# Arrays Interview Programs

1. **Write a java program to perform sum of array elements?**

**Input:**

**5 9 2 7 6**

**Output:**

**29**

```
class SumOfArray
{
	public static void main(String[] args)
	{
		int[] arr={5,9,2,7,6};

		int sum=0;

		//for each loop
		for(int i:arr)
		{
			sum+=i;
		}

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

2. **Write a java program to display array elements in reverse order?**

**Input:**

**5 9 2 7 6**

**Output:**

**6 7 2 9 5**

```
class ReverseArray
{
	public static void main(String[] args)
	{
		int[] arr={5,9,2,7,6};

		//reading reverse
		for(int i=arr.length-1;i>=0;i--)
		{
			System.out.print(arr[i]+" ");
		}

	}
}
```

3. **Write a java program to display even elements from given array?**

**Input:**

**3 4 7 9 2 6 1**

**Output:**

**4 2 6**

```
class ArrayEvenElements
{
	public static void main(String[] args)
	{
		int[] arr={3,4,7,9,2,6,1};

		//for each loop
		for(int i:arr)
		{
			if(i%2==0)
			{
				System.out.print(i+" ");
			}
		}
	}
}
```

4. **Write a java program to display odd elements from given array?**

**Input:**

**3 4 7 9 2 6 1**

**Output:**

**3 7 9 1**

```
class ArrayOddElements
{
	public static void main(String[] args)
	{
		int[] arr={3,4,7,9,2,6,1};

		//for each loop
		for(int i:arr)
		{
			if(i%2!=0)
			{
				System.out.print(i+" ");
			}
		}
	}
}
```

5. **Write a java program to count number of even and odd elements from given array?**

**Input:**

**3 4 7 9 2 6 1**

**Output:**

**Even elements : 3**
**Odd elements : 4**

```
class EvenOddElements
{
	public static void main(String[] args)
	{
		int[] arr={3,4,7,9,2,6,1};

		int even=0;
		int odd=0;
		//for each loop
		for(int i:arr)
		{
			if(i%2==0)
			{
				even++;
			}
			else
			{
				odd++;
			}
		}
		System.out.println("Even Elements :"+even);
		System.out.println("Odd Elements :"+odd);
	}
}
```

6. **Write a java program to display prime elements from given array?**

**Input:**

5 9 3 11 15 7 14

**Output:**

5 3 11 7

```
class ArrayPrimeElements
{
	public static void main(String[] args)
	{
		int[] arr={5,9,3,11,15,7,14};

		//for each loop
		for(int n:arr)
		{
			boolean flag=true;
			for(int i=2;i<=n/2;i++)
			{
				if(n%i==0)
				{
					flag=false;
					break;
				}
			}
			if(flag==true)
				System.out.print(n+" ");
		}
	}
}
```

7. **Write a java program to display array elements in sorting order?**

**Input:**

5 9 3 11 15 7 14

**Output:**

3 5 7 9 11 14 15

```
class ArraySortingOrder
{
	public static void main(String[] args)
	{
		int[] arr={5,9,3,11,15,7,14};

		//ascending logic
		for(int i=0;i<arr.length;i++)
		{
			for(int j=0;j<arr.length;j++)
			{
				if(arr[i]<arr[j])
				{
					int temp=arr[i];
					arr[i]=arr[j];
					arr[j]=temp;
				}
			}
		}

		//display elements
		for(int i:arr)
		{
			System.out.print(i+" ");
		}

	}
}
```

8. **Write a java program to display array elements in descending order?**

**Input:**

5 9 3 11 15 7 14

**Output:**

15 14 11 9 7 5 3

```
class ArrayDescendingOrder
{
	public static void main(String[] args)
	{
		int[] arr={5,9,3,11,15,7,14};

		//descending logic
		for(int i=0;i<arr.length;i++)
		{
			for(int j=0;j<arr.length;j++)
			{
				if(arr[i]>arr[j])
				{
					int temp=arr[i];
					arr[i]=arr[j];
					arr[j]=temp;
				}
			}
		}

		//display elements
		for(int i:arr)
		{
			System.out.print(i+" ");
		}

	}
}
```

9. **Write a java program to display highest element from given array?**

**Input:**

**7 2 4 9 1 6 3**

**Output:**

**9**

```
class HighestElement
{
	public static void main(String[] args)
	{
		int[] arr={7,2,4,9,1,6,3};

		int big=arr[0];

		for(int i:arr)
		{
			if(i>big)
			{
				big=i;
			}
		}
		System.out.println(big);
	}
}
```

**10. Write a java program to display least element from given array?**

**Input:**

**7 2 4 9 1 6 3**

**Output:**

**1**

```
class LeastElement
{
	public static void main(String[] args)
	{
		int[] arr={7,2,4,9,1,6,3};

		int small=arr[0];

		for(int i:arr)
		{
			if(i<small)
			{
				small=i;
			}
		}
		System.out.println(small);
	}
}
```

**11. Write a java program to display three highest elements from given array?**

**Input:**

**7 2 4 9 1 6 3**

**Output:**

**9 7 6**

```
class ThreeHighestElements
{
	public static void main(String[] args)
	{
		int[] arr={7,2,4,9,1,6,3};

		int first=Integer.MIN_VALUE;
		int second=Integer.MIN_VALUE;
		int third=Integer.MIN_VALUE;

		//for each loop
		for(int i:arr)
		{
			if(i>first)
			{
				third=second;
				second=first;
				first=i;
			}
			else if(i>second)
			{
				third=second;
				second=i;
			}
			else if(i>third)
			{
				third=i;
			}
		}
		System.out.println(first+" "+second+" "+third);
	}
}
```

**12. Write a java program to display duplicate elements from given array?**

**Input:**

**3 6 1 2 3 9 4 4 7 6 10**

**Output:**

**3 6 4**

```
class DuplicateElements
{
	public static void main(String[] args)
	{
		int[] arr={3,6,1,2,3,9,4,4,7,6,10};

		//duplicate elements
		for(int i=0;i<arr.length;i++)
		{
			for(int j=i+1;j<arr.length;j++)
			{
				if(arr[i]==arr[j])
				{
					System.out.print(arr[i]+" ");
				}
			}
		}

	}
}
```

**13. Write a java program to display unique elements from given array?**

**Input:**

**3 6 1 2 3 9 4 4 7 6 10**

**Output:**

**1 2 9 7 10**

```
class UniqueElements
{
	public static void main(String[] args)
	{
		int[] arr={3,6,1,2,3,9,4,4,7,6,10};

		//unique elements
		for(int i=0;i<arr.length;i++)
		{
			int cnt=0;

			for(int j=0;j<arr.length;j++)
			{
				if(arr[i]==arr[j])
				{
					cnt++;
				}
			}
			if(cnt==1)
				System.out.print(arr[i]+" ");
		}

	}
}
```

**14. Write a java program to find out most repeating element from given array?**

**Input:**

**5 1 2 4 2 9 7 2 2 6 6 2**

**Output:**

**2 is repeating for 5 times**

```
class MostRepeatingElement
{
	public static void main(String[] args)
	{
		int[] arr={5,1,2,4,2,9,7,2,2,6,6,2};

		int maxCount=0;
		int element=0;

		for(int i=0;i<arr.length;i++)
		{
			int cnt=0;

			for(int j=0;j<arr.length;j++)
			{
				if(arr[i]==arr[j])
				{
					cnt++;
				}
			}

			if(cnt>maxCount)
			{
				maxCount=cnt;
				element=arr[i];
			}
		}
	System.out.println(element+" is repeating for "+maxCount+" times");
	}
}
```

**15. Write a java program to multiply two arrays?**

**Input:**

**arr1 = 5 3 2**
**arr2 = 1 4**

**output:**

**7448 (532*14)**

```
class MultiplyArray
{
	public static void main(String[] args)
	{
		int[] arr1 ={5,3,2};
		int[] arr2 ={1,4};

		//caller method
		int a=Integer.parseInt(arrayToString(arr1));
		int b=Integer.parseInt(arrayToString(arr2));
		System.out.println(a*b);
	}

	public static String arrayToString(int[] arr)
	{
		StringBuffer sb=new StringBuffer();
		for(int i:arr)
		{
			sb.append(i);
		}

		return sb.toString();
	}
}
```

**16. Write a java program to perform sum of two arrays and display them in third array?**

**Input:**

**1 6 2 9 4**

**8 2 4 1 3**

**Output:**

**9 8 6 10 7**

```
class  SumOfElements
{
	public static void main(String[] args)
	{
		int[] arr1={1,6,2,9,4};
		int[] arr2={8,2,4,1,3};

		int[] resArr=new int[arr1.length];

		for(int i=0;i<arr1.length && i<arr2.length;i++)
		{
			resArr[i]=arr1[i]+arr2[i];
		}

		//display
		for(int i:resArr)
		{
			System.out.print(i+" ");
		}

	}
}
```

**17. Write a java program to display 10 Fibonacci numbers?**

**output:**

```
0 1 1 2 3 5 8 13 21 34
```

```
class Fibonacci
{
	public static void main(String[] args)
	{
		int n=10;

		int[] fibonacci=new int[n];

		fibonacci[0]=0;
		fibonacci[1]=1;
		System.out.print(fibonacci[0]+" "+fibonacci[1]+" ");

		for(int i=2;i<fibonacci.length;i++)
		{
			fibonacci[i]= fibonacci[i-1]+fibonacci[i-2];
			System.out.print(fibonacci[i]+" ");
		}
	}
}
```

**18. Write a java program to segregate array?**

**Input:**

**0 1 1 0 0 0 1 1 0 1**

**Output:**

**0 0 0 0 0 1 1 1 1 1**

```
class SegregateArrayElements
{
	public static void main(String[] args)
	{
		int[] arr={0,1,1,0,0,0,1,1,0,1};

		int[] resArr=new int[arr.length];

		int j=0;
		for(int i:arr)
		{
			if(i==0)
			{
				resArr[j++]=i;
			}
		}
		//inserting 1's
		while(j<arr.length)
		{
			resArr[j++]=1;
		}

		//display
		for(int i:resArr)
		{
			System.out.print(i+" ");
		}
	}
}
```

**19. Write a java program to merge two arrays and display them in sorting order?**

**Input:**

**5 2 1 3 4**
**9 7 8 6 10**

**Output:**

**1 2 3 4 5 6 7 8 9 10**

```
import java.util.Arrays;
class MergeSortArray
{
	public static void main(String[] args)
	{
		int[] arr1={5,2,1,3,4};
		int[] arr2={9,7,8,6,10};

		int size1=arr1.length; // 5
		int size2=arr2.length; // 5

		arr1=Arrays.copyOf(arr1,size1+size2);

		int j=0;

		for(int i=size1;i<arr1.length;i++)
		{
			arr1[i]=arr2[j++];
		}

		//sorting
		Arrays.sort(arr1);

		//display
		for(int i:arr1)
		{
			System.out.print(i+" ");
		}

	}
}
```

**20. Write a java program to display lucky number from given array?**

**Input:**

**1 2 2 3 3 3**

**Output:**

**3**

```
import java.util.*;
class LuckyNumberInArray
{
	public static void main(String[] args)
	{
		int[] arr={1,2,2,3,3,3};
		System.out.println(luckyInteger(arr));
	}
	public static int luckyInteger(int[] arr)
	{
		HashMap<Integer,Integer> hm=new HashMap<Integer,Integer>();
		for(int i=0;i<arr.length;i++)
		{
			if(hm.containsKey(arr[i]))
			{
				hm.put(arr[i],hm.get(arr[i])+1);
			}
			else
			{
				hm.put(arr[i],1);
			}
		}
		int x=0;
		int max=-1;

		for(Map.Entry<Integer,Integer> entry: hm.entrySet())
		{
			if(entry.getKey()==entry.getValue())
			{
				x=entry.getValue();
				max=Math.max(x,max);
			}
		}
		return max;
	}
}
```

**21. Write a java program to delete first occurrence of a given element?**

**Input:**

arr = 6 4 2 3 9 2 7 2 1
element = 2

**Output:**

6 4 3 9 2 7 2 1

```
class DeleteElement
{
	public static void main(String[] args)
	{
		int[] arr ={6,4,2,3,9,2,7,2,1};
		int element = 2;

		int[] resArr=new int[arr.length-1];

		int j=0,cnt=0;

		for(int i=0;i<arr.length;i++)
		{
			if(arr[i]==element && cnt==0)
			{
				cnt=1;
				continue;
			}
			resArr[j++]=arr[i];
		}
		//display
		for(int i:resArr)
		{
			System.out.print(i+" ");
		}

	}
}
```

**22. Write a java program to insert new element in a given index?**

**Input:**

arr = 8 4 1 6 9 2
element = 10
index = 5

**Output:**

8 4 1 6 9 10 2

```
import java.util.Arrays;
class InsertElementOnIndex
{
	public static void main(String[] args)
	{
		int[] arr ={8,4,1,6,9,2};
		int element = 10;
		int index = 5;

		arr=Arrays.copyOf(arr,arr.length+1);

		for(int i=arr.length-1;i>=index;i--)
		{
				arr[i]=arr[i-1];
		}
		arr[index]=element;

		//display
		for(int i:arr)
		{
			System.out.print(i+" ");
		}

	}
}
```

**23. Write a java program to find out missing element from given array?**

**Input:**

**5 6 2 3 1**

**Output:**

**4**

```
class MissingElement
{
	public static void main(String[] args)
	{
		int[] arr={5,6,2,3,1};

		int sum_of_arr_ele=arr.length+1;

		int sum=(sum_of_arr_ele * (sum_of_arr_ele + 1))/2;

		for(int i:arr)
		{
			sum-=i;
		}
		System.out.println("Missing element is ="+sum);
	}
}
```

**24. Write a java program to display leader elements from given array?**

**Q) Write a java program to display leader elements from given array?**

**input:**

**4 6 64 7 12 5 9**

**output:**

**9 12 64**

```
class LeaderElements
{
	public static void main(String[] args)
	{
		int[] arr={4,6,64,7,12,5,9};

		int max=arr[arr.length-1];

		System.out.print(max+" ");

		//reading reverse
		for(int i=arr.length-2;i>=0;i--)
		{
			if(arr[i]>max)
			{
				max=arr[i];
				System.out.print(max+" ");
			}
		}
	}
}
```

**25. Write a java program to identify and print all elements in an array that are greater than both their immediate predecessors and successors, considering the first and last elements as having only one neighbor?**

**Input:**

**1 3 20 4 75 0 90**

**Output:**

**20 75 90**

```
package com.ihub.www;
public class IdentifyElements
{
   public static void main(String[] args)
  {
      int[] array = {1, 3, 20, 4, 75, 0, 90};

      // Check the first element
      if (array[0] > array[1])
      {
         System.out.print(array[0] + " ");
      }

      // Check each element in the array
      for (int i = 1; i < array.length - 1; i++)
      {
         // element is greater than its immediate predecessor & successor
         if (array[i] > array[i - 1] && array[i] > array[i + 1])
         {
            System.out.print(array[i] + " ");
         }
      }

      // Check the last element
      if (array[array.length - 1] > array[array.length - 2])
      {
         System.out.print(array[array.length - 1]);
      }
   }
}
```

**26. Write a java program to determine the smallest number of coins needed to total 86 rupees. Use the denominations provided in the array {1,2,5,10}?**

**Output:**

```
1 coin(s) of 1 rupee(s)
1 coin(s) of 5 rupee(s)
8 coin(s) of 10 rupee(s)
```

```
public class MinimumCoins
{
   public static void main(String[] args)
   {
      int[] denominations = {1, 2, 5, 10};
      int amount = 86;
      int[] result = findMinimumCoins(denominations, amount);

      System.out.println("Minimum number of coins needed:");
      for (int i = 0; i < result.length; i++)
      {
         if (result[i] > 0) {
         System.out.println(result[i] + " coin(s) of " + denominations[i] + " rupee(s)");
         }
      }
   }

   public static int[] findMinimumCoins(int[] denominations, int amount)
   {
      int[] coinsCount = new int[denominations.length];

      // Start with the largest denomination
      for (int i = denominations.length - 1; i >= 0; i--)
      {
         // Calculate the number of coins needed for the current denomination
         coinsCount[i] = amount / denominations[i];
         amount %= denominations[i];
      }
      return coinsCount;
   }
}
```

**27. Write a java program to display largest prime number in the list?**

**Input:**

**2 3 4 5 7 9 11 12**

**Output:**

**11**

```
public class LargestPrime
{
   public static void main(String[] args)
   {
      int[] arr={2,3,4,5,6,7,8,9,11,12};

                int largestElement=Integer.MIN_VALUE;

                for(int n:arr)
                {
                        boolean flag=true;
                        for(int i=2;i<=n/2;i++)
                        {
                                if(n%i==0)
                                {
                                        flag=false;
                                        break;
                                }
                        }
                        if(flag==true)
                        {
                                if(n>largestElement)
                                {
                                        largestElement=n;
                                }
                        }
                }
                System.out.println(largestElement);
   }
}
```

**28. Write a java program to display pair of elements whose sum of equals to given number?**

**Input:**

**arr = 5 2 3 8 9 1 4 6**

**sum = 10**

**Output:**

**2 8**

**9 1**

**4 6**

```
class PairOfElements
{
	public static void main(String[] args)
	{
		int[] arr={5,2,3,8,9,1,4,6};

		int sum=10;

		for(int i=0;i<arr.length;i++)
		{
			for(int j=i+1;j<arr.length;j++)
			{
				if(arr[i]+arr[j]==sum)
				{
					System.out.println(arr[i]+"  "+arr[j]);
				}
			}

		}
	}
}
```

**29. Write a java program to display triplets of elements whose sum of equals to given number?**

**Input:**

**arr = 5 2 3 8 9 1 4 6**

**sum = 10**

**Output:**

**5 2 3**

**5 1 4**

**3 1 6**

```
class TripletOfElements
{
	public static void main(String[] args)
	{
		int[] arr={5,2,3,8,9,1,4,6};

		int sum=10;

		for(int i=0;i<arr.length;i++)
		{
			for(int j=i+1;j<arr.length;j++)
			{
				for(int k=j+1;k<arr.length;k++)
				{
					if(arr[i]+arr[j]+arr[k]==sum)
					{
				System.out.println(arr[i]+"  "+arr[j]+"   "+arr[k]);
					}
				}
			}

		}
	}
}
```

**30. Write a java program to display distinct elements from positive integer array?**

**Input:**

**1 2 2 3 3 3 4 4 4 4**

**Output:**

**1 2 3 4**

```
import java.util.*;
class DistinctElements
{
	public static void main(String[] args)
	{
		int[] arr={1,2,2,3,3,3,4,4,4,4};

		Set<Integer> set=new LinkedHashSet<Integer>();

		for(int i:arr)
		{
			set.add(i);
		}

		set.forEach(element -> System.out.print(element+" "));
	}
}
```

**31. Write a java program to display distinct and unique elements from given array?**

**Input:**

**4 7 1 2 7 9 6 6 4 8**

**Output:**

**Unique elements : 4 7 1 2 9 6 8**
**Duplicate elements : 4 7 6**

```
import java.util.*;
class DuplicateUniqueElements
{
	public static void main(String[] args)
	{
		int[] arr={4,7,1,2,7,9,6,6,4,8};

		Set<Integer> uniques=new LinkedHashSet<Integer>();
		Set<Integer> duplicates=new LinkedHashSet<Integer>();

		for(int i:arr)
		{
			if(uniques.contains(i))
			{
				if(!duplicates.contains(i))
				{
					duplicates.add(i);
					continue;
				}
			}
			uniques.add(i);
		}
		System.out.print("Unique Elements : ");
		uniques.forEach(element-> System.out.print(element+" "));

		System.out.println();

		System.out.print("Duplicate Elements : ");
		duplicates.forEach(element-> System.out.print(element+" "));
	}
}
```

**32. Given an integer array nums.find the contiguous subarray which has the largest sum 6?**

**Input:**

**nums = [-2,1,-3,4,-1,2,1,-5,4]**

**Output:**

**6**

**Explaination : [4,-1,2,1] has the largest continguous subarray.**

```
class Test
{
	public static void main(String[] args)
	{
		int[] nums={-2,1,-3,4,-1,2,1,-5,4};
		int maxCount=nums[0];
		int sum=nums[0];
		for(int i=1;i<nums.length;i++)
		{
			if(sum>0){
				sum+=nums[i];
			}
			else{
				sum=nums[i];
			}
			if(sum>maxCount){
				maxCount=sum;
			}
		}
		System.out.println(maxCount);
	}
}
```

**33. Write a java program to display array elements in spiral form**

**Input:**

```
1 2 3
4 5 6
7 8 9
```

**Output:**

```
1 2 3 6 9 8 7 4 5
```

```
class SpiralForm
{
	public static void main(String[] args)
	{
		int[][] matrix={
					{1,2,3},
					{4,5,6},
					{7,8,9}
				};

		int rows=matrix.length;
		int cols=matrix[0].length;

		int top=0;
		int bottom=rows-1;
		int left=0;
		int right=cols-1;

		while(true)
		{
			if(left>right)
			{
				break;
			}
			for(int i=left;i<=right;i++)
			{
				System.out.print(matrix[top][i]+" ");
			}
			top++;
```

```
                    if(top>bottom)
                    {
                        break;
                    }
                    for(int i=top;i<=bottom;i++)
                    {
                        System.out.print(matrix[i][right]+" ");
                    }
                    right--;

                    if(left>right)
                    {
                        break;
                    }
                    for(int i=right;i>=left;i--)
                    {
                        System.out.print(matrix[bottom][i]+" ");
                    }
                    bottom--;

                    if(top>bottom)
                    {
                        break;
                    }
                    for(int i=bottom;i>=top;i--)
                    {
                        System.out.print(matrix[i][left]+" ");
                    }
                    left++;
                }
            }
        }
```