1) How to reverse a string in java?
Write a java program to reverse a given string
String str = "I am not a string";
output: "gnirts a ton ma I"
programme:

```
package assigments;

import java.util.Scanner;

/**
 * @author 003OQT744
 *
 */
public class reverseString {
public static String reverseString(String str) {
	StringBuilder sb= new StringBuilder(str);
	sb.reverse();
	return sb.toString();
}
	public static void main(String[] args) {
		// TODO Auto-generated method stub

		String str;
		try (Scanner sc = new Scanner(System.in)) {
			System.out.print("Enter a string:");
			str = sc.nextLine();
		}
		System.out.println(reverseString(str));
	}

}
```

2) How to count occurrences of each character in a string in java?

Write a java program to count the number of occurrences of each character in a given string. For example, If "Java J2EE Java JSP J2EE" is the given string then occurrences of each character in this string is E=4, 2=2, v=2, =4, P=1, S=1, a=4, J=5.

Programme:

```
package assigments;


import java.util.Scanner;

public class charCount {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String st;
		int i,len;
		int counter[]=new int[256];
		//Scanner sc= new Scanner(System.in);
		//System.out.print("Enter the string:");
		st= "Java J2EE Java JSP J2EE";
		len = st.length();
		for(i=0;i < len; i++) {
			counter[(int)st.charAt(i)]++;
```

```
		}
		for(i=0;i<256;i++) {
			if(counter[i] !=0) {
				System.out.println((char)i+"="+ counter[i]);
			}
		}
	}

}
```

3) How to find second largest number in an integer array?

Write a java program to find second largest number in an array of numbers.

int[] a = {45, 51, 28, 75, 49, 42};

Programme:

```
import java.util.Arrays;
public class arraySecondLarg {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int[] a = {45, 51, 28, 75, 49, 42};
		//invoking sort() method of the Arrays class
		Arrays.sort(a);
		//to get the second largest number do arrays length - 2 to get
second last index value
		int b= a[a.length-2];
		System.out.println("Second Largest Number: "+ b);

	}

}
```

4) How to find largest number in an integer array?

Programme:

```
public class arrayLargest {

		static int arr[] = {10, 324, 45, 90, 9808,10929821};

		// Method to find maximum in arr[]
		static int largest()
		{
			int i;

			// Initialize maximum element
			int max = arr[0];

			// Traverse array elements from second and
			// compare every element with current max
			for (i = 1; i < arr.length; i++)
				if (arr[i] > max)
					max = arr[i];

			return max;
		}

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

```
            // TODO Auto-generated method stub

            // Java Program to find maximum in arr[]

                        System.out.println("Largest in given array is " +
largest());

            }


      }
```

5) How to create a pyramid of numbers in java?

Write a java program to create a pyramid of numbers or any other symbols in pyramid patterns.

a. 1st pattern:

12345
12345
12345
12345
12345

```
public class numberPattern1 {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            int rows=5;
            for(int i=1;i<=5;i++) {
                  for(int j=1;j<=5;j++) {
                        System.out.print(j+" ");

                  }
                  System.out.println();
            }
      }

}
```

b. 2nd pattern:

11111
22222
33333
44444
55555

```
public class numberPattern2 {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            int i;int j;
            for(i=1;i<=5;i++) {
                  for(j=1;j<=5;j++) {
                  System.out.print(i + " ");
            }
            System.out.println();
      }
```

```
}
}
```

c. 3rd pattern:

1
12
123
1234
12345

```
public class numberPattern2 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i;int j;
		for(i=1;i<=5;i++) {
			for(j=1;j<=5;j++) {
			System.out.print(i + " ");
		}
		System.out.println();
	}

}
}
```

d. 4th pattern:

1
22
333
4444
55555

```
public class numberPattern3 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i,j,row=5;
		for(i=1;i<=row;i++) {
			for(j=1;j<=i;j++) {
				System.out.print(j+" ");
			}
			System.out.println();
		}
	}

}
```

e. 5th pattern:

1
23
456
78910

```
public class numberPattern5 {
```

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int row=4;
		int value=1;
		for(int i =1;i<=row;i++) {
			for(int j=1;j<=i;j++) {
				System.out.print(value+" ");
				value++;
			}
			System.out.println();
		}
	}

}
```

f. 6th pattern:

```
 1
 12
 123
 1234
12345
```

```
public class numberPattern8 {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		for(int i=1; i<=5;i++) {
			for(int k= 4;k>=i;k--) {
				System.out.print(" ");
			}
			for(int j=1; j<=i;j++) {
				System.out.print(j);
			}
			System.out.println();

		}
}
}
```

g. 7th pattern:

```
  1
 121
12321
1234321
123454321
```

```
public class numberPattern9 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i, j, row;
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter the number of rows:");
		row= sc.nextInt();
		for(i=1;i<=row;i++) {
			for(j=1;j<=row-i;j++)
		  {
			System.out.print(" ");
		  }
```

```
                for(j=1;j<=i;j++)
                {
                  System.out.print(j);
                }

                for(j=i-1;j>=1;j--)
                {
                    System.out.print(j);
                }
System.out.println();
            }
      }

}
```

h. 8th pattern:

```
*****
*****
*****
*****
*****
```

```
public class starPattern1 {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            int i;int j;
            for(i=1;i<=5;i++) {
                  for(j=1;j<=5;j++) {
                  System.out.print( "* ");
            }
            System.out.println();
      }

      }

}
```

i. 9th pattern:

```
*
**
***
****
*****
```

```
public class starPattern2 {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            int rows=5;
            for(int i=1;i<=rows;i++) {
                  for(int j=1;j<=i;j++) {
                        System.out.print("* ");
                  }
                  System.out.println();
            }
```

```
	}

}
```

j. 10th pattern:

```
  *
 **
***
****
*****
public class starPattern3 {


	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i,j,k;
		for(i=1;i<=5;i++) {
			for(k=4;k>=i;k--) {
				System.out.print(" ");
			}
			for(j=1;j<=i;j++) {
				System.out.print("*");
			}
			System.out.println();
		}



	}
}
```

k. 11th pattern:

```
*****
****
***
**
*
public class starPattern6 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int rows = 5;

	    for (int i = rows; i >= 1; --i) {
	      for (int j = 1; j <= i; ++j) {
	        System.out.print("* ");
	      }
	      System.out.println();
	    }
	}

}
```

l. 12th pattern:

```
*****
 ****
  ***
   **
    *
```

```
public class starPattern7 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int rows = 5;

	    for (int i = rows; i >= 1; --i) {
		  for(int k=4;k>i;--k) {
			System.out.print(" ");
		  }
		  for (int j = 1; j <= i; ++j) {
		    System.out.print("*");
		  }
		  System.out.println();
	    }
	}

}
```

m. 13th pattern:

```
    *
   ***
  *****
 *******
*********
```

```
import java.util.Scanner;

public class starPattern4 {
	public static void main(String[] args) {
	int i, j;

	for(i=1;i<=5;i++) {
		for(j=1;j<=5-i;j++)
	  {
		System.out.print(" ");
	  }

	  for(j=1;j<=i;j++)
	  {
	    System.out.print("*");
	  }

	  for(j=i-1;j>=1;j--)
	  {
		System.out.print("*");
	  }
	  System.out.println();
	}
}
}
```

n. 14th pattern:

```
   *
  * *
 * * *
* * * *
* * * * *
* * * * * *
```

```
public class starPattern5 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		for(int i=1; i<=6;i++) {
			for(int k= 5;k>=i;k--) {
				System.out.print(" ");
			}
			for(int j=1; j<=i;j++) {
				System.out.print("* ");
			}
			System.out.println();

		}
	}

}
```

6) How to remove duplicate elements from ArrayList in java?
Write a java program to remove duplicate elements from already constructed ArrayList.
Programme:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.LinkedHashSet;
import java.util.Set;

public class dupArrayelemRemove {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

				ArrayList<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 1,
3,6,6, 4,7,3,5,3,6,4,4,6,2,6));
				System.out.println("ArrayList with duplicate elements: " + numbers);

				// convert the arraylist into a set
				Set<Integer> set = new LinkedHashSet<>();
				set.addAll(numbers);

				// delete all elements of arraylist
				numbers.clear();

				// add element from set to arraylist
```

```
                        numbers.addAll(set);
                        System.out.println("ArrayList without duplicate elements: " + numbers);
        }

}
```

7) How to reverse each word of a string in java?

Write a java program to reverse each word of a given string. For example, If "Java Concept Of The Day" is input string then output should be "avaJ tpecnoC fO ehT yaD".

Programme:

```
package assigments;

import java.util.Scanner;

/**
 * @author 003OQT744
 *
 */
public class reverseString {
public static String reverseString(String str) {
      StringBuilder sb= new StringBuilder(str);
      sb.reverse();
      return sb.toString();
}
      public static void main(String[] args) {
            // TODO Auto-generated method stub

            String str;
            try (Scanner sc = new Scanner(System.in)) {
                  System.out.print("Enter a string:");
                  str = sc.nextLine();
            }
            System.out.println(reverseString(str));
      }

}
```

8) Write a java program to swap two numbers without using third or temp variable?

```
package assigments;

import java.util.Scanner;

public class numberSwap {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            System.out.println("Enter the numbers A and B ");
            Scanner s= new Scanner(System.in);
            //int A = s.nextInt();
            //int B = s.nextInt();
            int A=2;
            int B= 3;
            System.out.print("Numbers befor swaping A & B: "+ A +" "+ B);
            A = A+B;//A*B
            B = A-B;//A/B
```

```
            A = A-B;//A/B
            System.out.println("Number after swap:"+A +" "+B);


      }

}
```

9) How to remove all vowels from a string in java?

Write a java program to remove all vowels from a string. Your program should take input string from the user and remove all vowels from this string and print it without vowels

```
package assigments;

import java.util.Scanner;

public class vowelsRemove {
//Write a java program to remove all vowels from a string
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            System.out.println("Enter the String:");
            Scanner sc = new Scanner(System.in);
            String s = sc.next();
            String s1="";
            s1 = s.replaceAll("[aeiouAEIOU]", "");
            System.out.println("String after removing Vowel: "+s1);
      }

}
```

10) Write a java program to add two numbers without using arithmatic operator(+,-,*,/,%).

```
package assigments;

public class addingWithoutOperator {
//Write a java program to add two numbers without using arithmatic
operator(+,-,*,/,%)
      public static int add(int a, int b) {
            /*
             * the while loop will run untill b is qual to 0
             * [this is when loop will run first
                  * a= 4 =0100
                  * b= 5 =0101
                  * carry = a&b=    0100 = 4
                  * & AND bitwise operator
                  * Bitwise XOR
                  * a=4=0100
                  *b=5=0101
                  *a^b=0001
                  *b= carry = 4= 0100 << 1=0000 ]
                  * */
            while(b!=0) {
                  int carry = a&b;//bitwise AND a&b

                  a= a^b;// bitwise XOR , ^ XOR bitwise operator

                  b=carry<<1;// left shift carry by 1 // << left shift

            }
            return a;
      }
```

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println(add(4, 5));//ans 9

	}

}
```

11) Write a java program to Multiple two numbers without using multiplication operator.

```
package assigments;

import java.util.stream.Stream;

public class mulWithoutop {
	public static int mult(int a, int b) {
		int res = 0;
		while(b!=0) {
			res+=a;
			b--;
		}return res;
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		 short sh = (short)'A';
		 int x=12_4;
		System.out.println("Muliplication: "+mult(3,5));
		System.out.println(sh +x);
	}



}
```

12) Write a java program to find factorial of a given number.

```
package assigments;
//Write a java program to find factorial of a given number.
public class Factorial {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i, fact =1;
		int number = 8;
		for(i=1;i<=number;i++) {
			fact=fact*i;
		}
		System.out.println(fact);
	}

}
```

13) Write a java program to Find length of String without using java inbuilt length method.

```
package assigments;

public class stringLength {
//Write a java program to Find length of String without using java inbuilt
length method.
```

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String sd="Hello Everyone";
		int i=0;
		for(char c: sd.toCharArray()) {
			i++;
		}
		System.out.println("Length of the given string: "+i);
	}

}
```

14) Write a java program to find whether a number entered by user is odd and even number.

```
package assigments;

import java.util.Scanner;

public class oddEven {
//Write a java program to find whether a number entered by user is odd and
even number
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter The Number");
		int num = sc.nextInt();
		if(num%2 == 0) {
			System.out.println(num + " is even");
		}else {
			System.out.println(num+ " is odd");
		}

	}

}
```

15) Write a java program to find all prime numbers between 1 to 100.

```
import java.util.Scanner;

public class prime {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int n, i, m = 0, flag = 0;
		Scanner sd = new Scanner(System.in);
		System.out.println("Enter the number: ");
		n = sd.nextInt();
		m = n / 2;

		for (i = 2; i <= m; i++) {
			if (n % i == 0) {
				System.out.println("The number is not prime.");
				flag = 1;
				break;
			}
		}
		if (flag == 0) {
			System.out.println("The number is prime.");
```

```
			}
		}

}
package assigments;

public class primeNumber {
//Write a java program to find all prime numbers between 1 to 100
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int ct=0,n=0,i=1,j=1;
		while(n<25)
		{
		j=1;
		ct=0;
		while(j<=i)
		{
		if(i%j==0)
		ct++;
		j++;
		}
		if(ct==2)
		{
		System.out.printf("%d ",i);
		n++;
		}
		i++;
		}
		}

	}
```

16) How to find all pairs of elements in an array whose sum is equal to given number?

Write a java program to find all pairs of elements in the given array whose sum is equal to a given number. For example, if {4, 5, 7, 11, 9, 13, 8, 12} is an array and 20 is the given number, then you have to find all pairs of elements in this array whose sum must be 20. In this example, (9, 11), (7, 13) and (8, 12) are such pairs whose sum is 20.

```
package assigments;
/**
 * Find pairs with given sum in an array for sorted array
 * @author 003OQT744
 *
 */
public class pairSum_method2 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//Sorted array with distinct elements
		 int arr[] = {-5,2,4,5,7, 11, 20};
		 int sum = 15;


		 //Two indexes low and high
		 int low = 0;
```

```
            int high = arr.length - 1;

            while(low < high) {

               /*
                 if sum of arr[low] + arr[high] is greater than the
                 value of sum then decrement the value of high.
              */
               if((arr[low] + arr[high]) > sum) {
                     high--;

               } else if ((arr[low] + arr[high]) < sum) {
                     low++;

               } else if((arr[low] + arr[high]) == sum) {
                     System.out.println(" Pair (" + arr[low] + " , " +
arr[high] + " )");
                     low++;
                     high--;
               }
            }
         }

      }
```

```
package assigments;
//How to find all pairs of elements in an array whose sum is equal to given
number
// using for loop
public class pairs_value {
      static  void pairs_value(int inputArray[],int inputNumber) {
            System.out.println("pairs of elements and their sum: ");
            for(int i=0;i < inputArray.length; i++) {
                  for(int j = i+1; j < inputArray.length; j++) {
                        if(inputArray[i]+inputArray[j] == inputNumber) {
                              System.out.println(inputArray[i]+" +
"+inputArray[j]+" =  "+inputNumber);
                        }
                  }
            }
      }
      public static void main(String[] args) {
            // TODO Auto-generated method stub
             pairs_value(new int[] {2, 7, 4, -5, 11, 5, 20}, 15);
             pairs_value(new int[] {12, 45, 2, 1, 9, 13, 8, 16},14);
              pairs_value(new int[] {14, -15, 9, 16, 25, 45, 12, 8}, 30);
              pairs_value(new int[] {4, 5, 7, 11, 9, 13, 8, 12},20);
      }

}
```

17) All permutations of a string.

Write a java program to find all permutations of a string recursively. For example, all permutations of string “JSP“ are,

JSP

JPS

SJP

SPJ

PJS

PSJ

```
package assigments;

public class Permutation {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String str = "JSP";
	  int n = str.length();
	  Permutation permutation = new Permutation();
	  permutation.permute(str, 0, n - 1);
	 }

    /**
     * permutation function
     * @param str string to calculate permutation for
     * @param l starting index
     * @param r end index
     */
    private void permute(String str, int l, int r)
    {
        if (l == r)
            System.out.println(str);
        else {
            for (int i = l; i <= r; i++) {
                str = swap(str, l, i);
                permute(str, l + 1, r);
                str = swap(str, l, i);
            }
        }
    }

    /**
     * Swap Characters at position
     * @param a string value
     * @param i position 1
     * @param j position 2
     * @return swapped string
     */
    public String swap(String a, int i, int j)
    {
        char temp;
        char[] charArray = a.toCharArray();
        temp = charArray[i];
        charArray[i] = charArray[j];
        charArray[j] = temp;
        return String.valueOf(charArray);
    }
}
```