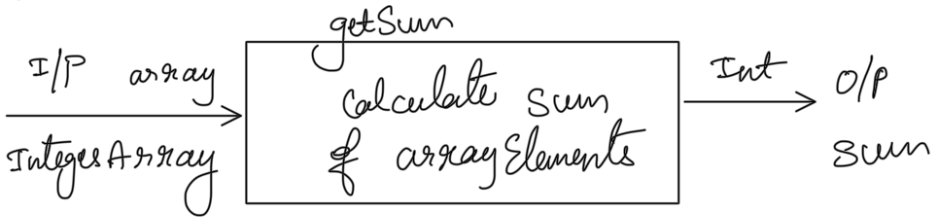
JAVA HANDS ON PRACTICE

Day 23
02/08/25

Sum of all elements in Integer Array

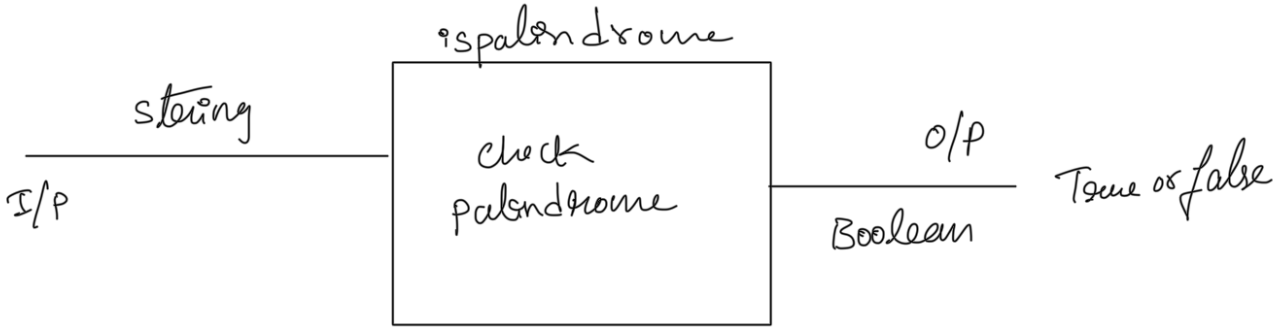


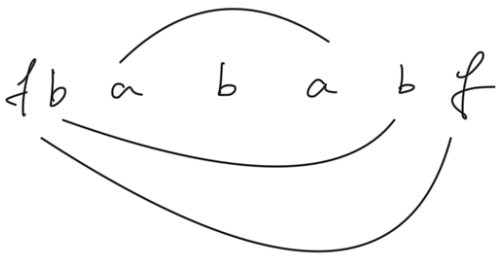
1. Read 1 value at a time (Iteration - loop)
2. Sum → 0
3. add Sum returns Sum at End

```
133     }
134
135     public static int getArrSum(int[] arr){
136         int sum=0;
137         for(int i:arr)sum+=i;
138         return sum;
139     }
140
141     blic static void invoke_getArrSum(){
142         int arr [] = {2,4,5,6,7};
143         System.out.println("Array Elements [2,4,5,6,7] sum : "+ getArrSum(arr));
144     }
145
    Run | Debug | Run main | Debug main
```

Prints the Sum of Array Elements

Q) Write a function to get a string Palindrome or Not



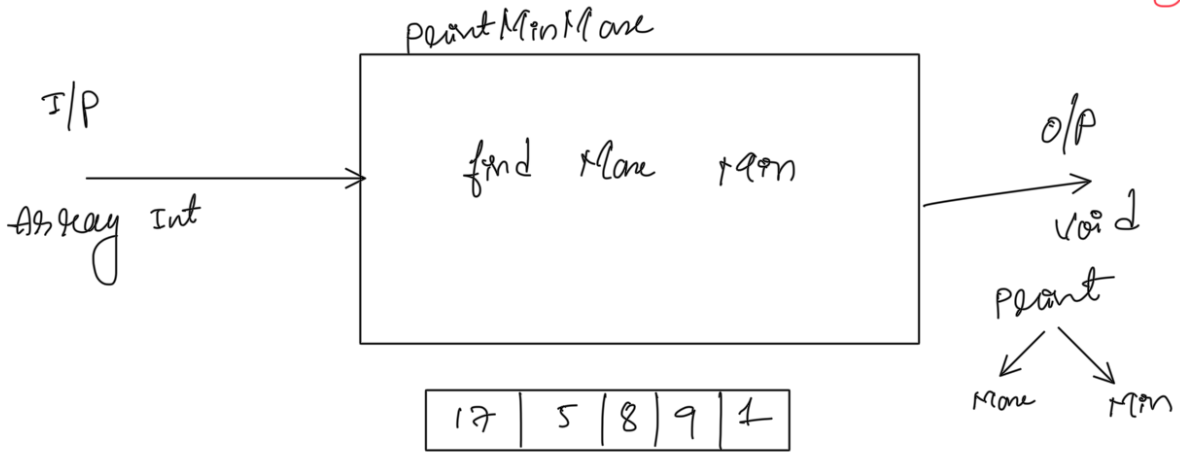


```java
public static boolean isPalindrome(String s){
    int st=0; int end = s.length()-1;
    while (st<end){
        if(s.charAt(st)!=s.charAt(end))return false;
        st++;
        end--;
    }

    return true;
}

public static void invoke_isPalindrome(){
    System.out.println("String ''malayalam'' is palindrome : "+ isPalindrome(s:"malayalam"));
}
```
Run | Debug | Run main | Debug main

8) function to print MinMax function in an Array.



1. Min Max
2. Loop traversal
3. print MinMax

```java
public static void printMinMax(int[] arr){
    int max = Integer.MIN_VALUE;
    int min = Integer.MAX_VALUE;
    for(int i: arr){
        if(max<i)max=i;
        if(min>i)min=i;
    }
    System.out.println("Min value in array is "+ min );
    System.out.println("Max value in array is "+ max );
}
public static void invoke_printMinMax(){
    int arr[] = {3,5,7,-2,6,9,2};
    System.out.println(Arrays.toString(arr));
    printMinMax(arr);
}
```
Run | Debug | Run main | Debug main

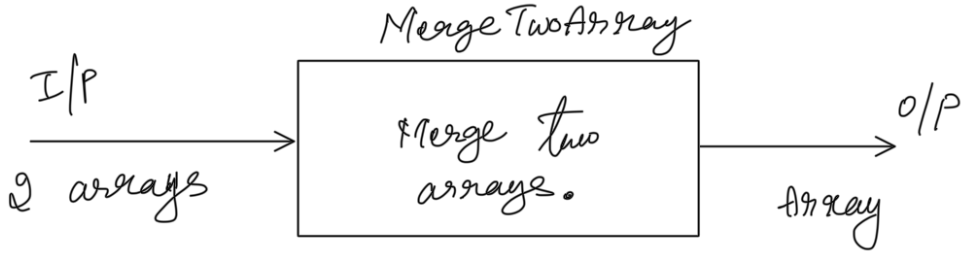
function to Search In a Sorted Integer Array.

Binary Search

Continue Boolean or Int


```java
177
178     public static int binarySearch(int[] arr,int key){
179         int st = 0 ;
180         int  end = arr.length-1;
181         while (st<=end) {
182             int mid = st+((end-st)/2);
183             if(arr[mid]==key)return mid;
184             if(arr[mid]>key)end=mid-1;
185             else st = mid+1;
186         }
187         return -1;
188     }
189
190     public static void invoke_binarySearch(){
191         int arr[]= {1,4,6,8,12,32,67,90};
192         System.out.println("Key 32 found at position "+ (binarySearch(arr, key:32)+1));
193     }
    Run | Debug | Run main | Debug main
```

8) Function to Merge two arrays and Combine output




```java
194
195     public static int[] mergeArrays(int[] arr1, int[] arr2) {
196         int n1 = arr1.length;
197         int n2 = arr2.length;
198         int[] merged = new int[n1 + n2];
199         for (int i = 0; i < n1; i++) {
200             merged[i] = arr1[i];
201         }
202         for (int i = 0; i < n2; i++) {
203             merged[n1 + i] = arr2[i];
204         }
205         return merged;
206     }
207     public static void invoke_mergeArrays(){
208         int[] array1 = {1, 3, 5, 7};
209         int[] array2 = {2, 4, 6, 8};
210         System.out.println("arr1 : "+ Arrays.toString(array1));
211         System.out.println("arr2 : "+ Arrays.toString(array2));
212         int[] mergedArray = mergeArrays(array1, array2);
213         System.out.print(s:"Merged Array: ");
214         System.out.println(Arrays.toString(mergedArray));
215     }
```

Flowcharts and Dry Run of Java Programs

Veeresh K

August 2, 2025

1. Sum of Array Elements

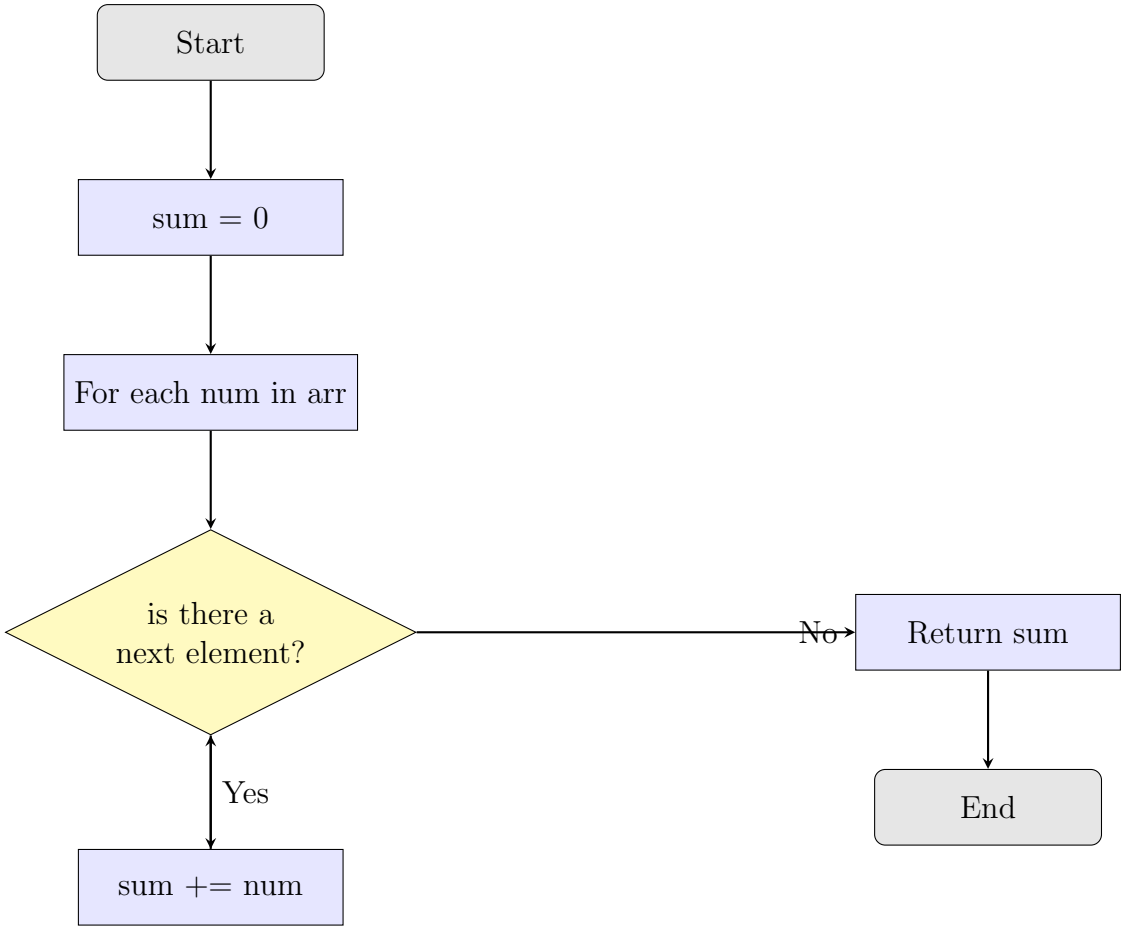
Code:

```java
public static int getSum(int[] arr) {
    int sum = 0;
    for (int num : arr) {
        sum += num;
    }
    return sum;
}
```

Dry Run:

- Input: {1, 2, 3}
- sum = 0
- sum = 0 + 1 = 1
- sum = 1 + 2 = 3
- sum = 3 + 3 = 6
- Output: 6

Flowchart:


Start
sum = 0
For each num in arr
is there a
next element?
No
Return sum
End
Yes
sum += num

2. Binary Search

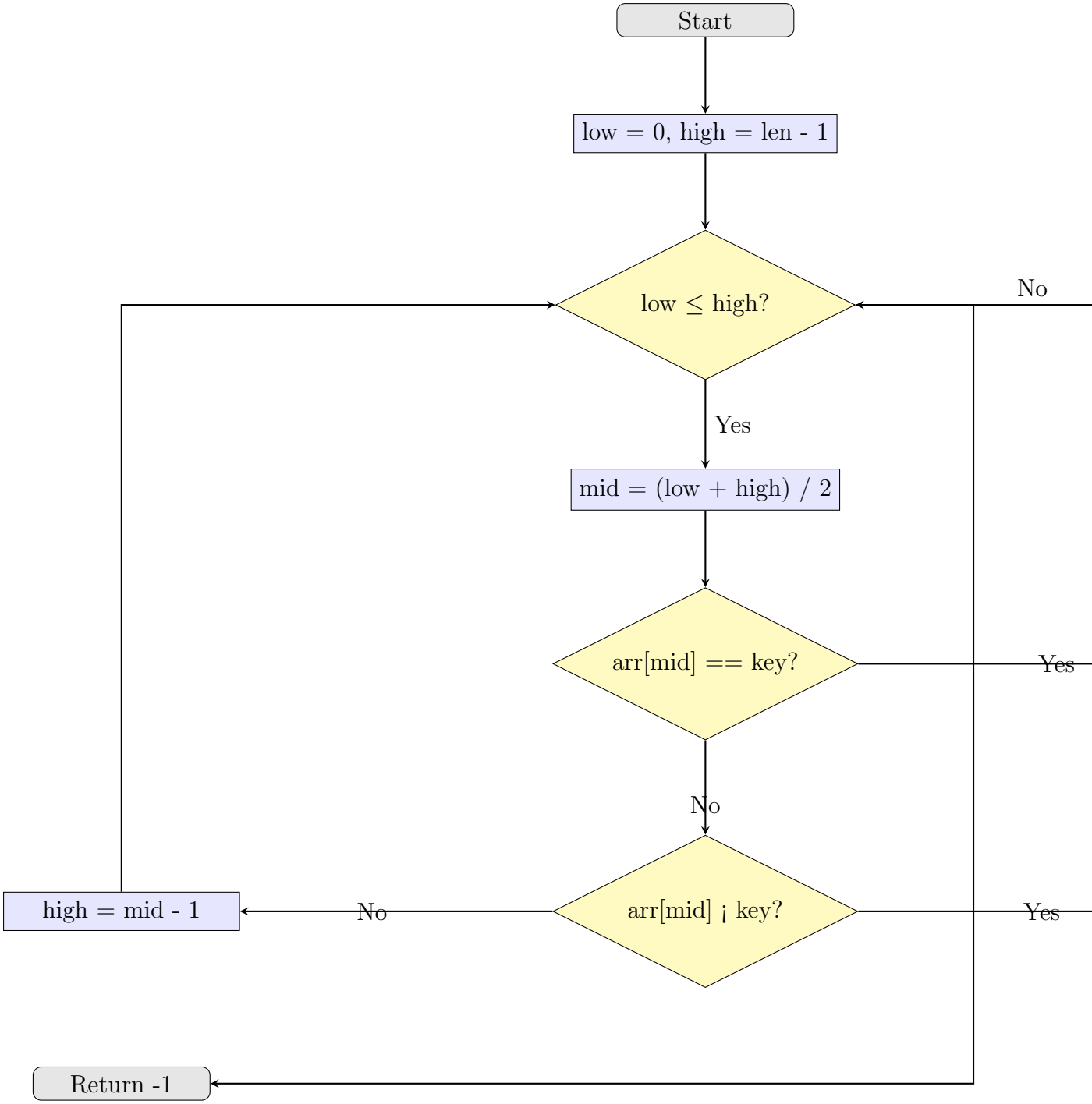
Code:

```
public static int binarySearch(int[] arr, int key) {
    int low = 0, high = arr.length - 1;
    while (low <= high) {
        int mid = low + (high - low) / 2;
        if (arr[mid] == key) {
            return mid;
        } else if (arr[mid] < key) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return -1;
}
```

Dry Run:

- Input: arr = {1,3,5,7}, key = 5
- low = 0, high = 3
- **Iteration 1:** mid = (0+3)/2 = 1. arr[1]=3 ¡ 5 ⇒ low = 1 + 1 = 2
- **Iteration 2:** low = 2, high = 3. mid = (2+3)/2 = 2. arr[2]=5 == 5 ⇒ return 2
- Output: 2

Flowchart:


Start
low = 0, high = len - 1
low ≤ high?
No
Yes
mid = (low + high) / 2
arr[mid] == key?
Yes
No
arr[mid] ¡ key?
Yes
No
high = mid - 1
Return -1

3. Palindrome String Check

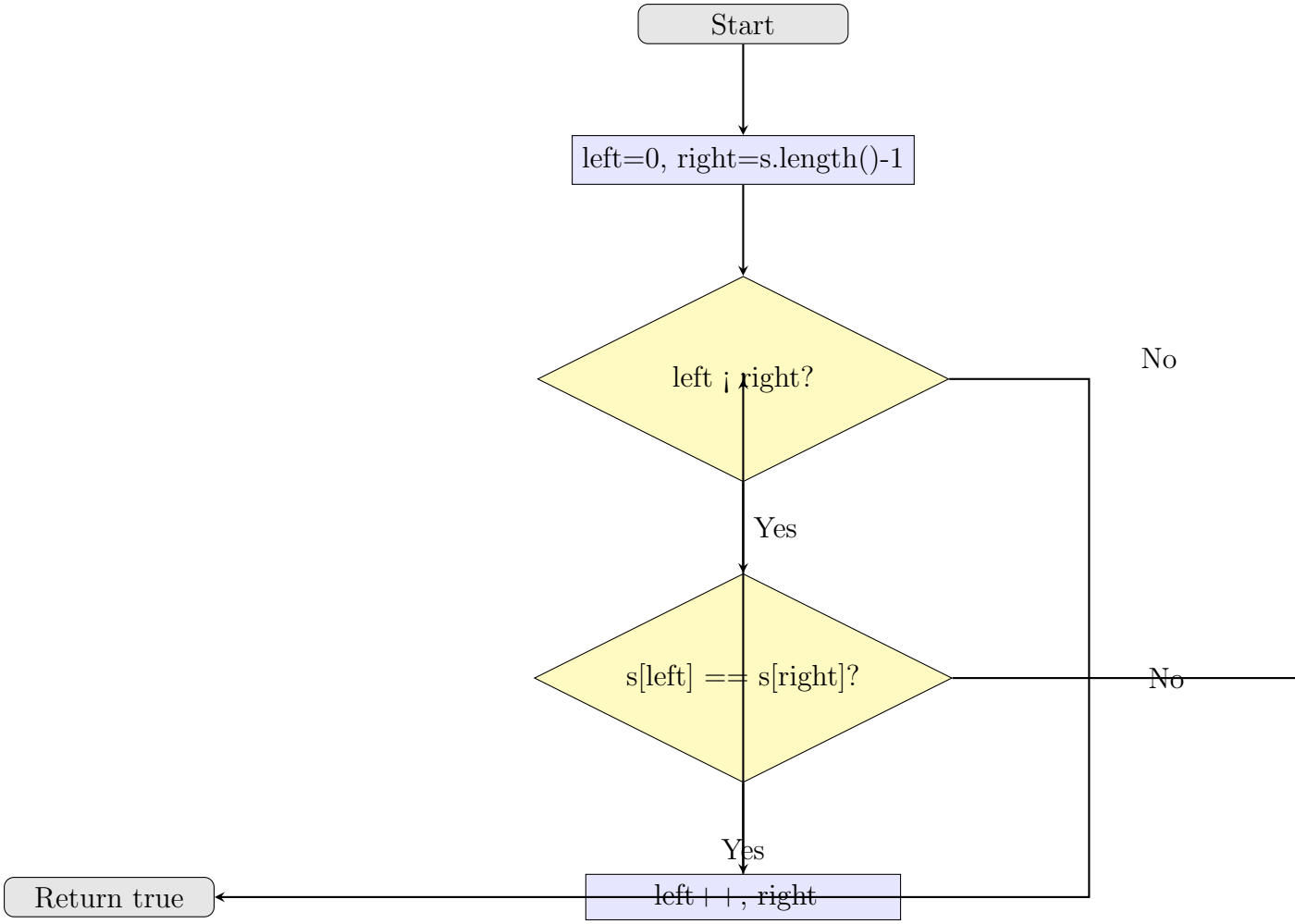
Code:

```
public static boolean isPalindrome(String s) {
    int left = 0, right = s.length() - 1;
    while (left < right) {
        if (s.charAt(left) != s.charAt(right)) {
            return false;
        }
        left++;
        right--;
    }
    return true;
}
```

Dry Run:

- Input: ”madam”
- **Iteration 1:** left = 0, right = 4. ’m’ == ’m’. left++, right–. left=1, right=3.
- **Iteration 2:** left = 1, right = 3. ’a’ == ’a’. left++, right–. left=2, right=2.
- **Loop ends:** ‘left‘ is no longer less than ‘right‘.
- Output: **true**

Flowchart:


Start
left=0, right=s.length()-1
left ¡ right?
No
Yes
s[left] == s[right]?
No
Yes
left++, right
Return true