**Q1: Remove Specific String From String Array:**

package com.java.paras;

import java.util.Scanner;

public class RemoveString

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

String str[]={"hello","hai","hey","hi","hai","hmmm"};

System.*out*.println("-- String Array are: --");

for (int i = 0; i < str.length; i++)

{

System.*out*.print(str[i]+" ");

}

S.o.p("\n-- Enter the specific String for removing --");

String st=sc.next();

String s[]=*removeItem*(str,st);

S.o.p("-- After Removing Special String from Array: --");

for (int i = 0; i < s.length; i++)

{

System.*out*.println(s[i]);

}

}

public static String[] removeItem(String[] input, String item)

{

if (input == null)

{

return null;

}

else if (input.length <= 0)

{

return input;

}

else

{

String[] output = new String[input.length - 1];

int count = 0;

for (String i : input)

{

if (!i.equals(item))

{

output[count++] = i;

}

}

return output;

}

}

}

**Output**:

-- String Array are: --

hello hai hey hi hai hmmm

-- Enter the specific String for removing --

hello

-- After Removing Special String from Array: --

hai

hey

hi

hai

hmmm

**Q2. String Palindrome Test :**

package com.java.JavaTest;

import java.util.Scanner;

public class StringPalindrome

{

public static void main(String[] args)

{

System.*out*.println("Enter the String:");

Scanner sc=new Scanner(System.*in*);

String str1=sc.next();

if(*IsPalindrome*(str1))

{

System.*out*.print(str1+": The String is Palindrome");

}

else

{

System.*out*.print(str1+": String is not Palindrome");

}

}

static boolean IsPalindrome(String s)

{

System.*out*.println();

char ch[]=s.toCharArray();

for (int i = 0; i < ch.length / 2; i++)

{

if (ch[i] != ch[ch.length - 1 - i])

return false;

}

return true;

}

}

**Output**:

Enter the String:

paras

paras: The given String is not Palindrome

Enter the String:

madam

madam: The given String is Palindrome

**Q3. String Permutation Program:**

package com.java.paras;

import java.util.Scanner;

public class PermutationTest

{

static int *count*=1;

public static void main(String[] args) throws Exception

{

System.*out*.println("Enter the String: ");

Scanner sc=new Scanner(System.*in*);

String str=sc.nextLine();

System.*out*.println("The Arrangement of String are: ");

*showPattern*(" ", str);

}

public static void showPattern(String st, String str)

{

if(str.length() <= 1)

System.*out*.println((*count*++)+" Times: "+st + str);

else

for (int i = 0; i < str.length(); i++)

{

try

{

String newString=str.substring(0,i)+str.substring(i+1);

*showPattern*(st + str.charAt(i), newString);

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

}

**Output:**

-- Enter the String:--

dog

The Arrangement of String are:

1 Times: dog

2 Times: dgo

3 Times: odg

4 Times: ogd

5 Times: gdo

6 Times: god

**Q4: Find longest palindrome from a String:**

package com.java.paras;

public class BIgPalindrome

{

public static void main(String[] args)

{

System.*out*.println("-- The String is: --");

String bigString = "aaabbaaaccdeqjncsdddmmmkkkmmmddd";

System.*out*.println(bigString);

System.*out*.println("-- After Finding Big Palindrome --");

String bigPoli = "";

for (int i = 0; i < bigString.length(); i++)

{

for (int j = i + 1; j < bigString.length(); j++)

{

String s = bigString.substring(i, j);

if (*isPolindrome*(s))

{

if (s.length() > bigPoli.length())

{

bigPoli = s;

}

}

}

}

System.*out*.println(bigPoli);

}

public static boolean isPolindrome(String s)

{

boolean poli = false;

String a = "";

for (int i = s.length() - 1; i >= 0; i--)

{

a = a + s.charAt(i);

}

if (s.equals(a))

{

poli = true;

}

return poli;

}

}

**Output:**

-- The String is: --

aaabbaaaccdeqjncsdddmmmkkkmmmddd

-- After Finding Big Palindrome --

ddmmmkkkmmmdd

**Q5: Reverse String:**

package com.java.paras;

import java.util.Scanner;

public class MOver

{

public static void main(String[] args)

{

/\*System.out.println("Enter Your String: ");

Scanner str=new Scanner(System.in);\*/

System.*out*.println("-- Befor String looks like: --");

String str="paras puru hello";

System.*out*.println(str);

System.*out*.println("-- After String looks like: --");

char ch=0;

for (int i = str.length()-1; i >=0; i--)

{

ch=str.charAt(i);

System.*out*.print(ch);

}

}

}

**Output:**

-- Befor String looks like: --

paras puru hello

-- After String looks like: --

olleh urup sarap

**Q6: Reverse String:**

package com.java.paras;

public class ReverseString

{

public static void main(String[] args)

{

System.*out*.println("-- Befor String looks like: --");

String st="paras puru hello";

System.*out*.println(st);

System.*out*.println("-- After String looks like: --");

String str[]=st.split(" ");

for (int i = str.length-1; i >=0; i--)

{

System.*out*.print(str[i]+" ");

}

}

}

Output:

-- Befor String looks like: --

paras puru hello

-- After String looks like: --

hello puru paras

**Q7: Reverse String Using Recursive Method:**

package com.java.paras;

import java.util.Scanner;

public class StringRecursiveReversal

{

String reverse = "";

public String reverseString(String str){

if(str.length() == 1)

{

return str;

}

else

{

reverse += str.charAt(str.length()-1) + reverseString(str.substring(0,str.length()-1));

return reverse;

}

}

public static void main(String a[])

{

StringRecursiveReversal srr = new StringRecursiveReversal();

System.*out*.println("-- Enter the String: --");

Scanner sc=new Scanner(System.*in*);

String str = sc.nextLine();

System.*out*.println("-- Reversed String is: --");

System.*out*.println(srr.reverseString(str));

}

}

**Output**:

-- Enter the String: --

paras puru

-- Reversed String is: --

urup sarap

**Q8: Reverse String Array Example:**

package com.java.paras;

public class ReverseStringArray

{

public static void main(String[] args)

{

System.*out*.println("-- Befor String looks like: --");

String st[]={"paras", "puru", "hello","malena"};

for (int i = 0; i < st.length; i++)

{

System.*out*.print(st[i]+" ");

}

System.*out*.println();

System.*out*.println("-- After String looks like: --");

for (int i = st.length-1; i >=0; i--)

{

System.*out*.print(st[i]+" ");

}

}

}

**Output**:

-- Befor String looks like: --

paras puru hello malena

-- After String looks like: --

malena hello puru paras

**Q9: OwnList Example:**

package com.java.JavaTest;

import java.util.Arrays;

class MyListt

{

private Object[] arr;

private int size=0;

public MyListt()

{

arr = new Object[10];

}

public Object get(int index)

{

if(index < size)

{

return arr[index];

}

else

{

throw new ArrayIndexOutOfBoundsException();

}

}

public Object remove(int index)

{

if(index < size)

{

Object obj=arr[index];

arr[index]=null;

int t=index;

while(t < size)

{

arr[t] = arr[t+1];

arr[t+1] = null;

t++;

}

size--;

return obj;

}

else

{

throw new ArrayIndexOutOfBoundsException();

}

}

public void add(Object obj)

{

if(arr.length-size<=5)

{

increaseLength();

}

arr[size++] = obj;

}

public int size()

{

return size;

}

public void increaseLength()

{

arr = Arrays.*copyOf*(arr, arr.length\*2);

}

}

public class OwnList

{

public static void main(String[] args)

{

MyList li=new MyList();

li.add("paras");

li.add("puru");

li.add("ani");

li.add("motu");

li.add("anil");

li.add("raj");

for (int i = 0; i < li.size(); i++)

{

System.*out*.println("All Element: "+li.get(i));

}

System.*out*.println("Remove Element: "+li.remove(2));

for (int i = 0; i < li.size(); i++)

{

S.*o*.println("After Removing Element: "+li.get(i));

}

}

}

**Output:**

All Element: paras

All Element: puru

All Element: ani

All Element: motu

All Element: anil

All Element: raj

Remove Element: ani

After Removing Element: paras

After Removing Element: puru

After Removing Element: motu

After Removing Element: anil

After Removing Element: raj

**Q10: Producer Consumer Problem:**

package com.java.paras;

class CubbyHole

{

int content;

boolean available;

public synchronized int get()

{

if(available==false)

{

try

{

wait();

}

catch (Exception e)

{

e.printStackTrace();

}

}

available = false;

notify();

return content;

}

public synchronized void put(int value)

{

if(available==true)

{

try

{

wait();

}

catch (Exception e)

{

e.printStackTrace();

}

}

content = value;

available = true;

notifyAll();

}

}

class Producer extends Thread

{

CubbyHole chole;

int value;

Producer(CubbyHole chole, int value)

{

super();

this.chole = chole;

this.value = value;

}

public void run()

{

for (int i = 0; i < 10; i++)

{

chole.put(value);

System.*out*.println("Producer: "+i+" put: "+i);

try

{

Thread.*sleep*(1000);

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

}

class Consumer extends Thread

{

CubbyHole chole;

int number;

Consumer(CubbyHole chole, int number)

{

super();

this.chole = chole;

this.number = number;

}

public void run()

{

int value;

for (int i = 0; i < 10; i++)

{

try

{

value=chole.get();

System.*out*.println("Consumer: "+i+" got: "+value);

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

}

public class PCSolution

{

public static void main(String[] args)

{

CubbyHole c=new CubbyHole();

Producer p1=new Producer(c, 1);

Consumer c1=new Consumer(c, 1);

p1.start();

c1.start();

}

}

**Output:**

Consumer #1 got: 0

Producer #1 put: 0

Producer #1 put: 1

Consumer #1 got: 1

Consumer #1 got: 2

Producer #1 put: 2

Producer #1 put: 3

Consumer #1 got: 3

Consumer #1 got: 4

Producer #1 put: 4

Consumer #1 got: 5

Producer #1 put: 5

Producer #1 put: 6

Consumer #1 got: 6

**Q11: Removing Special character from String:**

package com.java.paras;

public class RemoveWhiteSpace

{

public static void main(String[] args)

{

System.*out*.println("Before String looks like: ");

String str="%^%$^f53djhggk473hdgs?><?fhdfg889kh";

System.*out*.println(str);

System.*out*.println("After String looks like: ");

char ch[]=str.toCharArray();

for (int i = 0; i < ch.length; i++)

{

if((ch[i]>=65 && ch[i]<=91)||(ch[i]>=97 && ch[i]<=122))

{

System.*out*.print(ch[i]);

}

}

}

}

**Output**:

Before String Looks like:

%^%$^f53djhggk473hdgs?><?fhdfg889kh

After String looks like:

Fdjhggkhdgsfhdfgkh

**Q12: To Get Initial Character of Every word:**

package com.java.paras;

import java.io.IOException;

import java.util.Scanner;

public class InitailCharacter

{

public static void main(String[] args)throws IOException

{

Scanner sc=new Scanner(System.*in*);

String str;

char x;

int l;

System.*out*.println("-- Enter any sentence: --");

str=sc.nextLine();

str=" "+str;

str=str.toUpperCase();

l=str.length();

System.*out*.println("—- 1st character of every Words: --");

for(int i=0;i<l;i++)

{

x=str.charAt(i);

if(x==' ')

System.*out*.print(str.charAt(i+1)+" ");

}

}

}

**Output**:

-- Enter any sentence: --

Ranchi Bokaro Dhanbad Jamshedpur Hazaribagh

-- First character of every Words: --

R B D J H

**Q13: Exchange first and last character in a String:**

package com.java.paras;

public class ReplaceFirstAndLast

{

public static void main(String[] args)

{

String str="paras puru";

System.*out*.println("-- The String is: --");

System.*out*.println(str);

String st=ReplaceFirstAndLast.*frontBack*(str);

System.*out*.println("-- After Replacing 1st & last Character :-- ");

System.*out*.println(st);

}

public static String frontBack(String str)

{

if (str == null || str.length() == 0)

{

return str;

}

char[] cs = str.toCharArray();

char first = cs[0];

cs[0] = cs[cs.length -1];

cs[cs.length -1] = first;

return new String(cs);

}

}

**Output:**

-- The String is: --

india

-- After Replacing 1st & last Character is:--

andii

--------------------------------------

-- The String is: --

paras puru

-- After Replacing 1st & last Character is:--

uaras purp

**Q14: Find Shortest and Longest word in a Sentence:**

package com.java.paras;

import java.util.Scanner;

class ShortestAndLongestWord

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("-- Enter any sentence: -- ");

String str = sc.nextLine();

str = str + " ";

String x = "", maxw = "", minw = "";

char ch;

int p, maxl = 0, minl = str.length();

for (int i = 0; i < str.length(); i++)

{

ch = str.charAt(i);

if (ch != ' ')

{

x = x + ch;

}

else

{

p = x.length();

if (p < minl)

{

minl = p;

minw = x;

}

if (p > maxl)

{

maxl = p;

maxw = x;

}

x = "";

}

}

System.*out*.println("-- Shortest and Longest Word is: --");

System.*out*.println("Shortest word is = " +minw+ "\nLength is = " + minl);

System.*out*.println("Longest word is = " + maxw + "\nLength is = " + maxl);

}

}

**Output**:

-- Enter any sentence: --

All people love hindustan as a human being

-- Shortest and Longest Word is: --

Shortest word is = a

Length is = 1

Longest word is = hindustan

Length is = 9

**Q15: Swap two string without Third string:**

package com.java.paras;

import java.io.\*;

import java.util.Scanner;

public class SwapTwoStrings

{

public static void main (String[] args)throws IOException

{

Scanner sc=new Scanner(System.in);

System.out.println("-- Enter the 1st String :-- ");

String s1=sc.next();

int len1=s1.length();

System.out.println("-- Enter the 2nd String :--");

String s2=sc.next();

System.out.println("-------------------------------");

System.out.println("-- Strings Before Swapping :--");

System.out.println("1st String = "+s1);

System.out.println("2nd String = "+s2);

/\*Swapping Process Begins\*/

s1=s1+s2;

s2=s1.substring(0,len1);

s1=s1.substring(len1);

/\*Swapping Process Ends\*/

System.out.println("-------------------------------");

System.out.println("-- Strings After Swapping :--");

System.out.println("1st String = "+s1);

System.out.println("2nd String = "+s2);

}

}

**Output:**

Enter the 1st String :

paras

Enter the 2nd String :

puru

-------------------------------

Strings Before Swapping :

1st String = paras

2nd String = puru

-------------------------------

Strings After Swapping :

1st String = puru

2nd String = paras

**Q16: Before And After a Prime Number Program:**

package com.java.paras;

import java.util.Scanner;

public class BAPrimeNumber

{

public static void main(String[] args)

{

int i,j,pos=0;

System.*out*.println("Enter the prime no: ");

int n=new Scanner(System.*in*).nextInt();

//int n1=n+n; for After

//for (i = n+1; i<=n1; i++) for After

for(i=n-1;i>=2;i--)

{

pos=i;

for (j =2 ; j < i/2; j++)

{

if(i%j==0)

break;

}

if(j >= (i/2))

break;

}

if(n>=3)

System.*out*.println("Before " +n+ " : Prime no. is: "+pos);

else

System.*out*.println("please enter valid no..!!!!");

}

}

**Output:**

Enter the prime no:

12

Before 12 : Prime no. is: 11

-------------------------------

Enter the prime no:

11

After 11 : Prime no. is: 13

**Q17: Number To Words Program:**

package com;

import java.util.Scanner;

public class NumberToWord

{

public void pw(int n,String ch)

{

String one[]={" "," one"," two"," three"," four"," five"," six"," seven"," eight"," nine"," ten"," eleven"," twelve", " thirteen"," fourteen"," fifteen"," sixteen"," seventeen"," eighteen"," nineteen"};

String ten[]={" "," "," twenty"," thirty"," forty"," fifty"," sixty"," seventy"," eighty"," ninety"};

if(n > 19)

{

System.*out*.print(ten[n/10]+" "+one[n%10]);

}

else

{

System.*out*.print(one[n]);

}

if(n > 0)

System.*out*.print(ch);

}

public static void main(String[] args)

{

int n=0;

Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter an integer number: ");

n=sc.nextInt();

if(n <= 0)

{

System.*out*.println("Enter number greater than 0");

System.*exit*(0);

}

else

{

NumberToWord no=new NumberToWord();

System.*out*.println("After conversion number in words is: ");

no.pw((n/10000000)%100, " crore");

no.pw(((n/100000)%100), " lacks");

no.pw(((n/1000)%100), " thousand");

no.pw(((n/100)%10), " hundred");

no.pw(n%100, " ");

}

System.*out*.println(" Rupees Only");

}

}

Output:

Enter an integer number:

1256253

After conversion number in words is:

twelve lacks fifty six thousand two hundred fifty three Rupees Only

**Q18: Convert Number to Word:**

package com.java.paras;

import java.util.Scanner;

public class Number2Words

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*);

Number2Words call = new Number2Words();

System.*out*.print("Enter a number : ");

int n = sc.nextInt();

call.convert(n);

}

public void convert(int n)

{

int c;

if(n!=0)

{

c = n%10;

convert(n/10);

num2words(c);

}

}

public void num2words(int n)

{

String words[] ={"ZERO","ONE","TWO","THREE","FOUR","FIVE","SIX","SEVEN","EIGHT","NINE"};

System.*out*.print(words[n] +" ");

}

}

**Output**:

Enter a number : 0123456

ONE TWO THREE FOUR FIVE SIX

**Q19: Power of n Number:**

package com.java.JavaTest;

import java.util.Scanner;

public class PowerN

{

public static void main(String[] args)

{

System.*out*.println("Enter the Base: ");

Scanner sc=new Scanner(System.*in*);

int base=sc.nextInt();

System.*out*.println("Enter the power range: ");

int n=sc.nextInt();

*powerN*(base,n);

}

public static void powerN(int base, int n)

{

System.*out*.println("Power of "+base+" is:");

int result = 1;

int j=0;

for (int i = 0; i < n; i++)

{

j++;

result = result \* base;

System.*out*.println(base+"^"+j+" : "+result);

}

}

}

**Output**:

Enter the Base:

2

Enter the power range:

5

Power of 2 is:

2^1 : 2

2^2 : 4

2^3 : 8

2^4 : 16

2^5 : 32

### Q20: To find the sum of the following series using recursion : 1^2 + 2^2 + 3^2 +.......+ n^2:

package com.java.paras;

import java.util.Scanner;

public class SumOfSeries

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

SumOfSeries call = new SumOfSeries();

System.*out*.println("Enter value of n : ");

int n = sc.nextInt();

System.*out*.println("Sum of series = " +call.sum(n));

}

int sum(int n)

{

if(n==1)

return 1;

else

return (n\*n)+sum(n-1);

}

}

**Output**:

Enter value of n :

5

Sum of series = 55

### Q21: to find the sum of natural numbers upto a given number using recursion:

package com.java.paras;

import java.util.Scanner;

class SumOfNaturalNumbers

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.*in*);

SumOfNaturalNumbers call = new SumOfNaturalNumbers();

System.*out*.println("Enter the Range: ");

int n = sc.nextInt();

int sum=call.sum(n);

System.*out*.println("Sum of Natural Numbers till " +n+" :");

System.*out*.println(sum);

}

int sum(int n)

{

if(n==1)

return 1;

else

return n+sum(n-1);

}

}

**Output**:

Enter the Range:

6

Sum of Natural Numbers till 6 :

21

**Q21:Two Threads Run Sequentially Program:**

package com.java.JavaTest;

class TaskEvenOdd implements Runnable

{

private int max;

private Printer print;

private boolean isEvenNumber;

TaskEvenOdd(Printer print, int max, boolean isEvenNumber)

{

this.print = print;

this.max = max;

this.isEvenNumber = isEvenNumber;

}

@Override

public void run()

{

int number = isEvenNumber == true ? 2 : 1;

while(number <= max)

{

if(isEvenNumber)

{

print.printEven(number);

}

else

{

print.printOdd(number);

}

number+=2;

}

}

}

class Printer

{

boolean isOdd= false;

synchronized void printEven(int number)

{

while(isOdd == false)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.*out*.println("Even:"+number);

isOdd = false;

notifyAll();

}

synchronized void printOdd(int number)

{

while(isOdd == true)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.*out*.println("Odd:"+number);

isOdd = true;

notifyAll();

}

}

public class PrintEvenOddTester

{

public static void main(String ... args)

{

Printer print = new Printer();

Thread t1 = new Thread(new TaskEvenOdd(print, 10, false));

Thread t2 = new Thread(new TaskEvenOdd(print, 10, true));

t1.start();

t2.start();

}

}

**Output**:

Odd:1

Even:2

Odd:3

Even:4

Odd:5

Even:6

Odd:7

Even:8

Odd:9

Even:10

**Q22: Sorting Map By Key:**

package com.java.JavaTest;

import java.util.\*;

public class SBKey

{

public static void main(String[] args)

{

Map<String, String> map = new HashMap<String, String>();

map.put("2", "B");

map.put("1", "A");

map.put("4", "D");

map.put("3", "B");

map.put("7", "C");

map.put("5", "z");

map.put("6", "b");

map.put("8", "a");

System.*out*.println("Unsort Map......");

*printMap*(map);

System.*out*.println("Sorted Map......By Key");

Map<String, String> treeMap = new TreeMap<String, String>( map);

*printMap*(treeMap);

}

public static void printMap(Map<String, String> map)

{

for (Map.Entry entry : map.entrySet())

{

S.o.p("Key : " + entry.getKey() +" Value : "+entry.getValue());

}

}

}

**Outout**:

Unsort Map......

Key : 3 Value : B

Key : 2 Value : B

Key : 1 Value : A

Key : 7 Value : C

Key : 6 Value : b

Key : 5 Value : z

Key : 4 Value : D

Key : 8 Value : a

Sorted Map......By Key

Key : 1 Value : A

Key : 2 Value : B

Key : 3 Value : B

Key : 4 Value : D

Key : 5 Value : z

Key : 6 Value : b

Key : 7 Value : C

Key : 8 Value : a

**Q11: Sorting Map By Value:**

package com.java.JavaTest;

import java.util.\*;

public class SBValue

{

public static void main(String[] args)

{

Map<String, String> map = new HashMap<String, String>();

map.put("2", "B");

map.put("1", "A");

map.put("4", "D");

map.put("3", "B");

map.put("7", "C");

map.put("5", "z");

map.put("6", "b");

map.put("8", "a");

System.*out*.println("Unsort Map......");

*printMap*(map);

System.*out*.println("Sorted Map....By Value");

Map<String, String> sortedMap = *sortByComparator*(map);

*printMap*(sortedMap);

}

private static Map sortByComparator(Map map)

{

List list = new LinkedList(map.entrySet());

// sort list based on comparator

Collections.*sort*(list, new Comparator()

{

public int compare(Object o1, Object o2)

{

return ((Comparable) ((Map.Entry) (o1)).getValue()).compareTo(((Map.Entry) (o2)).getValue());

}

});

// put sorted list into map again

//LinkedHashMap make sure order in which keys were inserted

Map sortedMap = new LinkedHashMap();

for (Iterator it = list.iterator(); it.hasNext();)

{

Map.Entry entry = (Map.Entry) it.next();

sortedMap.put(entry.getKey(), entry.getValue());

}

return sortedMap;

}

public static void printMap(Map<String, String> map)

{

for (Map.Entry entry : map.entrySet())

{

System.*out*.println("Key : " + entry.getKey() + " Value : " + entry.getValue());

}

}

}

**Output:**

Unsort Map......

Key : 3 Value : B

Key : 2 Value : B

Key : 1 Value : A

Key : 7 Value : C

Key : 6 Value : b

Key : 5 Value : z

Key : 4 Value : D

Key : 8 Value : a

Sorted Map....By Value

Key : 1 Value : A

Key : 3 Value : B

Key : 2 Value : B

Key : 7 Value : C

Key : 4 Value : D

Key : 8 Value : a

Key : 6 Value : b

Key : 5 Value : z

**Q23: 2nd Way to sort a Map By Value in Descending Order**

package com.java.JavaTest;

import java.util.\*;

public class SortTesting

{

public static void main(String[] args)

{

HashMap<String,Double> map = new HashMap<String,Double>();

ValueComparator bvc = new ValueComparator(map);

TreeMap<String,Double> sorted\_map = new TreeMap<String,Double>(bvc);

map.put("A",99.5);

map.put("B",85.4);

map.put("C",66.4);

map.put("D",120.3);

map.put("D",50.3);

System.*out*.println("unsorted map: "+map);

sorted\_map.putAll(map);

System.*out*.println("results: "+sorted\_map);

}

}

class ValueComparator implements Comparator<String>

{

Map<String, Double> base;

public ValueComparator(Map<String, Double> base)

{

this.base = base;

}

// Note: this comparator imposes orderings that are inconsistent with equals.

public int compare(String a, String b)

{

if (base.get(a) >= base.get(b))

{

return -1;

}

else

{

return 1;

} // returning 0 would merge keys

}

}

Output:

unsorted map: {D=50.3, A=99.5, B=85.4, C=66.4}

results: {A=99.5, B=85.4, C=66.4, D=50.3}

**Q24: Remove duplicate value from Array:**

package com.java.paras;

import java.util.\*;

public class RemDupValFromArray

{

public static void main(String[] args)

{

String[] strArr = {"one","two","three","four","four","five"};

//convert string array to list

List<String> tmpList = Arrays.*asList*(strArr);

//create a treeset with the list, which eliminates duplicates

TreeSet<String> unique = new TreeSet<String>(tmpList);

System.*out*.println(unique);

}

}

**Output:**

[five, four, one, three, two]

**Q25: Remove duplicate value from ArrayList:**

package com.java.paras;

import java.util.\*;

public class RemDupFromList

{

public static void main(String[] args)

{

List li = new ArrayList();

li.add("one");

li.add("two");

li.add("three");

li.add("one");//Duplicate

li.add("one");//Duplicate

// We have facility to pass a List into Set constructor and vice verse to cast

// List li2 = new ArrayList(new HashSet(li));

// If you need to preserve the order use 'LinkedHashSet'

List li2 = new ArrayList(new LinkedHashSet(li));

Iterator it= li2.iterator();

while(it.hasNext())

{

System.*out*.println(it.next());

}

}

}

**Output**:

one

two

three

**Q26: Object sorting Example using Comparable and Comparator:**

package com.java.paras;

import java.util.\*;

public class ObjectSortingExample

{

public static void main(String args[])

{

Order ord1 = new Order(101,2000, "Sony");

Order ord2 = new Order(102,4000, "Hitachi");

Order ord3 = new Order(103,6000, "Philips");

List<Order> orders = new ArrayList<Order>();

orders.add(ord3);

orders.add(ord1);

orders.add(ord2);

System.*out*.println("Unsorted Collection : " + orders);

Collections.*sort*(orders);

System.*out*.println("List of Order object sorted in natural order : " + orders);

Collections.*sort*(orders, Collections.*reverseOrder*());

System.*out*.println("List of object sorted in descending order : " + orders);

Collections.*sort*(orders, new Order.OrderByAmount());

System.*out*.println("List of Order object sorted using Comparator - amount : " + orders);

Collections.*sort*(orders, new Order.OrderByCustomer());

System.*out*.println("Collection of Orders sorted using Comparator - by customer : " + orders);

}

}

class Order implements Comparable<Order>

{

private int orderId;

private int amount;

private String customer;

//Constructor

//Setter and Getter

public static class OrderByAmount implements Comparator<Order>

{

public int compare(Order o1, Order o2)

{

return o1.amount > o2.amount ? 1 : (o1.amount < o2.amount ? -1 : 0);

}

}

public static class OrderByCustomer implements Comparator<Order>

{

public int compare(Order o1, Order o2)

{

return o1.customer.compareTo(o2.customer);

}

}

public int compareTo(Order o)

{

return this.orderId > o.orderId ? 1 : (this.orderId < o.orderId ? -1 : 0);

}

public String toString()

{

return String.*valueOf*(orderId);

}

}

**Output:**

Unsorted Collection : [103, 101, 102]

List of Order object sorted in natural order : [101, 102, 103]

List of object sorted in descending order : [103, 102, 101]

List of Order object sorted using Comparator-by amount: [101,102,103]

Collection of Order sorted using Comparator-by customer: [102,103,101]

**Q26: Add Duplicate value in Set and User-Defined Program:**

package com.jlcindia.List;

//We can store any type of object and here we storing user defined object

import java.util.\*;

class Contact

{

String firstName;

String lastName;

int phone;

public Contact(String firstName, String lastName, int phone)

{

super();

this.firstName = firstName;

this.lastName = lastName;

this.phone = phone;

}

@Override

public String toString()

{

return "[fName=" + firstName + " lName=" + lastName +" phone=" + phone + "]";

}

}

public class UserDefinedObject

{

public static void main(String[] args)

{

Contact con1=new Contact("Paras", "puru", 901965);

Contact con2=new Contact("Paras", "puru", 901965);

Contact con3=new Contact("Ajit", "parivesh", 901965);

Contact con4=new Contact("Anil", "aryan", 901965);

Contact con5=new Contact("Prem", "mahto", 901965);

Contact con6=new Contact("Anil", "aryan", 901965);

Set<Contact> set=new HashSet<Contact>();

set.add(con1);

set.add(con2);

set.add(con3);

set.add(con4);

set.add(con5);

set.add(con6);

for (Contact contact : set)

{

System.*out*.println(contact);

}

}

}

**Output:**

[fName=Paras lName=puru phone=901965]

[fName=Anil lName=aryan phone=901965]

[fName=Prem lName=mahto phone=901965]

[fName=Anil lName=aryan phone=901965]

[fName=Ajit lName=parivesh phone=901965]

[fName=Paras lName=puru phone=901965]

**Q27: Custom Exception Program:**

package com.jlcindia.Exception;

import java.util.Scanner;

class CustomExceptionTest extends Exception

{

String s;

public CustomExceptionTest(String s)

{

this.s=s;

}

@Override

public String toString()

{

return "[" + s + "]";

}

}

public class MyException

{

public static void main(String[] args)

{

System.*out*.println("Enter the age:");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

try

{

*validate*(n);

}

catch (Exception e)

{

System.*out*.println("Exception occured: "+e);

}

finally

{

System.*out*.println("Execution completed..!!");

}

}

public static void validate(int age)throws CustomExceptionTest

{

if(age >= 18)

{

System.*out*.println("You are Eligible for vote");

}

else

{

throw new CustomExceptionTest("not Eligible for vote");

}

}

}

**Output**:

Enter the age:

18

You are Eligible for vote

Execution completed..!!

**Q28: Java program to check whether kaprekar no or Not?**

package com.java.paras;

import java.util.Scanner;

public class KaprekarNumber

{

public static void meth(int n)

{

int temp, k, a = 0, r, rem, quo;

temp = n;

k = n \* n;

while (n > 0)

{

r = n % 10;

a++;

n = n / 10;

}

rem = k % ((int) Math.*pow*(10, a));

quo = k / ((int) Math.*pow*(10, a));

if ((rem + quo) == temp)

System.*out*.println("This is kaprekar number..!!!");

else

System.*out*.println("This is not kaprekar number..!!");

}

public static void main(String ar[])

{

System.*out*.println("-- Enter the value: --");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

*meth*(n);

}

}

**Output**:

45 is Kaprekar number bcoz 45 square = 2025 = 20+25

-------------------------------------------------------

-- Enter the value: --

45

This is kaprekar number..!!!

-----------------------------------

-- Enter the value: --

10

This is not kaprekar number..!!

**Q29: Remove Duplicate element from Array:**

package com.java.paras;

import java.util.Scanner;

public class DublicatesRemove

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("-- Enter size of the array: --");

int l = sc.nextInt();

int[] a = new int[l];

// insert elements in the array logic

System.*out*.println("-- Enter the element: --");

for (int i = 0; i < l; i++)

{

int el = sc.nextInt();

a[i] = el;

}

System.*out*.println("Before Removing Duplicate element");

for (int i = 0; i < a.length; i++)

{

System.*out*.println(a[i]);

}

// sorting elements in the array logic

for (int i = 0; i < l; i++)

{

for (int j = 0; j < l - 1; j++)

{

if (a[j] > a[j + 1])

{

int temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

System.*out*.println("After Removing Duplicate element");

// remove duplicate elements logic

int b = 0;

a[b] = a[0];

for (int i = 1; i < l; i++)

{

if (a[b] != a[i])

{

b++;

a[b]=a[i];

}

}

for(int i=0;i<=b;i++)

{

System.*out*.println(a[i]);

}

}

}

**Output:**

-- Enter size of the array: --

7

-- Enter the element: --

2 2 4 3 5 4 7

Before Removing Duplicate element

2 2 4 3 5 4 7

After Removing Duplicate element

2 3 4 5 7

**Q30: Store Different Type of element in Array**

package com;

public class ArraySize

{

public static void main(String[] args)

{

Object obj[]=new Object[5];

obj[0]="Paras";

obj[1]=34;

obj[2]=75638485;

obj[3]=546.f;

obj[4]='A';

System.*out*.println("The element in Array is:");

for (int i = 0; i < obj.length; i++)

{

System.*out*.println(obj[i]);

}

Object obj1[]=new Object[8];

//Increasing the length of the Array

for (int i = 0; i < obj.length; i++)

{

obj1[i]=obj[i];

}

//Storing the element in new Array

System.*out*.println("After Storing element in new Array: ");

obj1[6]="-----";

obj1[7]="puru";

obj1[8]=57382949;

obj=obj1;

for (int i = 0; i < obj.length; i++)

{

System.*out*.println(obj[i]);

}

}

}

**Output:**

The element in Array is:

Paras

34

75638485

546.0

A

After Storing the element in new Array:

Paras

34

75638485

546.0

A

-----

puru

57382949

**Q31: Prime Number Validation:**

package com.java.paras;

import java.util.Scanner;

public class PrimeTest

{

public static void main(String[] args)

{

System.*out*.println("-- Enter the Number: --");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

if(*isPrime*(n))

{

System.*out*.println("This is prime no..!!");

}

else

{

System.*out*.println("Sorry..This is not Prime No..!!");

}

}

public static boolean isPrime(int n)

{

if(n <= 1)

{

return false;

}

for (int i = 2; i < n; i++)

{

if (n%i == 0)

{

return false;

}

}

return true;

}

}

**Output:**

-- Enter the Number: --

1

Sorry..Plz enter valid prime No..!!

------------------------------

-- Enter the Number: --

2

This is prime no..!!

**Q32: Prime number validation using Recursion:**

package com.java.paras;

import java.util.Scanner;

public class PrimeUsingRecurxion

{

public static void main(String[] args)

{

System.*out*.println("-- Enter the value: --");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

if(n <= 1)

{

System.*out*.println("Plz enter greater than 1..!!!");

return;

}

if(*isPrime*(n,n -1))

{

System.*out*.println(n + ": is a Prime No..!!!");

}

else

{

System.*out*.println(n + ": is Not a Prime No..!!");

}

}

static boolean isPrime(int num, int div)

{

if(div <= 1)

{

return true;

}

if(num % div == 0)

{

return false;

}

return *isPrime*(num, div-1);

}

}

**Output:**

-- Enter the value: --

1

Plz enter greater than 1..!!!

-------------------------------

-- Enter the value: --

2

2: is a Prime No..!!!

-------------------------------

-- Enter the value: --

4

4: is Not a Prime No..!!

**Q33: Prime number generate between 1 to N number:**

import java.util.Scanner;

public class PrimeNumberExample

{

public static void main(String args[])

{

System.*out*.println("Enter the Range for prime number: ");

int limit = new Scanner(System.*in*).nextInt();

System.*out*.println("prime number from 1 to " + limit);

for(int number = 2; number<=limit; number++)

{

//print prime numbers only

if(*isPrime*(number))

{

System.*out*.println(number);

}

}

}

public static boolean isPrime(int number)

{

for(int i=2; i<number; i++)

{

if(number%i == 0)

{

return false; //number is divisible so its not prime

}

}

return true; //number is prime now

}

}

**Output:**

Enter the Range for prime number:

20

prime number from 1 to 20

2 3 5 7 11 13 17 19

**Q34: Sum of First 5 Prime numbers are:**

package com.java.JavaTest;

public class SumOfPrimeNumbers

{

public static void main(String[] args)

{

long result = 0;

int number = 2;

int count = 0;

while (count < 5)

{

if (*checkPrime*(number) == true)

{

result = result + number;

count++;

}

number++;

}

System.*out*.println("Sum of first 5 prime numbers is: " +result);

}

public static boolean checkPrime(int number)

{

for (int i = 2; i < number; i++)

{

if (number % i == 0)

{

return false;

}

}

return true;

}

}

**Output:**

Sum of first 5 prime numbers is: 28

**Q35: Armstrong number validation program:**

package com.java.paras;

import java.util.\*;

public class ArmstrongTest

{

public static void main(String args[])

{

System.*out*.println("Enter 3 digit no. for Armstrong Test:");

int number = new Scanner(System.*in*).nextInt();

if(*isArmStrong*(number))

{

System.*out*.println(number + ": is an Armstrong number");

}

else

{

System.*out*.println(number + ": is not an Armstrong number");

}

}

private static boolean isArmStrong(int number)

{

int result = 0;

int orig = number;

while(number != 0)

{

int rem = number%10;

result = result + rem\*rem\*rem;

number = number/10;

}

if(orig == result)

{

return true;

}

return false;

}

}

**Output**:

Enter 3 digit no. for Armstrong Test:

153

153: is an Armstrong number

---------------------

Enter 3 digit no. for Armstrong Test:

123

123: is not an Armstrong number

**Q36: Armstrong number validation using Recursion:**

package com.java.paras;

import java.util.\*;

public class ArmstrongUsingRecursion

{

public static void main(String[] args)

{

ArmstrongUsingRecursion obj= new ArmstrongUsingRecursion();

int n,m;

Scanner sc=new Scanner(System.*in*);

System.*out*.println("-- Enter a number: -- ");

n=sc.nextInt();

m=obj.checknum(n);

if(n==m)

{

System.*out*.println("This is a armstrong number");

}

else

{

System.*out*.println("This is not a armstrong number");

}

}

int checknum(int n)

{

if(n==0)

return 0;

else

return (int)Math.*pow*(n%10,3)+ checknum(n/10);

}

}

**Output**:

-- Enter a number: --

153

This is a armstrong number

--------------------------

-- Enter a number: --

25

This is not a armstrong number

**Q37: Generate Armstrong number between 1 to N:**

**1st Way:**

package com.java.paras;

public class ArmsstrongNumbers

{

public static void main(String args[])

{

int i=0,j=0,k=0,sum=0,num=0,digit=0;

for(i=0;i<1000;i++)

{

if(i<=9)

{

j=1;//assign if the number is a single digit no.

}

else if(i<=99)

{

j=2;//assign if the number is a two digit number

}

else

{

j=3;//assign if the number is a three digit no.

}

sum=0;num=i;

for(k=1;k<=j;k++)//calculate the sum of cube of digits

{

digit = num%10;

num /=10;

sum +=digit\*digit\*digit;

}

if(sum == i)//if sum of cubes of digits are equal then

{

System.*out*.println(i+" is an Armstrong Number!");

}

}

}

}

**2nd Way:**

package com.java.paras;

public class ArmstrongNumber

{

public static void main(String[] args)

{

System.*out*.println("List of Armstrong No bet(100 - 999):");

for(int i = 0; i <= 999; i++)

{

int a = i / 100;

int b = (i - a \* 100) / 10;

int c = (i - a \* 100 - b \* 10);

int d = a\*a\*a + b\*b\*b + c\*c\*c;

if(i == d)

System.*out*.format("%d\n", i);

}

}

}

**Output:**

List of Armstrong Numbers between (100 - 999):

0

1

153

370

371

407

**Q38: Perfect number validation program:**

Perfect Number-  If the sum of all the factors of a number is equal to the number itself then it is a perfect number. This program finds whether a given number is perfect number or not.

Example:- 6 = 1+2+3

28 = 1+2+4+7+14

package com.java.paras;

import java.util.Scanner;

public class PerfectNumber

{

public static void main(String[] args)

{

int n,s=0;

System.*out*.println("Enter a number: ");

Scanner sc=new Scanner(System.*in*);

n=sc.nextInt();

for(int i=1;i<n;i++)

if(n%i==0)

s=s+i;

if(s==n)

System.*out*.println("it is a perfect no...!!!");

else

System.*out*.println("it is not a perfect no...!!!");

}

}

**Output:**

Enter a number:

28

it is a perfect no...!!!

---------------------------

Enter a number:

10

it is not a perfect no...!!!

**Q39: Validate power of 2 in java:**

package com.java.paras;

import java.util.Scanner;

public class FindPowerOfTwo

{

public static void main(String[] args)

{

System.*out*.println("-- Enter the Number: --");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

if(*checkIfItsPowerOf2*(n))

{

System.*out*.println("This no is power of 2 in java");

}

else

{

System.*out*.println("Sorry..This is not a power of 2");

}

}

public static boolean checkIfItsPowerOf2(int n)

{

if( (n & -n) == n)

{

return true;

}

else

{

return false;

}

}

}

**Output: Power of 2 are: 2 4 8 16 32...**

-- Enter the Number: --

32

This number is power of Two in java

-- Enter the Number: --

3

Sorry..This is not a power of Two

**Q40: Fibonacci Series:**

package com.java.paras;

import java.util.Scanner;

public class FibonnaciSeries

{

public static void main(String args[])

{

Scanner scan= new Scanner(System.*in*);

System.*out*.println("Please Enter a No.");

int n=scan.nextInt();

System.*out*.println("Fibonnaci Series are: ");

int prev, next, sum, i;

prev=next=1;

for(i=1;i<=n;i++)

{

System.*out*.print(prev+" ");

sum=prev+next;

prev=next;

next=sum;

}

}

}

**Output**:

Please Enter a No.

10

Fibonnaci Series are:

1 1 2 3 5 8 13 21 34 55

**Q41: Fibonacci Series Using Recursion:**

package com.java.paras;

import java.util.Scanner;

public class RecursionFibonaci

{

public static void main(String[] args)

{

System.*out*.println("Enter the Range: ");

int n = new Scanner(System.*in*).nextInt();

System.*out*.println("Fibonacci series upto " +n+" numbers : ");

for(int i=1; i<=n; i++)

{

System.*out*.print(*fibonacci*(i) +" ");

}

}

public static int fibonacci(int number){

if(number == 1 || number == 2){

return 1;

}

return *fibonacci*(number-1) + *fibonacci*(number -2);

}

}

**Output**:

Enter the Range:

10

Fibonacci series upto 10 numbers :

1 1 2 3 5 8 13 21 34 55

**Q42: Tribonacci Series:**

A tribonacci series is one in which the sum next term is the sum of previous three terms eg. 0    1   2  3    6     11    20………….

package com.java.paras;

import java.util.Scanner;

public class Tribonacci

{

public static void main(String[] args)

{

System.*out*.println("Enter a word : ");

Scanner sc=new Scanner(System.*in*);

int n = sc.nextInt();

Tribonacci tri=new Tribonacci();

tri.func(n);

}

void func(int n)

{

int a=0, b=1, c= 2, d=0, i;

System.*out*.println("Tribonacci Series are:");

System.*out*.print( a +", "+ b + ", "+ c);

for(i=4;i<=n;i++)

{

d=a+b+c;

System.*out*.print(" , " + d);

a=b;

b=c;

c=d;

}

}

}

**Output:**

Enter a word :

10

Tribonacci Series are:

0, 1, 2 , 3 , 6 , 11 , 20 , 37 , 68 , 125

**Q43: To print possible sum combinations of a number.**

package com.java.paras;

import java.util.Scanner;

public class NumberCombinations

{

public static void main(String[] args)

{

System.*out*.println("Enter a number : ");

Scanner sc=new Scanner(System.*in*);

int n = sc.nextInt();

NumberCombinations nc=new NumberCombinations();

System.*out*.println("Number Combination may be: ");

nc.makeCombinations(n);

}

void makeCombinations(int n)

{

int sum,x,y,z,p;

for(x=1 ; x<=9 ; x++)

{

for(y=x ; y<=9 ; y++)

{

sum =x+y;

if(sum==n)

System.*out*.println(x + " + " + y);

for(z=1 ; z<=9 ; z++)

{

sum=sum+z;

if(sum<n)

continue;

else

break;

}

if(sum==n)

{

System.*out*.print(x + " + " + y);

for(p=1 ; p<=z ; p++)

{

System.*out*.print(" + " + p);

}

System.*out*.println();

}

}

}

}

}

**Output:**

Enter a number :

7

Number Combination may be:

1 + 3 + 1 + 2

1 + 5 + 1

1 + 6

2 + 2 + 1 + 2

2 + 4 + 1

2 + 5

3 + 3 + 1

3 + 4

**Q44: Factorial Number:**

import java.util.Scanner;

public class Factorial

{

public static void main(String[] args)

{

System.*out*.println("Please enter the value: ");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

int f=1;

for (int i = 1; i <= n; i++)

{

f=f\*i;

}

System.*out*.println("Factorial of 5 is: "+f);

}

}

**Output:**

Please enter the value:

5

Factorial of 5 is: 120

**Q45: Factorial Number Using Recursion:**

package com.java.paras;

import java.util.Scanner;

public class FactorialRecursion

{

public static void main(String[] args)

{

System.*out*.println("Enter the Range: ");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

System.*out*.println("The factoral of "+n+" is:");

int fact=*factorial*(n);

System.*out*.println(fact);

}

public static int factorial(int f)

{

if (f==0)

return 0;

else if(f==1)

return 1;

return f\**factorial*(f-1);

}

}

**Output:**

Enter the Range:

5

The factoral of 6 is:

120

**Q46: GCD of two number using Recursion:**

package com.java.paras;

import java.util.Scanner;

public class GCD

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter 1st Value: ");

int num1=sc.nextInt();

System.*out*.println("Enter 2nd Value: ");

int num2=sc.nextInt();

int res=*findGCD*(num1,num2);

System.*out*.println("GCD of 2 Number is: ");

System.*out*.println(res);

}

public static int findGCD(int num1, int num2)

{

if(num2 == 0)

{

return num1;

}

return *findGCD*(num2, num1 % num2);

}

}

**Output**:

Enter 1st Value:

30

Enter 2nd Value:

20

GCD of 2 Number is:

10

**Q47: Repeated Character Count:**

package com.java.paras;

public class RepeatedCharCount

{

public static void main(String[] args)

{

System.*out*.println("The String looks like: ");

String str="paras1puru1anilanni";

System.*out*.println(str);

char ch[]=str.toCharArray();

boolean found;

int count=0;

System.*out*.println("--Repeated Character & Digits are:-- ");

for (int i = 0; i < ch.length; i++)

{

found = false;

count=0;

for (int j = i-1; j >=0; j--)

{

if(ch[i]==ch[j])

{

found=true;

break;

}

}

if(found)

continue;

for (int k = 0; k < ch.length; k++)

{

if(ch[k]==ch[i])

count++;

}

System.*out*.println(ch[i]+ "---> " +count+ " Times");

}

}

}

**Ouput**:

The String looks like:

String str="paras441pu23ru1anilanni";

-- Repeated Character are: --

The String looks like:

paras441pu23ru1anilanni

-- Repeated Character are: --

p---> 2 Times

a---> 4 Times

r---> 2 Times

s---> 1 Times

4---> 2 Times

1---> 2 Times

u---> 2 Times

2---> 1 Times

3---> 1 Times

n---> 3 Times

i---> 2 Times

l---> 1 Times

**Q48: Move Each character to Next Position:**

package com.java.paras;

import java.util.Scanner;

public class CharacterMove

{

public static void main(String[] args)

{

System.*out*.println("Enter a String: ");

Scanner sc=new Scanner(System.*in*);

String str = sc.next();

CharacterMove cm=new CharacterMove();

System.*out*.println("Move Each character to Next Position: ");

String st = cm.*move*(str);

System.*out*.println(st);

}

public static String move(String input)

{

char[] arr = input.toCharArray();

*swap*(arr, 0, arr.length-1);

int i = arr.length - 1;

while(i > 1)

{

*swap*(arr, i, i-1);

i--;

}

return new String(arr);

}

private static void swap(char[] arr, int x, int y)

{

char c = arr[x];

arr[x] = arr[y];

arr[y] = c;

}

}

**Output**:

Enter a String:

paraspuru

Move Each character to Next Position:

uparaspur

**Q49: String Sorting:**

package com.java.paras;

public class StringSorting

{

public static void main(String[] args)

{

String str[]=new String[]{“paras","puru”,”nil","omp","pem"};

System.*out*.println("-- String before sorting: --");

for (int i = 0; i < str.length; i++)

{

System.*out*.print(str[i]+"\t");

}

for (int i = 0; i < str.length; i++)

{

for (int j = i+1; j < str.length; j++)

{

if(str[j].compareTo(str[i])<0)

{

String temp=str[i];

str[i]=str[j];

str[j]=temp;

}

}

}

System.*out*.println();

System.*out*.println("-- Sorted String are: --");

for (int i = 0; i < str.length; i++)

{

System.*out*.print(str[i]+"\t");

}

}

}

**Output**:

-- String before sorting: --

paras puru nil omp pem

-- Sorted String are: --

nil omp paras pem puru

### Q50: to arrange the letters of a word in alphabetical order:

package com.java.paras;

import java.util.Scanner;

public class AlphabeticalOrder

{

static String *n*;

static int *size*;

public static void main(String[] args)

{

System.*out*.println("Enter a word : ");

Scanner sc=new Scanner(System.*in*);

AlphabeticalOrder call = new AlphabeticalOrder();

*n* = sc.next();

*size* = *n*.length();

call.alphabetical();

}

public void alphabetical()

{

char b[] = new char[*size*];

for(int i=0;i<*size*;i++)

b[i] = *n*.charAt(i);

char t;

for(int j=0;j<*size*-1;j++)

{

for(int k=0;k<*size*-1-j;k++)

{

if(b[k]>b[k+1])

{

t=b[k];

b[k]=b[k+1];

b[k+1]=t;

}

}

}

System.*out*.println("Original word is:: " +*n*);

System.*out*.print("Sorted word is:: ");

for(int m=0;m<*size*;m++)

System.*out*.print(b[m]);

}

}

**Output:**

Enter a word :

paraspuru

Original word is:: paraspuru

Sorted word is:: aapprrsuu

-----------------------------

Enter a word :

WelCome

Original word is:: WelCome

Sorted word is:: CWeelmo

**Q51: Number Sorting Example:**

package com.java.paras;

public class SortingNumber

{

public static void main(String[] args)

{

int arr[]={12,10,15,20,30,05,01};

System.*out*.println("-- String before sorting: --");

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

{

System.*out*.print(arr[i]+"\t");

}

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

{

for (int j = i+1; j < arr.length; j++)

{

if(arr[i] >= arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

System.*out*.println();

System.*out*.println("-- Sorted String are: --");

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

{

System.*out*.print(arr[i]+"\t");

}

}

}

**Output**:

-- String before sorting: --

12 10 15 20 30 5 1

-- Sorted String are: --

1 5 10 12 15 20 30

**Q52: Bubble Sorting Example:**

import java.util.Scanner;

public class BubbleSort

{

public static void main(String[] args)

{

int a[]=new int[10];

System.*out*.println("Enter the number: ");

int n=new Scanner(System.*in*).nextInt();

System.*out*.println("Enter the element one by one: ");

for (int i = 0; i < n; i++)

{

a[i]=new Scanner(System.*in*).nextInt();

}

*bubbleSort*(a,n);

System.*out*.println("Sorted element are: ");

for (int i = 0; i < n; i++)

{

System.*out*.print(a[i]+" ");

}

}

public static void bubbleSort(int[] a, int n)

{

for (int j = 1; j < n; j++)

{

for (int i = 0; i< n-j; i++)

{

if(a[i] >= a[i+1])

{

int temp=a[i];

a[i]=a[i+1];

a[i+1]=temp;

}

}

}

}

}

**Output**:

Enter the number:

5

Enter the element one by one:

20 12 10 30 50

Sorted element are:

10 12 20 30 50

**Q53: Insertion Sort Program:**

package com.java.paras;

public class InsertionSort

{

public static void main(String[] args)

{

System.*out*.println("-- Array Elements are: --");

int[] arr1 = {10,34,2,56,7,67,88,42};

for (int i = 0; i < arr1.length; i++)

{

System.*out*.print(arr1[i]+" ");

}

int[] arr2 = *doInsertionSort*(arr1);

System.*out*.println("\n-- Sorted Elements are: --");

for(int i:arr2)

{

System.*out*.print(i+" ");

}

}

public static int[] doInsertionSort(int[] arr)

{

int temp;

for (int i = 1; i < arr.length; i++)

{

for(int j = i ; j > 0 ; j--)

{

if(arr[j] < arr[j-1])

{

temp = arr[j];

arr[j] = arr[j-1];

arr[j-1] = temp;

}

}

}

return arr;

}

}

**Output:**

-- Array Elements are: --

10 34 2 56 7 67 88 42

-- Sorted Elements are: --

2 7 10 34 42 56 67 88

**Q54: Selection Sort Program:**

package com.java.paras;

import java.util.Scanner;

public class SelectionSort

{

public static void main(String a[])

{

int arr[]=new int[10];

Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter the no. of elements to sort: ");

int n=Integer.*parseInt*(sc.next());

System.*out*.println("Enter "+n+" elements:");

for (int i = 0; i < n; i++)

{

arr[i] = sc.nextInt();

}

*doSelectionSort*(arr,n);

System.*out*.println("\n-- Sorted Elements are: --");

for (int i = 0; i < n; i++)

{

System.*out*.print(arr[i]+" ");

}

}

public static void doSelectionSort(int[] arr,int n)

{

int pos, small, temp;

for (int i = 0; i < n - 1; i++)

{

small = arr[i];

pos = i;

for (int j = i + 1; j < n; j++)

{

if (arr[j] < small)

{

small = arr[j];

pos = j;

}

temp = arr[pos];

arr[pos] = arr[i];

arr[i] = temp;

}

}

}

}

**Output**:

Enter the no. of elements to sort:

5

Enter 5 elements:

12 80 50 33 20

-- Sorted Elements are: --

12 20 33 50 80

**Q55: Linear Search Program:**

import java.util.Scanner;

public class LinearSearch

{

public static void main(String[] args)

{

int key,pos;

int a[]=new int[10];

System.*out*.println("Enter the number: ");

int n=new Scanner(System.*in*).nextInt();

System.*out*.println("Enter element one by one: ");

for (int i = 0; i < n; i++)

{

a[i]=new Scanner(System.*in*).nextInt();

}

System.*out*.println("please enter the key");

key=new Scanner(System.*in*).nextInt();

pos=*linearSearch*(key,a,n);

if(pos==0)

{

System.*out*.println("Key not found");

}

else

{

System.*out*.println("Key found at position: "+pos);

}

}

public static int linearSearch(int key,int[] a, int n)

{

for (int i = 0; i <n; i++)

{

if(key == a[i])

{

return i+1;

}

}

return 0;

}

}

**Output:**

Enter the number:

5

Enter the element one by one:

30 20 10 40 60

please enter the key

20

Key found at position: 2

**Q56: Remove Duplicate Value from Array without HashSet:**

package com.java.paras;

import java.util.Arrays;

public class RemoveDuplicateValue

{

public static void main(String args[])

{

int arr[] = {1,4,5,4,1,2,3,5,9,7,12,-5,1,4,-1,-5,12,1};

System.*out*.println("The Array looks like: ");

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

{

System.*out*.print(arr[i]+" ");

}

Arrays.*sort*(arr);

*removieDuplicates*(arr);

}

public static void removieDuplicates(int[] arr)

{

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

newArr[0] = arr[0];

int noDupCount = 1;

System.*out*.println();

System.*out*.println("After Removing Duplicate Value: ");

for (int c = 1; c < arr.length-1; c++)

{

boolean bool = false;

for (int d = 0; d < newArr.length-1; d++)

{

if (newArr[d] == arr[c])

{

bool = true;

}

}

if(!bool)

{

newArr[noDupCount++] = arr[c];

}

}

int[] newUniques = new int[noDupCount];

for(int i = 0; i < noDupCount; i++)

{

newUniques[i] = newArr[i];

System.*out*.print(newArr[i]+" ");

}

}

}

**Output:**

The Array looks like:

1 4 5 4 1 2 3 5 9 7 12 -5 1 4 -1 -5 12 1

After Removing Duplicate Value:

-5 -1 1 2 3 4 5 7 9 12

**Q57: Find Largest and Smallest Number from Array:**

package com.java.paras;

public class FindLargestSmallestNumber

{

public static void main(String[] args)

{

System.*out*.println("The Array element are: ");

int numbers[] = new int[]{32,43,53,54,32,65,63,98,43,23};

for (int i = 0; i < numbers.length; i++)

{

System.*out*.print(numbers[i]+" ");

}

int smallest = numbers[0];

int largest = numbers[0];

for(int i=1; i< numbers.length; i++)

{

if(numbers[i] > largest)

largest = numbers[i];

else if (numbers[i] < smallest)

smallest = numbers[i];

}

System.*out*.println();

System.*out*.println("Largest Number is : " + largest);

System.*out*.println("Smallest Number is : " + smallest);

}

}

**Output:**

The Array elements are:

32 43 53 54 32 65 63 98 43 23

Largest Number is : 98

Smallest Number is : 23

**Q58: Number of Occurrences of a character in a String:**

package com.java.paras;

public class CharOccuranceTimes

{

public static void main(String args[])

{

System.*out*.println("-- The String is: --");

String str = "Today is Monday";//count number of "a" on String

System.*out*.println(str);

System.*out*.println("Number of character 'a' on String is:");

int charCount = 0;

for(int i =0 ; i<str.length(); i++)

{

if(str.charAt(i) == 'a')

{

charCount++;

}

}

System.*out*.println(charCount+ " Times");

}

}

**Output:**

-- The String is: --

Today is Monday

Number of character 'a' on String is:

2 Times

**Q59: Converting DecimalToBinary Program:**

package com.java.paras;

import java.util.Scanner;

public class DecimalToBinary

{

public void printBinaryFormat(int number)

{

int binary[] = new int[25];

int index = 0;

while(number > 0)

{

binary[index++] = number % 2;

number = number / 2;

}

for(int i = index-1;i >= 0;i--)

{

System.*out*.print(binary[i]);

}

}

public static void main(String a[])

{

System.*out*.println("-- Enter the Decimal Value: --");

Scanner sc=new Scanner(System.*in*);

int n=sc.nextInt();

DecimalToBinary dtb = new DecimalToBinary();

System.*out*.println("-- Binary Format are: --");

dtb.printBinaryFormat(n);

}

}

**Output:**

-- Enter the Decimal Value: --

10

-- Binary Format are: --

1010

**Q60: Converting BinaryToDecimal Program:**

package com.java.paras;

import java.util.Scanner;

public class Binarytodecimal

{

public static void main(String args[])

{

int bin, dec=0,k=1;

Scanner scanner=new Scanner(System.*in*);

System.*out*.println("Enter the binary number : ");

bin=scanner.nextInt();

while(bin!=0)

{

dec=dec+ (bin%10)\*k;

k\*=2;

bin/=10;

}

System.*out*.println("Decimal number is = "+dec);

}

}

**Output**:

Enter the binary number:

1010

Decimal number is = 10

Enter the binary number :

23

Decimal number = 7

**Q61: Program to check given number is Binary or Not:**

package com.java.paras;

import java.util.Scanner;

public class MyBinaryCheck

{

public boolean isBinaryNumber(int binary)

{

boolean status = true;

while(true)

{

if(binary == 0)

{

break;

}

else

{

int tmp = binary%10;

if(tmp > 1)

{

status = false;

break;

}

binary = binary/10;

}

}

return status;

}

public static void main(String a[])

{

MyBinaryCheck mbc = new MyBinaryCheck();

S.o.p("Is 1000111 binary? :"+mbc.isBinaryNumber(1000111));

S.o.p("Is 10309111 binary? :"+mbc.isBinaryNumber(10309111));

}

}

**Output**:

Is 1000111 binary? :true

Is 10309111 binary? :false

**Q62: Remove Repeated Character from String:**

package com.java.paras;

import java.util.Scanner;

public class RemoveDupChar

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("-- Enter any word: --");

String str = sc.next();

str = str + " ";

String ans="";

char ch1,ch2;

System.*out*.println("After removing repeated characters = ");

for(int i=0; i<str.length()-1; i++)

{

ch1=str.charAt(i);

ch2=str.charAt(i+1);

if(ch1!=ch2)

{

ans = ans + ch1;

}

}

System.*out*.println(ans);

}

}

**Output**:

-- Enter the string: --

paarrass

After removing repeated characters :

paras

**2nd Way:**

---------------

package com.java.paras;

public class RemoveCharacterFromString

{

public static void main(String[] args)

{

System.*out*.println("-- The String is: --");

String str="My name is paras puru";

System.*out*.println(str);

System.*out*.println("After Removing char A from String");

*removeCharInString* (str, 'a');

}

public static void removeCharInString (String str, char charToBeRemoved)

{

if (str == null)

return;

for (int i = 0; i < str.length (); i++)

{

char ch = str.charAt (i);

if (ch == charToBeRemoved)

continue;

System.*out*.print(ch);

}

}

}

**Output:**

-- The String is: --

My name is paras puru

-- After Removing char A from String --

My nme is prs puru

**Q63: Differenciate Day Between two dates:**

package com.java.paras;

import java.io.\*;

public class Datediff

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.*in*));

System.*out*.println("enter the first date:");

System.*out*.print("enter the days=");

int d1=Integer.*parseInt*(br.readLine());

System.*out*.print("enter the month=");

int m1=Integer.*parseInt*(br.readLine());

System.*out*.print("enter the year=");

int y1=Integer.*parseInt*(br.readLine());

System.*out*.println("enter second date after first date:");

System.*out*.print("enter the days=");

int d2=Integer.*parseInt*(br.readLine());

System.*out*.print("enter the month=");

int m2=Integer.*parseInt*(br.readLine());

System.*out*.print("enter the year=");

int y2=Integer.*parseInt*(br.readLine());

int c,i,j,t=0;

int a[]={31,29,31,30,31,30,31,31,30,31,30,31};

int b[]={31,28,31,30,31,30,31,31,30,31,30,31};

c=y2-y1;

for(i=0;i<c;i++)

{

for(j=m2-1;j>=0;j--)

{

if(y2%4==0)

t+=a[j];

else

t+=b[j];

}

m2=12;

}

for(i=m2-1;i>=m1;i--)

{

if(y1%4==0)

t+=a[i];

else

t+=b[i];

}

t+=(d2-d1);

System.*out*.println("Difference of days between 2 date:"+t);

}

}

**Output:**

enter the first date:

enter the days=10

enter the month=1

enter the year=2013

enter second date after the first date:

enter the days=20

enter the month=1

enter the year=2013

Difference of days between two dates: 10

**Q64: String Anagram Test:**

Import com.java.paras;

import java.util.Scanner;

public class Anagram

{

public static void main(String[] args)

{

System.*out*.println("-- Enter the 1st String: -- ");

String str1=new Scanner(System.*in*).nextLine();

System.*out*.println("-- Enter the 2nd String: --");

String str2=new Scanner(System.*in*).nextLine();

if(*Method*(str1,str2))

{

System.*out*.println("Entered String are Anagram..!!!");

}

else

{

System.*out*.println("Sorry..This is not an Anagram!!");

}

}

public static boolean Method(String str3, String str4)

{

String str5=str3.toLowerCase();

String str6=str4.toLowerCase();

char ch1[]=str5.toCharArray();

char ch2[]=str6.toCharArray();

Arrays.*sort*(ch1);

Arrays.*sort*(ch2);

return Arrays.*equals*(ch1, ch2);

}

}

**Output:**

-- Enter the 1st String: --

paras puru

-- Enter the 2nd String: --

sapar uurp

Entered String are an Anagram..!!!

-- Enter the 1st String: --

Srinivas

-- Enter the 2nd String: --

rishkzsn

Sorry..This is not an Anagram!!!

**Q65: To Get distinct element & avoiding duplicate value from Array:**

package com.java.paras;

public class MyDisticntElements

{

public static void printDistinctElements(int[] arr)

{

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

{

boolean isDistinct = false;

for(int j=0;j<i;j++)

{

if(arr[i] == arr[j])

{

isDistinct = true;

break;

}

}

if(!isDistinct)

{

System.*out*.print(arr[i]+" ");

}

}

}

public static void main(String a[])

{

System.*out*.println("-- Array elements with duplication: --");

int[] nums = {5,2,7,2,3,4,7,8,2,3};

for (int i = 0; i < nums.length; i++)

{

System.*out*.print(nums[i]+" ");

}

System.*out*.println();

System.*out*.println("-- Distinct Elements without duplication: --");

MyDisticntElements.*printDistinctElements*(nums);

}

}

**Output:**

-- Array elements with duplication: --

5 2 7 2 3 4 7 8 2 3

-- Distinct Elements without duplication: --

5 2 7 3 4 8

**Q66: Find 1st 2nd 3rd Largest number from Array without duplication:**

package com.java.paras;

public class FindLargestNumber

{

public static void main(String[] args)

{

int temp1=0,temp2=0,temp3=0;

int arr[]={12,10,14,20,05,20,12};

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

{

if(arr[i] > temp1)

{

temp1=arr[i];

}

}

System.*out*.println("1st Largest no. is: "+temp1);

for (int j = 0; j < arr.length; j++)

{

if(arr[j] > temp2 && arr[j] < temp1)

{

temp2=arr[j];

}

}

System.*out*.println("2nd Largest no. is: "+temp2);

for (int k = 0; k < arr.length; k++)

{

if(arr[k] > temp3 && arr[k] < temp2)

{

temp3=arr[k];

}

}

System.*out*.println("3rd Largest no. is: "+temp3);

}

}

**Output:**

1st Largest no. is: 20

2nd Largest no. is: 14

3rd Largest no. is: 12

**Q67: Find 1st 2nd 3rd Smallest number from Array without duplication:**

package com.java.paras;

public class FindSmallestNumber

{

public static void main(String[] args)

{

int arr[]={100,99,98,97,10,10,95,11,9,9,99,85};

int min1=arr[1]; int index1=1;

int min2=arr[2]; int index2=1;

int min3=arr[3]; int index3=1;

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

{

if(arr[i] < min3 && arr[i] > min2)

{

min3 = arr[i];

index3 = i;

}

if(arr[i] < min2 && arr[i] > min1)

{

min3 = min2;

index3 = index2;

min2 = arr[i];

index2 = i;

}

if(arr[i] < min1)

{

min3 = min2;

index3 = index2;

min2 = min1;

index2 = index1;

min1 = arr[i];

index1 = i;

}

}

Sop("1st smallest at: "+index1+" position & value is: "+min1);

Sop("2nd smallest at: "+index2+" position & value is: "+min2);

Sop("3rd smallest at: "+index3+" position & value is: "+min3);

}

}

**Output**:

1st smallest no. at: 8 position & value are: 9

2nd smallest no. at: 4 position & value are: 10

3rd smallest no. at: 7 position & value are: 11

**Q68: Most Repeated Value in a Array:**

package com.java.paras;

public class MostRepeatedValue

{

public static void main(String[] args)

{

int arr[]={1,2,3,3,3,4,4,8,7,6,10,5,3};

int a=*getPopularElement*(arr);

System.*out*.println("The Most Repeated Value is: ");

System.*out*.println(a);

}

public static int getPopularElement(int[] arr)

{

int count=1,tempCount;

int popular = arr[0];

int temp = 0;

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

{

temp = arr[i];

tempCount = 0;

for (int j = 1; j < arr.length; j++)

{

if(temp == arr[j])

tempCount++;

}

if(tempCount > count)

{

popular = temp;

count = tempCount;

}

}

return popular;

}

}

**Ouput:**

The Most Repeated Value is:

3

**Q69: Find common element between two Array:**

package com.java.paras;

public class CommanElemtTwoArray

{

public static void main(String[] args)

{

int[] arr1 = {4,7,3,9,2};

int[] arr2 = {3,2,12,9,40,32,4};

System.*out*.println("-- Comman Elements are: --");

for(int i=0;i<arr1.length;i++)

{

for(int j=0;j<arr2.length;j++)

{

if(arr1[i]==arr2[j])

{

System.*out*.print(arr1[i]+" ");

}

}

}

}

}

**Output:**

-- Comman Elements are: --

4 3 9 2

**Q70: Palindrome Number Validation:**

package com.java.paras;

import java.util.Scanner;

public class PalindromeTest

{

public static void main(String[] args)

{

System.*out*.println("Enter the Number: ");

int num = new Scanner(System.*in*).nextInt();

int palindrome = num;

int rev = 0;

while(palindrome != 0)

{

int rem = palindrome % 10;

rev = rev \* 10 + rem;

palindrome = palindrome / 10;

}

if(num == rev)

{

S.o.p(num+" --> is a palindrome number...!!");

}

else

{

S.o.p(num+" --> is not a palindrome number...!!");

}

}

}

**Ouput**:

Enter the Number:

1331

1331 --> is a palindrome number...!!

**Q71: Serialization Test Program:**

package com.java.paras;

import java.io.\*;

class Hello implements Serializable

{

int sid;

String sname;

Hello(int sid,String sname)

{

this.sid=sid;

this.sname=sname;

}

}

public class Serialization

{

public static void main(String[] args) throws Exception

{

Student st=new Student(123,"paras puuruu");

FileOutputStream fos=new FileOutputStream("D:\\paras\\serial.ser");

ObjectOutputStream out=new ObjectOutputStream(fos);

out.writeObject(st);

out.flush();

System.*out*.println("Serialized Sucessfully..!!!!");

}

}

**Output:**

Serialized Sucessfully..!!!!

**Q72: De-Serialization Program:**

package com.java.paras;

import java.io.\*;

class Dserialization

{

public static void main(String args[])throws Exception

{

ObjectInputStream in=new ObjectInputStream(new FileInputStream("D:\\paraspuru\\serial.txt"));

Student s=(Student)in.readObject();

System.*out*.println("-- Dserialized data --");

System.*out*.println(s.id+" "+s.name);

in.close();

}

}

Output:

-- Dserialized data --

123 paras puuruu

**Q73: Diamond Shape Problem:**

package com.java.paras;

public class DiamondShape

{

public static void main(String[] args) throws Exception

{

for (int i = 0; i < 7; i++)

{

for (int j = i; j < 7; j++)

{

System.*out*.print(" ");

}

for (int k = 0; k < i; k++)

{

System.*out*.print("\* ");

}

System.*out*.println();

}

for (int i = 7; i > 0; i--)

{

for (int j = i; j < 7; j++)

{

System.*out*.print(" ");

}

for (int k = 0; k < i; k++)

{

System.*out*.print("\* ");

}

System.*out*.println();

}

}

}

**Ouput**:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Q74: Count Number of space, words, vowels, consonant, character etc..!!!**

import java.io.\*;

class Count

{

public void countSpace(String S)

{

int i, ln, c = 0;

ln = S.length();

for (i = 0; i < ln; i++)

{

if (S.charAt(i) == ' ')

c++;

}

System.*out*.println("number of spaces are: " + c);

}

public void countVowel(String S)

{

int i, v = 0,c=0, ln;

char ch;

ln = S.length();

for (i = 0; i < ln; i++)

{

ch = S.charAt(i);

if (ch=='a'|| ch=='e'|| ch=='i' || ch=='o' || ch=='u')

{

v++;

}

else

{

c++;

}

}

System.*out*.println("number of vowels are: " + v);

System.*out*.println("Number of Consonent are: "+c);

}

public void countCharacters(String str)

{

int letter = 0;

for (int i = 0; i < str.length(); i++)

{

char ch=str.charAt(i);

if (Character.*isLetter*(ch))

{

letter++;

}

}

System.*out*.println("---------------------------");

int count = 0;

char ch[]=str.toCharArray();

for (int i = 0; i < ch.length; i++)

{

if(ch[i]!='\0'||ch[i]=='\0')

{

count++;

}

}

System.*out*.println("number of characters are: " +letter);

System.*out*.println("number of characters are: " +count);

}

public void countWords(String str)

{

int count = 1;

for (int i=0;i <= str.length()-1;i++)

{

if (str.charAt(i) == ' ' && str.charAt(i+1)!=' ')

{

count++;

}

}

System.*out*.println("number of words are: " + count);

}

public static void main(String args[]) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.*in*));

Count obj = new Count();

System.*out*.println("-- Enter the string: --");

String S = br.readLine();

System.*out*.println("------------------------");

obj.countSpace(S);

obj.countWords(S);

obj.countVowel(S);

obj.countCharacters(S);

}

}

**Output:**

-- Enter the string: --

paras puru sir mvit

------------------------

number of spaces are: 3

number of words are: 4

number of vowels are: 6

Number of Consonent are: 13

---------------------------

number of characters are: 16

number of characters are: 19

**Q75: Check whether input number is unique or not?**

package com.java.paras;

import java.util.Scanner;

public class UniqeNumber

{

public static void main(String[] args)

{

System.*out*.println("-- Enter the value: --");

Scanner sc=new Scanner(System.*in*);

String s=sc.next();

int size=s.length();

int flag=0;

for(int i=0; i<size-1; i++)

{

for(int j=i+1; j<size; j++)

{

if(s.charAt(i)==s.charAt(j))

{

flag=1;

break;

}

}

}

if(flag==0)

System.*out*.println("\*\* The Number is a Unique Number \*\*");

else

System.*out*.println("\*\* The Number is Not a Unique Number \*\*");

}

}

**Output**:

-- Enter the value: --

1122

\*\* The Number is Not a Unique Number \*\*

-----------------------------------------

-- Enter the value: --

2543

\*\* The Number is a Unique Number \*\*

**Q76: Write a Java program that prints the numbers from 1 to 50. But for multiples of three print "Fizz"instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz":**

package com.java.paras;

public class FizzBuzzTest

{

public static void main(String[] args)

{

for(int i = 1; i <= 30; i++)

{

if(i % (3\*5) == 0) System.*out*.println("FizzBuzz");

else if(i % 5 == 0) System.*out*.println("Buzz");

else if(i % 3 == 0) System.*out*.println("Fizz");

else System.*out*.println(i);

}

}

}

**Output**:

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

**Q77: Display Denomination of an Amount:**

package com.java.paras;

import java.io.\*;

import java.util.Scanner;

class Denominations

{

public static void main(String[] args) throws IOException

{

Scanner sc=new Scanner(System.*in*);

int den[] = { 1000, 500, 100, 50, 20, 10, 5, 2, 1 };

System.*out*.println("-- Enter any Amount: --");

int amount = Integer.*parseInt*(sc.nextLine());

int copy = amount;

int totalNotes = 0, count = 0;

System.*out*.println("\n-- AMOUNT DENOMINATIONS: --\n");

for (int i = 0; i < 9; i++)

{

count = amount / den[i];

if (count != 0)

{

System.*out*.println(den[i] + "\tx\t" + count + "\t= " + den[i] \* count);

}

totalNotes = totalNotes + count;

amount = amount % den[i];

}

System.*out*.println("--------------------------------");

System.*out*.println("TOTAL\t\t\t= " + copy);

System.*out*.println("--------------------------------");

System.*out*.println("Total No. of Notes\t= " + totalNotes);

}

}

**Output:**

-- Enter any Amount: --

12565

-- AMOUNT DENOMINATIONS: --

1000 x 12 = 12000

500 x 1 = 500

50 x 1 = 50

10 x 1 = 10

5 x 1 = 5

--------------------------------

TOTAL = 12565

--------------------------------

Total Number of Notes = 16

**Q78: Sort Two Dimensional Array:**

package com.java.paras;

import java.util.Scanner;

public class SortTwoDimensArray

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("-- Enter the no. of rows: --");

int r=Integer.*parseInt*(sc.nextLine());

System.*out*.println("-- Enter the no. of columns: --");

int c=Integer.*parseInt*(sc.nextLine());

int A[][]=new int[r][c];

/\* Inputting the 2D Array \*/

System.*out*.println("-- Enter the elements: --");

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

A[i][j]=Integer.*parseInt*(sc.nextLine());

}

}

/\* Printing the original 2D Array \*/

System.*out*.println("The original array:");

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

System.*out*.print(A[i][j]+"\t");

}

System.*out*.println();

}

/\* Sorting the 2D Array \*/

int t=0;

for(int x=0;x<r;x++)

{

for(int y=0;y<c;y++)

{

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

if(A[i][j]>A[x][y])

{

t=A[x][y];

A[x][y]=A[i][j];

A[i][j]=t;

}

}

}

}

}

/\* Printing the sorted 2D Array \*/

System.*out*.println("The Sorted Array:");

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

System.*out*.print(A[i][j]+"\t");

}

System.*out*.println();

}

}

}

**Output:**

-- Enter the no. of rows: --

3

-- Enter the no. of columns: --

3

-- Enter the elements: --

12 3 26 20 26 75 10 1 56

The original array:

12 3 26

20 26 75

10 1 56

The Sorted Array:

1 3 10

12 20 26

26 56 75

**Q79: Tower Of Hanoi puzzle problem:**

package com.java.paras;

import java.util.Scanner;

public class TowerOfHanoi

{

public static void main(String[] args)

{

System.*out*.println("Please enter number of discs to move:");

Scanner scanner = new Scanner(System.*in*);

int num\_of\_discs = scanner.nextInt();

*solve*(num\_of\_discs, 'A', 'B', 'C');

}

public static void solve(int first\_disc, char aTower, char bTower, char cTower)

{

if (first\_disc == 1)

{

System.*out*.println("Disk 1 on tower " + aTower + " moving to tower " + cTower);

}

else

{

*solve*(first\_disc - 1, aTower, cTower, bTower);

System.*out*.println("Disk " + first\_disc + " on tower " + aTower + " moving to tower " + cTower);

*solve*(first\_disc - 1, bTower, aTower, cTower);

}

}

}

**Output:**

Please enter number of discs to move:

3

Disk 1 on tower A moving to tower C

Disk 2 on tower A moving to tower B

Disk 1 on tower C moving to tower B

Disk 3 on tower A moving to tower C

Disk 1 on tower B moving to tower A

Disk 2 on tower B moving to tower C

Disk 1 on tower A moving to tower C

**Q80: To get input from User interactively:**

package com.java.paras;

import java.util.Scanner;

public class UserInputExample

{

public static void main(String args[])

{

Scanner console = new Scanner(System.*in*);

System.*out*.println("System is ready to accept ,enter name: ");

String name = console.nextLine();

System.*out*.println("Hi " + name + ", Can you enter an int number now?");

int number = console.nextInt();

System.*out*.println("You have entered : " + number);

System.*out*.println("Thank you");

}

}

**Ouptut:**

System is ready to accept input, please enter name :

paras

Hi paras, Can you enter an int number now?

31

You have entered : 31

Thank you

**Q81: BinarySearch Program:**

package com.java.paras;

import java.util.Scanner;

public class BinarSearch

{

public static void main(String arg[])

{

Scanner sc=new Scanner(System.*in*);

System.*out*.println("Enter number of elements");

int num = sc.nextInt();

int a[] = new int[num];

System.*out*.println("Please enter element in ascending order:");

for (int i = 0; i < num; i++)

{

a[i] = sc.nextInt();

}

System.*out*.println("Enter the element to search");

int find = sc.nextInt();

int pos = *search*(a, find);

if (pos == -1)

{

System.*out*.println("Element not found");

}

else

{

System.*out*.println("Element found at: " + pos);

}

}

public static int search(int ar[], int find)

{

int start = 0;

int end = ar.length - 1;

int mid;

while (start <= end)

{

mid = (start + end) / 2;

if (ar[mid] == find)

{

return mid;

}

else if (ar[mid] < find)

{

start = mid + 1;

}

else if (ar[mid] > find)

{

end = mid - 1;

}

}

return -1;

}

}

**Output:**

Enter number of elements

5

Please enter element in ascending order:

1 2 3 4 5

Enter the element to search

4

Element found : 3

**Q82. Two Matrix multiplication:**

import java.util.Scanner;

public class MatrixTest

{

public static void main(String[] args)

{

int c[][]=new int[3][3];

int a[][]=new int[3][3];

int b[][]=new int[3][3];

System.*out*.println("Enter 1st Matrix");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

a[i][j]=new Scanner(System.*in*).nextInt();

}

}

System.*out*.println("Enter 2nd Matrix");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

b[i][j]=new Scanner(System.*in*).nextInt();

}

}

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

c[i][j]=0;

for (int k = 0; k < 3; k++)

{

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

}

}

}

System.*out*.println("A Matrix is: ");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

System.*out*.print(" "+a[i][j]);

}

System.*out*.println();

}

System.*out*.println("B Matrix is: ");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

System.*out*.print(" "+b[i][j]);

}

System.*out*.println();

}

System.*out*.println("Matrix are: ");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

System.*out*.print(" "+c[i][j]);

}

System.*out*.println(" ");

}

}

}

**Output**:  
Enter 1st Matrix

1 2 3 2 4 5 3 5 2

Enter 2nd Matrix

4 2 4 5 2 1 5 3 9

A Matrix is:

1 2 3

2 4 5

3 5 2

B Matrix is:

4 2 4

5 2 1

5 3 9

Matrix are:

29 15 33

53 27 57

47 22 35

**Q83. Thread Pool in java:**

package com.java.paras;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

class WorkerThread implements Runnable

{

private String message;

public WorkerThread(String s)

{

this.message=s;

}

public void run()

{

System.*out*.println(Thread.*currentThread*().getName()+" (Start) message = "+message);

processmessage();

System.*out*.println(Thread.*currentThread*().getName()+" (End)");

}

private void processmessage()

{

try

{

Thread.*sleep*(2000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

public class ThreadPool

{

public static void main(String[] args)

{

ExecutorService executor = Executors.*newFixedThreadPool*(5);

for (int i = 0; i < 10; i++)

{

Runnable worker = new WorkerThread("" + i);

executor.execute(worker);

}

executor.shutdown();

while (!executor.isTerminated())

{

}

System.*out*.println("Finished all threads");

}

}

**Output:**

pool-1-thread-1 (Start) message = 0

pool-1-thread-3 (Start) message = 2

pool-1-thread-5 (Start) message = 4

pool-1-thread-2 (Start) message = 1

pool-1-thread-4 (Start) message = 3

pool-1-thread-2 (End)

pool-1-thread-2 (Start) message = 5

pool-1-thread-5 (End)

pool-1-thread-5 (Start) message = 6

pool-1-thread-1 (End)

pool-1-thread-1 (Start) message = 7

pool-1-thread-3 (End)

pool-1-thread-3 (Start) message = 8

pool-1-thread-4 (End)

pool-1-thread-4 (Start) message = 9

pool-1-thread-3 (End)

pool-1-thread-2 (End)

pool-1-thread-4 (End)

pool-1-thread-5 (End)

pool-1-thread-1 (End)

Finished all threads

**Q.84 Singleton Design Pattern:**

package com.java.DesignPattern;

class Hello

{

private static Hello *hello*=null;

private Hello()

{

}

static public Hello getInstance()

{

if(*hello*==null)

{

*hello*=new Hello();

}

return *hello*;

}

public void showInstance()

{

System.*out*.println("Hi..ShowInstance()");

}

}

public class SingletonPattern

{

public static void main(String[] args)

{

Hello h1=Hello.*getInstance*();

h1.showInstance();

Hello h2=Hello.*getInstance*();

h2.showInstance();

}

}

**Output:**

Hi..ShowInstance()..!!

Hi..ShowInstance()..!!

**Q.85 MVC Design Pattern:**

package com.java.DesignPattern;

//Create Model.

class Student

{

private String rollNo;

private String name;

public String getRollNo()

{

return rollNo;

}

public void setRollNo(String rollNo)

{

this.rollNo = rollNo;

}

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

}

//Create View.

class StudentView

{

public void printStudentDetails(String studentName, String studentRollNo)

{

System.*out*.println("Student: ");

System.*out*.println("Name: " + studentName);

System.*out*.println("Roll No: " + studentRollNo);

}

}

//Create Controller.

class StudentController

{

private Student model;

private StudentView view;

public StudentController(Student model, StudentView view)

{

this.model = model;

this.view = view;

}

public void setStudentName(String name)

{

model.setName(name);

}

public String getStudentName()

{

return model.getName();

}

public void setStudentRollNo(String rollNo)

{

model.setRollNo(rollNo);

}

public String getStudentRollNo()

{

return model.getRollNo();

}

public void updateView()

{

view.printStudentDetails(model.getName(), model.getRollNo());

}

}

//Use the StudentController methods to demonstrate MVC design pattern usage.

public class MVCDesignPattern

{

public static void main(String[] args)

{

//fetch student record based on his roll no from the database

Student model = *retriveStudentFromDatabase*();

//Create a view : to write student details on console

StudentView view = new StudentView();

StudentController controller = new StudentController(model, view);

controller.updateView();

//update model data

controller.setStudentName("John");

controller.updateView();

}

private static Student retriveStudentFromDatabase()

{

Student student = new Student();

student.setName("Robert");

student.setRollNo("10");

return student;

}

}

**Output**:

Student:

Name: Robert

Roll No: 10

Student:

Name: John

Roll No: 10

**Q.86 Factory Design Pattern:**

package com.java.paras;

import java.util.Scanner;

abstract class Plan

{

protected double rate;

abstract void getRate();

public void calculateBill(int units)

{

System.*out*.println(units\*rate);

}

}

class DomesticPlan extends Plan

{

public void getRate()

{

rate=3.50;

}

}

class CommercialPlan extends Plan

{

public void getRate()

{

rate=7.50;

}

}

class InstitutionalPlan extends Plan

{

public void getRate()

{

rate=5.50;

}

}

class GetPlanFactory

{

public Plan getPlan(String planType)

{

if(planType == null)

{

return null;

}

if(planType.equalsIgnoreCase("DOMESTICPLAN"))

{

return new DomesticPlan();

}

else if(planType.equalsIgnoreCase("COMMERCIALPLAN"))

{

return new CommercialPlan();

}

else if(planType.equalsIgnoreCase("INSTITUTIONALPLAN"))

{

return new InstitutionalPlan();

}

return null;

}

}

public class GenerateBill

{

public static void main(String[] args)

{

GetPlanFactory planFactory = new GetPlanFactory();

System.*out*.println("Enter the name of plan for which the bill will be generated: ");

Scanner sc=new Scanner(System.*in*);

String planName=sc.next();

System.*out*.println("Enter the number of units for bill will be calculated: ");

int units=sc.nextInt();

Plan p = planFactory.getPlan(planName);

System.*out*.println("Bill amount for "+planName+" of "+units+" units is: ");

p.getRate();

p.calculateBill(units);

}

}

**Output:**

Enter the name of plan for which the bill will be generated:

DOMESTICPLAN

Enter the number of units for bill will be calculated:

500

Bill amount for DOMESTICPLAN of 500 units is:

1750.0

**Q87. Façade Design Pattern:**

package com.java.DesignPattern;

interface Shape

{

void draw();

}

class Rectangle implements Shape

{

public void draw()

{

System.*out*.println("Rectangle::draw()");

}

}

class Square implements Shape

{

public void draw()

{

System.*out*.println("Square::draw()");

}

}

class Circle implements Shape

{

public void draw()

{

System.*out*.println("Circle::draw()");

}

}

class ShapeMaker

{

private Shape circle;

private Shape rectangle;

private Shape square;

public ShapeMaker()

{

circle = new Circle();

rectangle = new Rectangle();

square = new Square();

}

public void drawCircle()

{

circle.draw();

}

public void drawRectangle()

{

rectangle.draw();

}

public void drawSquare()

{

square.draw();

}

}

public class FacadePattern

{

public static void main(String[] args)

{

ShapeMaker shape = new ShapeMaker();

shape.drawCircle();

shape.drawRectangle();

shape.drawSquare();

}

}

**Output:**

Circle::draw()

Rectangle::draw()

Square::draw()

**Q88. IPAddress Validataion Program:**

package com.java.paras;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class MyIpAddress

{

public static boolean isValidIP(String ipAddr)

{

Pattern ptn = Pattern.compile("^(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})$");

Matcher mtch = ptn.matcher(ipAddr);

return mtch.find();

}

public static void main(String a[])

{

System.out.println("126.23.45.12 is valid?: "+MyIpAddress.isValidIP("126.231.45.12"));

System.out.println("10.2a.56.32 is valid?: "+MyIpAddress.isValidIP("10.2a.56.32"));

System.out.println("10.23.45 is valid?: "+MyIpAddress.isValidIP("10.23.45"));

}

}

**Output**:

126.23.45.12 is valid?: true

10.2a.56.32 is valid?: false

10.23.45 is valid?: false