Write a program for check the number is prime or not....

```
public class PrimeNumberOrNot {
        public static void main(String[] args) {
                int i,m=0,flag=0;
                 int n=55;                                              //  number to be check
                 m=n/2;
                 if(n==0||n==1){
                  System.out.println(n+" is not prime number");
                 }else{
                  for(i=2;i<=m;i++){
                   if(n%i==0){
                    System.out.println(n+" is not prime number");
                    flag=1;
                    break;
                   }
                  }
                  if(flag==0)  { System.out.println(n+" is prime number"); }
                 }//end of else
               }
}
```

Write a program to print prime numbers

```
        public static void main(String[] args) {
                int n;
            int status = 1;
            int num = 3;
            //For capturing the value of n
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the value of n:");
            //The entered value is stored in the var n
            n = scanner.nextInt();
            if (n >= 1)
            {
              System.out.print("First "+n+" prime numbers are:");
              //2 is a known prime number
              System.out.print(2);
            }

            for ( int i = 2 ; i <=n ;  )
            {
              for ( int j = 2 ; j <= Math.sqrt(num) ; j++ )
              {
                if ( num%j == 0 )
                {
                  status = 0;
                  break;
                }
              }
              if ( status != 0 )
              {
                System.out.print(","+num);
                i++;
              }
              status = 1;
              num++;
            }    }
```

Write a program for Floyd triangle

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

```
public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i,j,n=5,m=1;
		for(i=1;i<=n;i++) {
			for(j=1;j<=i;j++) {
				System.out.print(m+" ");// here we can print any Symbols we want to *,$,# etc,..
				m++;
			}
			System.out.println();
		}
	}
```

Write a program for Sum of First n numbers

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

		int sum=0,limit=100;
		for(int i=0;i<=limit;i++) {
			sum=sum+i;
}

// Change operator * to get First N numbers Multiplication value but i value must be 1
// if change the for loop like i=1 and i=1+2 we get sum/mul of first n odd numbers
// if change the for loop like i=0 and i=1+2 we get sum/mul of first n even numbers
	System.out.printf("Sum of first %d is %d", limit,sum);

// printf method is just works like C language printf method
// %d is for integer %f for Floating point %c is for char %s for print String
// %n is for printing the message in next line separately
	}
```

Write a program to find the factorial number of given number

```
public static void main(String[] args) {
		// Fact of 4 is 1*2*3*4 = 24
		// Fact of 6 is 1*2*3*4*5*6 = 720
		int fact=1,num=4;
		// most commonly we declare i=0 but here we are doing multiplication so its must be 1
		for(int i=1;i<=num;i++) {
			fact=fact*i;
		}
		System.out.printf("Factorial of  %d is %d", num,fact);
	}
```

Write a program for check the given number is palindrome or not

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

		int num=121; // change the number with 123321,1221,121 like so on
		int temp=num;
// to check the number is palindrome or not we have to store that into temp becz num value will change after loop
		int rev = 0;
		while(num!=0)
		{
			rev=rev*10;
			rev=rev+num%10;
			num=num/10;
//		1st iteration rev=1(0*10=0+1) num =12
//		2nd iteration rev=12(1*10=10;then 10+2(num%10)) num=2
//		3rd iteration rev=121(12*10=120;then 120+1 =121;) num =2%10=1; so overall value is 121
//		after 3rd iteration 2/10 is become 0 so loop will fail;
		}

		System.out.println("The reverse number of"+temp+" is "+rev);

//		following program is To check Palindrome or not

		if(temp==rev) {
			System.out.println(temp+" is Palindrome");
		}else {
			System.out.println(temp+" is Not Palindrome");
		}
	}
```

Write a program to check the given number is Armstrong or not.

```
public static void main(String[] args) {
		// TODO Auto-generated method stub
//		Armstrong Number Ex 153
//		1^3 + 5^3 +3^3
//		1+125+27 = 153
		int r,num=155,sum=0,temp;
//change the num value to get diff outputs 153 371 etc,..
		temp=num;
		while(num>0) {

			r=num%10;
			num=num/10;
			sum=sum+(r*r*r);
					// first iteration r=5 num =15 sum= 125
					// 2nd iteration r=5 num=1 sum=125+125
					// 3rd iteraton r=1 num=0 sum=125+125+1
					// hence sum = 251 so 155 is not Armstrong Number
		}
		System.out.println("Armstrong Number of "+temp+" is  "+sum);

		if(temp==sum)
		{
			System.out.println(temp+" is Armstrong Number");
```

```
		}else {
			System.out.println(temp+" is Not Armstrong Number");
		}
}
```

Write a program to print Fibonacci series

```
public class Fibnocci {

	public static void main(String[] args) {

		int f1=0; // for 1st number
		int f2=1; // for 2nd number
		int f3;
		int n=10;
		System.out.print("Fibnocci series  "+f1+" , "+f2);
		for(int i=2;i<n;i++) {
			f3=f1+f2;
			System.out.print(" , "+f3);
			f1=f2;
			f2=f3;
		}

	}

}
```

Write a program for swapping two numbers

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

//		method 1 with 3rd variable

		int a=10,b=20,temp=0;
		System.out.printf("%n Before Swapping a=%d and b=%d ",a,b);
		temp=a;			a=b;			b=temp;
		System.out.printf("%n After Swapping a=%d and b=%d ",a,b);

//		method 2 Without 3rd variable a=a+b; b=a-b; a=a-b;
		int c=98,d=78;
		System.out.printf("%n Before Swapping c=%d and d=%d ",c,d);
		c=c+d;		d=c-d; c=c-d;
		System.out.printf("%n After Swapping c=%d and d=%d ",c,d);


	}
```

Some basic operations on java Arrays

```
import java.util.Arrays;

public class arrayOperations {

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

		int arr[]= {43,66,89,12,4,5,6,84,9};

//		method 1 to find max element in array
		int max=0;
		int sum=0;	// For find sum of array elements
		int temp=0;	//To perform sorting

		for(int i=0;i<arr.length;i++) {
			sum=sum+arr[i];
			if(arr[i]>max)// we use < to print minimum value
				max=arr[i];
		}
		System.out.println("largest element in Array is "+max);
		System.out.println("Sum of array elements is "+sum);


//		method 2 sorting and printing last element of array

//	Arrays.sort(arr);
//	System.out.println("largest element in Array is "+arr[arr.length-1]);
//
//		System.out.println("minimum element in Array is "+arr[0]);

		// Sorting Ascending and descending order

		for(int i=0;i<arr.length;i++) {
			for(int j=i+1;j<arr.length;j++) {
				if(arr[i]>arr[j])		// < use Less then to sort the array in
descending order
				{
					temp=arr[i];
					arr[i]=arr[j];
					arr[j]=temp;
				}
			}
		}

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

		System.out.println();
		System.out.println("largest element in Array is "+arr[arr.length-1]);

		System.out.println("minimum element in Array is "+arr[0]);
```

```
                System.out.println();


//              printing Even and Odd numbers in array
                System.out.println("Even numbers :");
                for(int i=0;i<arr.length;i++) {
                        if(arr[i]%2==0) {
                                System.out.print(" "+arr[i]);
                        }
                }
                System.out.println();
                System.out.println("Odd numbers :");
                for(int i=0;i<arr.length;i++) {
                        if(arr[i]%2!=0) {
                                System.out.print(" "+arr[i]);
                        }
                }

        }

}
```

Write a program for print alternative elements in given array

Input {23,43,54,63,73,83,92,13,56}

Output 1 {23,54,73,92,56}

Output 2 {43,63,83,13}

```
public class Printing_AlternativeElementsInArray {

        public static void main(String[] args) {

//      Printing Alternatives in Array like
//      Input is {1,2,3,4,5,6,7,8,9}
//      Output 1 is {1,3,5,7,9}
//      Output 2 is {2,4,6,8}
//      even and odd are different here we are printing the values based on Array index
                int[] arr= {23,43,54,63,73,83,92,13,56};

                for(int i=0;i<arr.length;i=i+2) {
// i=0 so its print form 1st Element of array
                        System.out.print(" "+arr[i]);
// Output is 23,54,73,92,56 each time Loop iterates the i(index value) value is increasing
2 so it skip printing one element every time becz of .
                }
                System.out.println();
                for(int i=1;i<arr.length;i=i+2) {
// i=1 so its print form 2nd Element of array
                        System.out.print(" "+arr[i]);
// Output is 43 63 83 13 each time Loop iterates the i(index value) value is increasing 2
        so it skip printing one element every time.
                        }
        }

}
```

Write a program to find the duplicate elements in an Array

```java
public class duplicatesInArray {

	public static void main(String[] args) {

		int arr[]= {10,40,20,10,30,50,70,60,70,50,40};

//		Print Duplicates in array method 1

		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[j]+" ");
				}
			}
			}

//		Print Duplicates in array method 2
		System.out.println(" ");
		int length= arr.length;
		Arrays.sort(arr);
		int[] temp=new int[length];
		int j=0;
		for(int i=0;i<length-1;i++) {
			if(arr[i]==arr[i+1]) {
//	if(arr[i]!=arr[i+1]) and temp[j++]=arr[length-1]; are remove the duplicates from array
				temp[j++]=arr[i];
//	if(arr[i]==arr[i+1]) it only print the duplicate values from an array
				}

		}
		temp[j++]=arr[length-1];

		for(int i=0;i<=temp.length-1;i++) {
			System.out.print(" "+temp[i]);
		}

	}
```

Searching techniques

Linear Searching approach

```
package javaInterviewPrograms;

public class sreachingLinearApproach {

	public static void main(String[] args) {

// Linear searching approach is very basic we just have to compare the searching element with all array elements until its found

		int[] arr= {20,32,94,76,89,65,34,2,4,98};
		int searchEle=65;			// we are finding the 65 and its index in the Given Array
		int flag=0;					// to check the element status
		for(int i=0;i<arr.length;i++)
		{
		if(arr[i]==searchEle) {
		System.out.println(searchEle+" Is Found at  the Index Of  "+i);
		flag=1;
		break;
			}

		}
//		System.out.println(flag);
//If We use the searching element other then elements of array the flag value is //not change and the following block of code will execute

		if(flag==0)
			System.out.println(searchEle+" Is Not Found");

//Output is 65 is found att Index of 5 becz array index is start from 0

	}
}
```

Binary Searching approach

```
public class binarySearchingApproach {

	public static void main(String[] args) {

//		Binary searching approach is more efficient then linear searching and
//		Array is must be in sorted order to perform binary search
//		To sort the array we can use Arrays.sort() or we can create our own sorting method(insertion,merge,selection,quik,etc,..)
//		Binary search is finds the middle element of the sorted array and compare the target element


		int arr[]= {21,45,3,1,2,6,56,7,68,56,34};
		int searchEle=45;
```

```
		Arrays.sort(arr);		// Here im using Arrays.sort() method to sort the array
		int first=0;			// its represent the 1st element of the array based on index
		int last=arr.length-1;// its represent the Last element of the array based on index
		int mid=0;
		int flag=0;				//	flag is to check point
		if(first>last)
			System.out.println("Element not Found");
		while(first<=last) {
			mid=(first+last)/2;		// we are specifying mid value here to divide the array into two parts
									// and it will continuously break the array untill element is found

			if(arr[mid]==searchEle) {

				System.out.println(searchEle+" is Found at Index  "+mid);
				flag=1;
				break;
			}
			else if(arr[mid]<searchEle) {
				first=mid+1;
			}else {
				last=mid-1;
			}


		}
		if(flag==0)
			System.out.println(searchEle+" Is Not Found");


//		Method 2 By using Arrays.binarySearch()

		int result=Arrays.binarySearch(arr, searchEle);
		 if (result < 0)
		System.out.println(searchEle+" is not found!");
	    else
		System.out.println(searchEle+" is found at index: "+result);

	}

}
```

Strings Basics

Write a program to change the Case of letters in the String

Input = “I Am SorRy :(“

Output = “i aM sORrY :(”

```
public class changeCase {

	public static void main(String[] args) {


		String str="I Am Sorry,:(";
		StringBuffer newStr=new StringBuffer(str);

// loop to change every character Case in string

		for(int i=0;i<str.length();i++) {
 //Checks for lower case character
			if(Character.isLowerCase(str.charAt(i))) {
//Convert it into upper case using toUpperCase() function
		newStr.setCharAt(i, Character.toUpperCase(str.charAt(i)));
				}
//Checks for Upper case character
			else if(Character.isUpperCase(str.charAt(i))) {
//Convert it into lower case using toLowerCase() function
		newStr.setCharAt(i, Character.toLowerCase(str.charAt(i)));
			}
		}
	    System.out.println("String after case conversion : " + newStr);
	}
}
```

Write a program to Calculate No of letters present in the String

Input = “I Am Sorry”

Output = 8

```
public class CharCountOfString {
	public static void main(String[] args) {
	String string = "I am Sorry";
      int count = 0;
//Counts each character except space
      for(int i = 0; i < string.length(); i++) {
	 if(string.charAt(i) != ' ')
	    count++;
      }
//Displays the total number of characters present in the given string
      System.out.println("Total number of characters in a string: " + count);
	}

}
```

Write a program to Calculate No of letters present in the String

Input = “This is very simple way to convey my Apologize ”

Output =
Number of vowels: 14
Number of consonants: 24

```
public class FindVowelsAndConsonants {

        public static void main(String[] args) {
    //To findout no.of Vowels and Consonants in the string
    //Counter variable to store the count of vowels and consonant
       int vCount = 0, cCount = 0;
       String str = "This is very simple way to convey my Apologize";    //
   //Converting entire string to lower case to reduce the comparisons
       str = str.toLowerCase();

       for(int i = 0; i < str.length(); i++) {
          //Checks whether a character is a vowel
          if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) ==
'o' || str.charAt(i) == 'u') {
             //Increments the vowel counter
             vCount++;
          }
          //Checks whether a character is a consonant
          else if(str.charAt(i) >= 'a' && str.charAt(i)<='z') {
             //Increments the consonant counter
             cCount++;
          }
       }
       System.out.println("Number of vowels: " + vCount);
       System.out.println("Number of consonants: " + cCount);
          }

}
```

Write a program to find the subsets of String

Input = “ANIL”

Output = A AN ANI ANIL N NI NIL I IL L

```
public class noOfsubsetsOfString {

        public static void main(String[] args) {

                String str="ANIL";
                int len=str.length();
                int temp=0;
                String[] arr=new String[len*(len+1)/2];
//        Total possible subsets for string of size n is n*(n+1)/2
//        This loop maintains the starting character
                for(int i=0;i<len;i++) {
//This loop adds the next character every iteration for the subset to form and add it to the
array
         for(int j = i; j < len; j++) {
                arr[temp]=str.substring(i,j+1);
                temp++;
```

```
        }

                }

                //printing all subsets of string
                for(int i=0;i<arr.length;i++) {
                        System.out.print("  "+arr[i]);
                }
        }

}
```

Write a program to find the duplicate characters in the given String

Input = “iiaammssrryy”

Output = i a m s r y
List of duplicate characters in String 'iiaammssrryy'
a : 2 r : 2 s : 2 i : 2 y : 2 m : 2

```
public static void main(String[] args) {
String str="iiaammssrryy";

//              method 1
                char[] temp=str.toCharArray();
                int len=temp.length;
                int count=0;
//              int k=0;
                System.out.println("Dupicates of String is \n");

                for(int i=0;i<len;i++) {
                        for(int j=i+1;j<len;j++) {
                                if(temp[i]==temp[j]) {

                                        System.out.print(temp[j]+"  ");
//                                      count++;
                                        break;
                                }}}

//              method 2
                System.out.println();
                Map<Character, Integer> charMap = new HashMap<Character,
Integer>();
        for (Character ch : temp) {
           if (charMap.containsKey(ch)) {
              charMap.put(ch, charMap.get(ch) + 1);
           } else {
              charMap.put(ch, 1);
           }  }

        // Iterate through HashMap to print all duplicate characters of String
        Set<Map.Entry<Character, Integer>> entrySet = charMap.entrySet();
        System.out.printf("List of duplicate characters in String '%s' %n", str);
        for (Map.Entry<Character, Integer> entry : entrySet) {
           if (entry.getValue() > 1) {
              System.out.printf("%s : %d %n", entry.getKey(), entry.getValue());
           }
```

```
    }
  }
```

Write a program to compare the given two strings are Anagrams or Not

Anagrams example

Anil and **Nail** are anagrams because chars in both strings are same more examples are

Mamu and Ammu are anagrams ; Poi and iop,oip,pio;

```
import java.util.Arrays;

public class StringAnagram {

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

		String str1="simary",str2="aryims";
		str1=str1.toLowerCase();
		str2=str2.toLowerCase(); // Converting to lower Case

		if(str1.length()!=str2.length())
		{
			System.out.println("Both are not Anagrams ");
		}
		else {
			// Converting to character arrays
			char[] string1=str1.toCharArray();
			char[] string2=str2.toCharArray();

//			Sort character arrays to compare

			Arrays.sort(string1);
			Arrays.sort(string2);

			if(Arrays.equals(string1,string2)==true) {
				System.out.println("Both Strings are Anagrams");
			}else {
				System.out.println("Both Strings are Not Anagrams");
			}


		}

	}

}
```

Write a program to find out given String is palindrome or not

To check the string is palindrome we have to reverse the string then we have to compare both result and given string..

```
public class stringReverseAndPalindrome {

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

		String str="yrsmai",temp="";
		int len=str.length();
//		method 1 reverse the string

		for(int i=len-1;i>=0;i--) {
			temp=temp+str.charAt(i);
		}
		System.out.println("reverse of the string is :: "+temp);

		if(str.equalsIgnoreCase(temp))
			System.out.println("\n Given String is Palindrome");
		else
			System.out.println("\n Given String is Not Palindrome");

//		String reverse method 2

		StringBuffer string=new StringBuffer(str);
		string.reverse();
		System.out.println("reverse of the given "+str+" is:: "+string);
	}

}
```

Write a program to swap two strings

Input :: str1=”anil”,str2=”kumar”

Output :: str1=”kumar”,str2=”anil”

```
public class swapingTwoStringsWithoutTemp {

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

		String str1="Anil";
		String str2="Kumar";

		System.out.println("Strings Before Swapping 1st String:: "+str1+"\n 2nd
String:: "+str2);
//		In  Println method \n is used to print in nextline
//		IN  Printf method we %n to print the some part of message in nextline

		str1=str1+str2;
		str2=str1.substring(0,str1.length()-str2.length());
		str1=str1.substring(str2.length());


		System.out.println("Strings After Swapping 1st String:: "+str1+"\n 2nd
String:: "+str2);
		System.out.println("Strings Before Swapping ::"+str1+str2);
```

```
        }
    }
```