* Write java code in notepad or notepad++ and save it under the extension .java
* cmd (path: .......\Java\bin)
* Javac filename.java

(If no errors then)

* Java filename

**############################################################################**

import java.util.\*;

class **LinearSearch**

{

public static void main(String args[])

{

int size,search, a[];

Scanner sc = new Scanner(System.in);

System.out.print("Array size: ");

size = sc.nextInt();

a = new int[size];

System.out.print("Array elements: ");

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

a[i] = sc.nextInt();

System.out.print("Search element: ");

search = sc.nextInt();

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

{

if ( a[i] == search)

{

System.out.print( search+ " found at location " + (i+1));

break;

}

if (i == size)

System.out.print(search+ " is not found.");

}

}

}

**############################################################################**

import java.util.\*;

class **BinarySearch**

{

public static void main(String args[])

{

int size,search, a[], first=0, last, mid;

Scanner sc = new Scanner(System.in);

System.out.println("Array size: ");

size = sc.nextInt();

a = new int[size];

last=size-1;

mid=(first + last)/2;

System.out.println("Array elements: ");

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

a[i] = sc.nextInt();

System.out.println("Search element: ");

search = sc.nextInt();

while( first <= last )

{

if ( a[mid] < search)

first = mid + 1;

else if ( a[mid] == search)

{

System.out.println( search+ " found at location " + (mid+1));

break;

}

else

last = mid-1;

mid = (first + last)/2;

}

if ( first > last )

System.out.println( search+ " is not found.");

}

}

**###########################################################################**

**//Binary Search COMMAND LINE ARGUMENTS**

import java.util.\*;

class **BinarySearch\_cmd**

{

public static void main(String args[])

{

int s,mid, low, high,c,n;

n=args.length-1;

low=0;

high=n-1;

s=Integer.parseInt(args[n]);

while(low<=high)

{

mid=(low + high)/2;

c=Integer.parseInt(args[mid]);

if(c==s)

{

System.out.println("Element has been found at position "+(mid+1));

break;

}

else if(c<s)

low=mid+1;

else

high=mid-1;

}

if (low>high)

System.out.println("Element doesn't exist");

}

}

//OUTPUT

C:\Users\prach\OneDrive\Desktop\Java\jdk-18.0.1.1\bin>javac BS\_cmd.java

C:\Users\prach\OneDrive\Desktop\Java\jdk-18.0.1.1\bin>java BS\_cmd 2 3 4 5 3 4

Element has been found at position 3

**############################################################################**

import java.util.\*;

class **BubbleSort**

{

public static void main(String args[])

{

int size,a[],x;

Scanner sc = new Scanner(System.in);

System.out.println("Array size: ");

size = sc.nextInt();

a = new int[size];

System.out.println("Array elements: ");

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

a[i] = sc.nextInt();

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

{

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

{

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

{

x = a[j];

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

a[j+1] = x;

}

}

}

System.out.println("Sorted array:");

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

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

}

}

**############################################################################**

import java.util.\*;

class **Fibonacci**

{

public static void main(String args[])

{

Scanner sc= new Scanner(System.in);

int n,n1=0,n2=1,n3,i;

System.out.print("Enter limit: ");

n=sc.nextInt();

System.out.print(n1+" "+n2);

n3=n1+n2;

for(i=2;n3<=n;i++)

{

System.out.print(" "+n3);

n1=n2;

n2=n3;

n3=n1+n2;

}

System.out.println(" Total terms= " +i);

}}

**############################################################################**

import java.util.\*;

class **Matrices**

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int m1,n1,m2,n2,i,j,k;

char c;

System.out.println("FIRST MATRIX:");

System.out.println("Order: ");

m1=sc.nextInt();

n1=sc.nextInt();

System.out.println("Elements: ");

int a[][]=new int[m1][n1];

for(i=0; i<m1; i++)

{

for(j=0;j<n1;j++)

{

if(j==0)

System.out.println("Enter element into row : "+(i+1));

a[i][j]=sc.nextInt();

}

System.out.println();

}

System.out.println("SECOND MATRIX:");

System.out.println("Order: ");

m2=sc.nextInt();

n2=sc.nextInt();

System.out.println("Elements: ");

int b[][]=new int[m2][n2];

for(i=0; i<m2; i++)

{

for(j=0;j<n2;j++)

{

if(j==0)

System.out.println("Enter element into row : "+(i+1));

b[i][j]=sc.nextInt();

}

System.out.println();

}

System.out.println("Enter the respective sign for the operation to be performed");

System.out.println("+ for addition, - for subtraction, \* for multiplication");

c=sc.next().charAt(0);

if(c=='+')

{

if(m1==m2 && n1==n2)

{

System.out.println("Result after ADDITION:");

for(i=0; i<m2; i++)

{

for(j=0;j<n2;j++)

{

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

}

System.out.println();

}

}

else

System.out.println("The order of the matrices are not equal");

}

else if(c=='-')

{

if(m1==m2 && n1==n2)

{

System.out.println("Result after SUBTRACTION:");

for(i=0; i<m2; i++)

{

for(j=0;j<n2;j++)

{

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

}

System.out.println();

}

}

else

System.out.println("The order of the matrices are not equal");

}

else if(c=='\*')

{

if(n1==m2)

{

int s[][]=new int[m1][n2];

for(i=0;i<m1;i++)

{

for(j=0;j<n2;j++)

{

s[i][j]=0;

for(k=0;k<n1;k++)

{

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

}

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

}

System.out.println();

}

}

else

System.out.println("The order of the matrices are not suitable for multiplication");

}

else

System.out.println("Invalid operation");

}

}

**############################################################################**

import java.util.\*;

class **Palindrome**

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n,t,r=0,s=0,l;

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

n=sc.nextInt();

t=n;

while(n!=0)

{

l=n%10;

s+=l;

r=(r\*10)+l;

n=n/10;

}

if(r==t)

System.out.println("It is a palindrome number");

else

System.out.println("It is NOT a palindrome number");

System.out.println("Sum of digits = "+s);

System.out.println("Reverse of the number = "+r);

}

}

**############################################################################**

import java.util.\*;

class **Armstrong**

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n,t,l,a=0;

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

n=sc.nextInt();

t=n;

while(n!=0)

{

l=n%10;

a+=(l\*l\*l);

n=n/10;

}

if(a==t)

System.out.println("It is an armstrong number");

else

System.out.println("It is NOT an armstrong number");

}

}

**############################################################################**

//Converting binary to decimal

import java.util.\*;

class **BinaryDecimal**

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n,l,d=0,i,t,k=0;

System.out.println("Enter a binary number");

n=sc.nextInt();

while(n!=0)

{

l=n%10;

t=1;

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

t\*=2;

d+=(t\*l);

k++;

n=n/10;

}

System.out.println("The corresponding decimal number = "+d);

}

}

**############################################################################**

//Simple Interest-Packages

// in simpleinterest folder

package simpleinterest;

public class si

{

public static void calculate(float P,float R,float T)

{

float i;

i=P\*R\*T/100;

System.out.println("Simple Interest= "+i);

}

}

//in bin

// class which will access the simpleinterest package

import simpleinterest.\*;

public class si2

{

public static void main(String args[])

{

si obj=new si();

obj.calculate(1,1,1);

}

}

**############################################################################**

**//single inheritance using constructor and super keyword**

class Room

{

int length, breadth;

Room(int x, int y)

{

length = x;

breadth = y;

}

int area()

{

return(length\*breadth);

}

}

class Hall extends Room

{

int height;

Hall(int x,int y,int z)

{

super(x,y);

height = z;

}

int Volume()

{

return(length\*breadth\*height);

}

}

class Area

{

public static void main(String args[])

{

Hall h1 = new Hall(10,10,10);

{

System.out.println("Area: "+h1.area());

System.out.println("Volume: "+h1.Volume());

}

}

}

**############################################################################**

**//Single Inheritance, without constructor**

import java.util.\*;

class Employee

{

String name;

int id;

double basic;

void getdata()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Employee name,Employee ID and basic pay. ");

name = sc.nextLine();

id = sc.nextInt();

basic = sc.nextDouble();

}

void display1()

{

System.out.println("Name = "+name);

System.out.println("ID = "+id);

System.out.println("Basic Pay = "+basic);

}

}

class Salary extends Employee

{

double DA, HRA, PF = 1200, GP, NP;

void calculate()

{

DA = basic \* 0.10;

HRA = basic \* 0.05;

GP = basic + DA + HRA;

NP = GP - PF;

}

void display2()

{

System.out.println("Dearness Allowance = "+DA);

System.out.println("House Rent Allowance = "+HRA);

System.out.println("Gross Pay = "+GP);

System.out.println("Net Pay = "+NP);

}

}

class Emp\_Salary

{

public static void main(String args[])

{

Salary s = new Salary();

s.getdata();

s.display1();

s.calculate();

s.display2();

}

}

**############################################################################**

**//Multilevel Inheritance-University**

import java.util.\*;

class University

{

Scanner sc=new Scanner (System.in);

String uname,ulocation,uacc;

void read1()

{

System.out.println("Enter university name ");

uname=sc.nextLine();

System.out.println("Enter university location ");

ulocation=sc.nextLine();

System.out.println("Enter university date of accreditation ");

uacc=sc.nextLine();

}

void display1()

{

System.out.println("University name is "+uname);

System.out.println("University location is "+ulocation);

System.out.println("University date of accreditation is "+uacc);

}

}

class College extends University

{

String cname,clocation,cacc;

void read2()

{

System.out.println("Enter college name ");

cname=sc.nextLine();

System.out.println("Enter college location ");

clocation=sc.nextLine();

System.out.println("Enter college date of accreditation ");

cacc=sc.nextLine();

}

void display2()

{

System.out.println("College name is "+cname);

System.out.println("College location is "+clocation);

System.out.println("College date of accreditation is "+cacc);

}

}

class Course extends College

{

String course\_name, duration;

double fees;

void read3()

{

System.out.println("Enter course name ");

course\_name=sc.nextLine();

System.out.println("Enter course duration ");

duration=sc.next();

System.out.println("Enter course fee ");

fees=sc.nextDouble();

}

void display3()

{

System.out.println("Course name is "+course\_name);

System.out.println("Course fee is "+fees);

System.out.println("Course duration is "+duration);

}

}

class MultilevelUni

{

public static void main(String args[])

{

Course c = new Course();

c.read1();

c.read2();

c.read3();

c.display1();

c.display2();

c.display3();

}

}

**############################################################################**

**//Hierarchical Inheritance-example**

// parent class

class P

{

int parentVariable = 5;

}

// child class

class C1 extends P

{

int childVariable = 1;

}

// child class

class C2 extends P

{

int childVariable = 2;

}

// child class

class C3 extends P

{

int childVariable = 3;

}

public class hierarchical

{

public static void main(String[] args)

{

// object created

C1 child1 = new C1();

C2 child2 = new C2();

C3 child3 = new C3();

System.out.println("Parent variable + Child variable of child1 = " + (child1.parentVariable + child1.childVariable));

System.out.println("Parent variable + Child variable of child2 = " + (child2.parentVariable + child2.childVariable));

System.out.println("Parent variable + Child variable of child3 = " + (child3.parentVariable + child3.childVariable));

}

}

**############################################################################**

**//Bank-constructor**

import java.util.\*;

class BankC

{

Scanner sc = new Scanner(System.in);

String name,acc\_type;

int acc\_no;

double balance,deposit,withdraw;

BankC (String n,String type, int no,double b) //constructor

{

name=n;

acc\_type=type;

acc\_no=no;

balance=b;

}

void DepAmt()

{

System.out.println("Enter amount to deposit: ");

deposit=sc.nextDouble();

balance=balance+deposit;

}

void wdraw()

{

System.out.println("Enter withdrawal amount: ");

withdraw=sc.nextDouble();

if (balance<withdraw)

System.out.println("Insuffient balance");

else

balance=balance-withdraw;

}

void display()

{

System.out.println("Account name: " +name);

System.out.println("Balance: "+balance);

}

public static void main(String args[])

{

BankC ob = new BankC("Khushi","Savings",42,1000);

ob.DepAmt();

ob.wdraw();

ob.display();

}

}

**############################################################################**

**###########################################################################**

import java.util.\*;

class Sum

{

public static void main(String[] args)

{

int a,b;

Scanner sc = new Scanner(System.in);

System.out.print("Enter First Number: ");

a = sc.nextInt();

System.out.print("Enter Second Number: ");

b = sc.nextInt();

System.out.println("Sum: "+(a+b));

}

}

**###########################################################################**

//Program to implement different string operations

class StringOperations

{

public static void main(String[] args)

{

String str1 = "Java";

String str2 = "Programming Lab";

//Length operation

System.out.println("Length of string: " + str1.length());

//Concatenation operation

System.out.println(str1 + str2);

System.out.println(str1.concat(str2));

//Convert to LowerCase operation

System.out.println("Convert to LowerCase: " + str1.toLowerCase());

//Convert to UpperCase operation

System.out.println("Convert to UpperCase: " + str2.toUpperCase());

//Contain operation

System.out.println("Contains sequence 'gram': " + str2.contains("gram"));

//charAt operation

System.out.println("Character at position 5: " + str2.charAt(5));

//Indexof operation

System.out.println("Index of character 'n': " + str2.indexOf('n'));

//EndsWith operation

System.out.println("EndsWith character 'a': " + str1.endsWith("a"));

//Compare to operation

System.out.println("Compare To 'JAVA': " + str1.compareTo("JAVA"));

//Compare to - Ignore case

System.out.println("Compare To 'JAVA' - Case Ignored: " + str1.compareToIgnoreCase("JAVA"));

//Replace operation

System.out.println("Replace 'Lab' with 'Class': " + str2.replace("Lab", "Class"));

}

}

**###########################################################################**

class MergeSort

{

// Merges two subarrays of arr[].

// First subarray is arr[l..m]

// Second subarray is arr[m+1..r]

void merge(int arr[], int l, int m, int r)

{

// Find sizes of two subarrays to be merged

int n1 = m - l + 1;

int n2 = r - m;

/\* Create temp arrays \*/

int L[] = new int[n1];

int R[] = new int[n2];

/\*Copy data to temp arrays\*/

for (int i = 0; i < n1; ++i)

L[i] = arr[l + i];

for (int j = 0; j < n2; ++j)

R[j] = arr[m + 1 + j];

/\* Merge the temp arrays \*/

// Initial indexes of first and second subarrays

int i = 0, j = 0;

// Initial index of merged subarray array

int k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

/\* Copy remaining elements of L[] if any \*/

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

/\* Copy remaining elements of R[] if any \*/

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

// Main function that sorts arr[l..r] using merge()

void sort(int arr[], int l, int r)

{

if (l < r) {

// Find the middle point

int m =l+ (r-l)/2;

// Sort first and second halves

sort(arr, l, m);

sort(arr, m + 1, r);

// Merge the sorted halves

merge(arr, l, m, r);

}

}

/\* A utility function to print array of size n \*/

static void printArray(int arr[])

{

int n = arr.length;

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

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

System.out.println();

}

// Driver code

public static void main(String args[])

{

int arr[] = { 12, 11, 13, 5, 6, 7 };

System.out.println("Given Array");

printArray(arr);

MergeSort ob = new MergeSort();

ob.sort(arr, 0, arr.length - 1);

System.out.println("\nSorted array");

printArray(arr);

}

}

**###########################################################################**

import java.util.\*;

class fibonacci\_prime

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int a=0,b=1,c=0,n,i=1,k=0,p,j;

System.out.println("Enter the number of elements in the series");

n=sc.nextInt();

int h[]=new int[n];

System.out.println("Fibonacci Series :");

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

{

if(i==1)

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

else if(i==2)

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

else

{

c=a+b;

if(c>1)

{

p=0;

for(j=2;j<c;j++)

{

if(c%j==0)

p=1;

}

if(p==0)

{

h[k]=c;

k++;

}

}

a=b;

b=c;

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

}

}

System.out.println();

System.out.println("Prime numbers from the above series are as follows");

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

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

}

}

**############################################################################**