



1stmax = 103
2nd = 99

```
import java.util.Scanner;
public class SecondLargestElement {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the size of an
array:");

        int n = scn.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the Element of an
array");

        for(int i = 0; i<n; i++){
            arr[i] = scn.nextInt();
        }
```

space - O(N)

O(1)

O(N)

```
    arr[i] = scn.nextInt();
}
```

O(N)

```
int firstMax = Integer.MIN_VALUE;
int secondMax = Integer.MIN_VALUE;
```

O(1)

```
for(int i =0; i<n; i++){
    if(arr[i] > firstMax){
        secondMax = firstMax;
        firstMax = arr[i];
    }else if(arr[i] > secondMax && arr[i] <
      firstMax){
        secondMax = arr[i];
    }
}
```

O(N)

```
        System.out.println("Second largest Element is
:" + secondMax);
    }
}
```

Time Complexity - O(N)
Space Complexity - O(N)

Problem 4: Array Swaps

Problem Statement:

Write a program to swap the first and last elements of an array.

Example:

- Input: `7, 9, 5, 3, 6`
- Output: `6, 9, 5, 3, 7`

Use Temp variable and swap the first & last element

Use Temp variable and swap the first & last element

```java
import java.util.Scanner;
public class ArraySwap {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the size of an array:");

        int n = scn.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the Element of an array");

        for(int i = 0; i<n; i++){
            arr[i] = scn.nextInt();
        }

        // Swap array first & last Element
        int temp = arr[0];
        arr[0] = arr[n-1];
        arr[n-1] = temp;

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

O(1)

O(N)

O(1)

O(N)

Time Complexity : O(N)
Space Complexity : O(N)

Problem: Span of an Array

Given an array of integers, determine the "span" of the array. The span of an array is defined as the difference between the maximum and minimum values within the array.

Example 1:

Input:

arr = [15, 30, 40, 4, 11, 9]

Output:

36

36

Explanation:

The maximum value in the array is 40 and the minimum value is 4. Therefore, the span is 40 - 4 = 36.

Example 2:

Input:

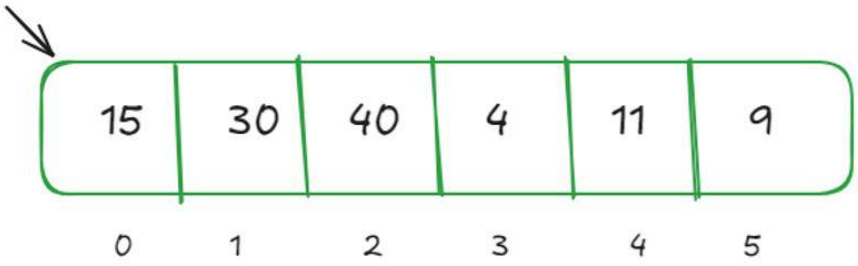
arr = [5, 3, 1, 9, 7]

Output:

8

Explanation:

The maximum value in the array is 9 and the minimum value is 1. Therefore, the span is 9 - 1 = 8.



max = 40
min = 4
span = max - min = 40-4 = 36.

```
import java.util.Scanner;
public class SpanOfArray {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the size of an array:");

        int n = scn.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the Element of an array");

        for(int i = 0; i<n; i++){
            arr[i] = scn.nextInt();
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;
        for(int i = 0 ; i<n; i++){
            if(arr[i] > max){
                max = arr[i];
            }
```

```
            if(arr[i] > max){
                max = arr[i];
            }
            if(arr[i] < min){
                min = arr[i];
            }
        }
        int span = max - min;
        System.out.println(span);


    }
}
```

Time & Space - O(N)

Attendance Code: 56484989