**Code Archival Project**

Siddharth Singh

Program - 1

1. Question -

A class Collection contains an array of 100 integers. Using the following class description, create an array with common elements from two integer arrays. Some of the members of the class are given below:

Class name : Collection

Data members :

ar[] : integer array of 100 elements

len : length of the array

Member methods :

Collection() : default constructor

Collection(int ) : parameterized constructor to assign the length of

the array

void input() : reads array elements

Collection common(Collection): return the Collection containing the common

Elements of current Collection and the collection

object passed as parameter.

void display() : displays the array Collection elements

Write the main() method to generate the necessary output.

1. Program -

import java.util.\*;

public class Collection

{

int ar[];

int len;

Collection() //default constructor

{

}

Collection(int l) //parameterized constructor

{

len = l;

ar = new int[len];

}

void input() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements");

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

{

ar[i] = sc.nextInt();

}

}

Collection common(Collection a, Collection b) //finds common elements

{

a.input();

b.input();

Collection c = new Collection(100);

int x = 0;

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

{

for(int j = 0; j<b.len; j++)

{

if(a.ar[i]==b.ar[j])

{

c.ar[x]=a.ar[i];

x++;

}

}

}

Collection c1= new Collection(x);

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

{

c1.ar[i] = c.ar[i];

}

return c1;

}

void display(Collection aa, Collection bb) //output

{

Collection d = new Collection();

d = d.common(aa,bb);

System.out.println("The common elements are:");

for(int i = 0; i<d.len; i++)

{

System.out.println(d.ar[i]);

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the first array");

int size1 = sc.nextInt();

Collection obj1 = new Collection(size1);

System.out.println("Enter the size of the second array");

int size2 = sc.nextInt();

Collection obj2 = new Collection(size2);

obj1.display(obj1, obj2);

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int ar[], int len

Step 3 - Collection()

{

}

Step 4 - Collection(int l)

{

len = l;

ar = new int[len];

}

Step 5 - void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements");

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

{

ar[i] = sc.nextInt();

}

}

Step 6 - Collection common(Collection a, Collection b)

{

a.input();

b.input();

Collection c = new Collection(100);

int x = 0;

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

{

for(int j = 0; j<b.len; j++)

{

if(a.ar[i]==b.ar[j])

{

c.ar[x]=a.ar[i];

x++;

}

}

}

Collection c1= new Collection(x);

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

{

c1.ar[i] = c.ar[i];

}

return c1;

}

Step 7 - void display(Collection aa, Collection bb)

{

Collection d = new Collection();

d = d.common(aa,bb);

System.out.println("The common elements are:");

for(int i = 0; i<d.len; i++)

{

System.out.println(d.ar[i]);

}

}

Step 8 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the first array");

int size1 = sc.nextInt();

Collection obj1 = new Collection(size1);

System.out.println("Enter the size of the second array");

int size2 = sc.nextInt();

Collection obj2 = new Collection(size2);

obj1.display(obj1, obj2);

}

Step 9 - Stop

1. Output -

Enter the size of the first array

10

Enter the size of the second array

10

Enter the elements

1

2

3

4

5

6

7

8

9

10

Enter the elements

5

6

7

8

9

10

11

12

13

14

The common elements are:

5

6

7

8

9

10

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| ar[] | int | Stores the array of numbers. |
| len | int | Store the length of the array of numbers. |
| l | int | Inputs, stores and assigns the length of the array to len. |
| i | int | Counter variable. |
| j | int | Counter variable. |
| x | int | Stores index number for assigning values of ar[] of object c, and in the end it also stores the length of ar[] in object c1. |
| size1 | int | Stores the size of the first array input by the user. |
| size2 | int | Stores the size of the second array input by the user. |

Program - 2

1. Question -

A Transpose of an array is obtained by interchanging the elements of the rows and columns. A class Transarray contains a two dimensional integer array of order [m x n]. The maximum value possible for both `m’ and `n’ is 20. Design a class Transarray to find the transpose of a given matrix. The details of the members of the class are given below

|  |  |  |
| --- | --- | --- |
| Class name | : | Transarray |
| Data members/instance variables | : |  |
| arr[] | : | stores the matrix elements |
| M | : | integer to store the number of rows |
| N | : | integer to store the number of columns |
| Member functions | : |  |
| Transarray() | : | default constructor |
| Transarray(int mm, int nn) | : | to initialize the size of the matrix, m=mm, n=nn |
| void fillarray() | : | to enter the elements of the matrix |
| void transpose(Transarray A) | : | to find the transpose of a given matrix |
| void disparrary() | : | displays the array in a matrix form |

Specify the class Transarray giving the details of the constructors, void fillarray(), void transpose(Transarray) and void disparray(). You need not write the main function.

1. Program -

import java.util.\*;

public class Transarray

{

int arr[][];

int M;

int N;

Transarray() //default constructor

{

}

Transarray(int mm, int nn) //parameterized constructor

{

M = mm;

N = nn;

}

void fillarray() //input

{

Scanner sc = new Scanner(System.in);

arr = new int[M][N];

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

void transpose(Transarray A) //transposes the array

{

int arr2[][] = new int[M][N];

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

{

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

{

arr2[j][i]=arr[i][j];

}

}

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

{

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

{

arr[i][j]=arr2[i][j];

}

}

}

void disparray() //output

{

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

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

{

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

{

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

}

System.out.println();

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns");

int m1 = sc.nextInt();

int n1 = sc.nextInt();

Transarray obj = new Transarray(m1, n1);

obj.fillarray();

obj.transpose(obj);

obj.disparray();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int arr[][], int M, int N

Step 3 - Transarray()

{

}

Step 4 - Transarray(int mm, int nn)

{

M = mm;

N = nn;

}

Step 5 - void fillarray()

{

Scanner sc = new Scanner(System.in);

arr = new int[M][N];

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Step 6 - void transpose(Transarray A)

{

int arr2[][] = new int[M][N];

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

{

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

{

arr2[j][i]=arr[i][j];

}

}

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

{

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

{

arr[i][j]=arr2[i][j];

}

}

}

Step 7 - void disparray()

{

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

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

{

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

{

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

}

System.out.println();

}

}

Step 8 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns");

int m1 = sc.nextInt();

int n1 = sc.nextInt();

Transarray obj = new Transarray(m1, n1);

obj.fillarray();

obj.transpose(obj);

obj.disparray();

}

Step 9 - Stop

1. Output -

Enter the number of rows and columns

3

3

Enter the elements of the matrix

1

2

3

4

5

6

7

8

9

Transposed Array:

1 4 7

2 5 8

3 6 9

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| arr[][] | int | Stores the matrix. |
| M | int | Stores the number of rows. |
| N | int | Sores the number of columns. |
| mm | int | Inputs the number of rows and assigns it to M. |
| nn | int | Inputs the number of columns and assigns it to N. |
| i | int | Counter variable. |
| j | int | Counter variable. |
| arr2[][] | int | Stores the transposed matrix. |
| m1 | int | Stores the number of rows input by the user. |
| n1 | int | Stores the number of columns input by the user. |

Program - 3

1. Question -

A class Array is declared as follows:

Class name : Array

Data members :

ar[] : integer array of size 100

n : size of the array

num : integer variable

Member methods :

Array(int) : a constructor to store 0 at each location of ar[] and to n.

void getArray() : to input n integers in array

int process(Array B, int k): to assign k to num and return the frequency of num from

the array ar[] from the array object B.

Array merge(Array a) : merge the current array with the parameterised array.

void display() : to print the array elements in such a way that only 4 elements

of array should be printed in one line with two spaces in

between the numbers.

Write main() method and call the above methods to generate the output.

1. Program -

import java.util.\*;

public class Array

{

int ar[];

int n;

int num;

Array(int n1) //parameterized constructor

{

n = n1;

ar = new int[n];

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

{

ar[i] = 0;

}

}

void getArray() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the array");

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

{

ar[i] = sc.nextInt();

}

}

int process(Array B, int k) //finds frequency

{

num = k;

int freq = 0;

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

{

if(B.ar[i]==num)

{

freq++;

}

}

return freq;

}

Array merge(Array a) //merges the arrays

{

Array c = new Array((this.n)+(a.n));

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

{

c.ar[i] = this.ar[i];

}

int x = 0;

for(int j = this.n; j<(this.n)+(a.n); j++)

{

c.ar[j] = a.ar[x];

x++;

}

return c;

}

void display() //output

{

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

{

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

if((i+1)%4==0)

{

System.out.println();

}

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

Array obj = new Array(size);

obj.getArray();

System.out.println("Enter the number you want to find the frequency of");

int number = sc.nextInt();

int frequency = obj.process(obj, number);

System.out.println("Frequency of " + number + " is: " + frequency);

System.out.println("Enter the size of the second array");

int size2 = sc.nextInt();

Array obj2 = new Array(size2);

obj2.getArray();

Array obj3 = new Array(size+size2);

obj3 = obj.merge(obj2);

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

obj3.display();

}

}

1. Algorithm -

Step 1 - Start

Step 2- Declare int ar[], int n, int num

Step 3 - Array(int n1)

{

n = n1;

ar = new int[n];

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

{

ar[i] = 0;

}

}

Step 4 - void getArray()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the array");

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

{

ar[i] = sc.nextInt();

}

}

Step 5 - int process(Array B, int k)

{

num = k;

int freq = 0;

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

{

if(B.ar[i]==num)

{

freq++;

}

}

return freq;

}

Step 6 - Array merge(Array a)

{

Array c = new Array((this.n)+(a.n));

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

{

c.ar[i] = this.ar[i];

}

int x = 0;

for(int j = this.n; j<(this.n)+(a.n); j++)

{

c.ar[j] = a.ar[x];

x++;

}

return c;

}

Step 7 - void display()

{

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

{

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

if((i+1)%4==0)

{

System.out.println();

}

}

}

Step 8 - public static void main()

{

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

Array obj = new Array(size);

obj.getArray();

System.out.println("Enter the number you want to find the frequency of");

int number = sc.nextInt();

int frequency = obj.process(obj, number);

System.out.println("Frequency of " + number + " is: " + frequency);

System.out.println("Enter the size of the second array");

int size2 = sc.nextInt();

Array obj2 = new Array(size2);

obj2.getArray();

Array obj3 = new Array(size+size2);

obj3 = obj.merge(obj2);

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

obj3.display();

}

Step 9 - Stop

1. Output -

Enter the size of the array

8

Enter the elements of the array

1

2

3

2

4

2

5

2

Enter the number you want to find the frequency of

2

Frequency of 2 is: 4

Enter the size of the second array

4

Enter the elements of the array

6

7

8

9

Merged array:

1 2 3 2

4 2 5 2

6 7 8 9

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| ar[] | int | Stores the array of elements. |
| n | int | Stores the length of the array. |
| num | int | Stores the number whose frequency is to be found. |
| n1 | int | Stores the length of the array and assigns it to n. |
| i | int | Counter variable. |
| k | int | Stores the number whose frequency is to be found and assigns it to num. |
| freq | int | Stores the frequency of the number stored in num. |
| x | int | Stores the index number of ar[] of object a for merging the 2 arrays. |
| size | int | Stores the size of the first array input by the user. |
| number | int | Stores the number input by the user whose frequency is to be found. |
| frequency | int | Stores the frequency of the number input by the user. |
| size2 | int | Stores the size of the second array input by the user. |

Program - 4

1. Question -

A class Time is declared as follows:

Class name : Time

Data members :

hrs, min : integers to store hour and minute

Methods :

Time() : default constructor to assign 0 to data members

void getTime(int h, int m): assigns hours and minutes

void printTime() : prints time in hh:mm format

Time sumTime(Time t1, Time t2): adds two time t1 with t2 and return them.

Implement the above class and write main() method and call the above methods to generate the output.

1. Program -

import java.util.\*;

public class Time

{

int hrs;

int min;

Time() //default constructor

{

hrs = 0;

min = 0;

}

void getTime(int h, int m) //assigns the hours and minutes

{

hrs = h + (m/60);

min = m%60;

}

void PrintTime() //output

{

System.out.println("Time: " + hrs + ":" + min);

}

Time sumTime(Time t1, Time t2) //finds sum of 2 times

{

Time t3 = new Time();

t3.hrs = t1.hrs + t2.hrs + ((t1.min + t2.min)/60);

t3.min = ((t1.min + t2.min)%60);

return t3;

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

Time obj1 = new Time();

Time obj2 = new Time();

System.out.println("Enter the hours and minutes of first time");

int hrs1 = sc.nextInt();

int min1 = sc.nextInt();

obj1.getTime(hrs1, min1);

obj1.PrintTime();

System.out.println("Enter the hours and minutes of second time");

int hrs2 = sc.nextInt();

int min2 = sc.nextInt();

obj2.getTime(hrs2, min2);

obj2.PrintTime();

Time obj3 = new Time();

obj3 = obj3.sumTime(obj1, obj2);

System.out.print("Total ");

obj3.PrintTime();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int hrs, int min

Step 3 - Time()

{

hrs = 0;

min = 0;

}

Step 4 - void getTime(int h, int m)

{

hrs = h + (m/60);

min = m%60;

}

Step 5 - void PrintTime()

{

System.out.println("Time: " + hrs + ":" + min);

}

Step 6 - Time sumTime(Time t1, Time t2)

{

Time t3 = new Time();

t3.hrs = t1.hrs + t2.hrs + ((t1.min + t2.min)/60);

t3.min = ((t1.min + t2.min)%60);

return t3;

}

Step 7 - public static void main()

{

Scanner sc = new Scanner(System.in);

Time obj1 = new Time();

Time obj2 = new Time();

System.out.println("Enter the hours and minutes of first time");

int hrs1 = sc.nextInt();

int min1 = sc.nextInt();

obj1.getTime(hrs1, min1);

obj1.PrintTime();

System.out.println("Enter the hours and minutes of second time");

int hrs2 = sc.nextInt();

int min2 = sc.nextInt();

obj2.getTime(hrs2, min2);

obj2.PrintTime();

Time obj3 = new Time();

obj3 = obj3.sumTime(obj1, obj2);

System.out.print("Total ");

obj3.PrintTime();

}

Step 8 - Stop

1. Output -

Enter the hours and minutes of first time

1

70

Time: 2:10

Enter the hours and minutes of second time

2

80

Time: 3:20

Total Time: 5:30

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| hrs | int | Stores the number of hours. |
| min | int | Stores the number of minutes. |
| h | int | Stores the number of hours and assigns it to hrs. |
| m | int | Stores the number of minutes and assigns it to min. |
| hrs1 | int | Stores the number of hours of the first time input by the user. |
| min1 | int | Stores the number of minutes of the first time input by the user. |
| hrs2 | int | Stores the number of hours of the second time input by the user. |
| min2 | int | Stores the number of minutes of the second time input by the user. |

Program - 5

1. Question -

Class Matrix contains a two dimensional integer array of order [ m x n ]. The maximum value possible for both m and n is 25. Design a class Matrix to find the difference of the two matrices. The details of the members of the class are given below:

Class name : Matrix

Data members

arr[][] : stores the matrix element

m : integer to store the number of rows

n : integer to store the number of columns

Member functions:

Matrix(int mm, int nn) : to initialize the size of the matrix m=mm and n=nn

void fillarray() : to enter the elements of the matrix

Matrix SubMat(Matrix A) : subtract the current object from the matrix of the parameterized object and return the resulting object.

void display() : display the matrix elements.

Specify the class Matrix giving details of the constructor(int,int), void fillarray(),Matrix

SubMat(Matrix) and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

1. Program -

import java.util.\*;

public class Matrix

{

int arr[][];

int m;

int n;

Matrix(int mm, int nn) //parameterized constructor

{

m = mm;

n = nn;

arr = new int[m][n];

}

void fillarray() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Matrix SubMat(Matrix A) //subtracts 2 matrices

{

Matrix S = new Matrix(A.m,A.n);

for(int i = 0; i<this.m; i++)

{

for(int j = 0; j<this.n; j++)

{

S.arr[i][j] = A.arr[i][j] - this.arr[i][j];

}

}

return S;

}

void display() //output

{

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

{

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

{

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

}

System.out.println();

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns of the first matrix");

int m1 = sc.nextInt();

int n1 = sc.nextInt();

Matrix obj1 = new Matrix(m1, n1);

obj1.fillarray();

System.out.println("First Matrix:");

obj1.display();

System.out.println("Enter the number of rows and columns of the second matrix");

int m2 = sc.nextInt();

int n2 = sc.nextInt();

Matrix obj2 = new Matrix(m2, n2);

obj2.fillarray();

System.out.println("Second Matrix:");

obj2.display();

Matrix obj3 = obj2.SubMat(obj1);

System.out.println("Subtracted Matrix:");

obj3.display();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int arr[][], int m, int n

Step 3 - Matrix(int mm, int nn)

{

m = mm;

n = nn;

arr = new int[m][n];

}

Step 4 - void fillarray()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Step 5 - Matrix SubMat(Matrix A)

{

Matrix S = new Matrix(A.m,A.n);

for(int i = 0; i<this.m; i++)

{

for(int j = 0; j<this.n; j++)

{

S.arr[i][j] = A.arr[i][j] - this.arr[i][j];

}

}

return S;

}

Step 6 - void display()

{

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

{

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

{

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

}

System.out.println();

}

}

Step 7 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns of the first matrix");

int m1 = sc.nextInt();

int n1 = sc.nextInt();

Matrix obj1 = new Matrix(m1, n1);

obj1.fillarray();

System.out.println("First Matrix:");

obj1.display();

System.out.println("Enter the number of rows and columns of the second matrix");

int m2 = sc.nextInt();

int n2 = sc.nextInt();

Matrix obj2 = new Matrix(m2, n2);

obj2.fillarray();

System.out.println("Second Matrix:");

obj2.display();

Matrix obj3 = obj2.SubMat(obj1);

System.out.println("Subtracted Matrix:");

obj3.display();

}

Step 8 - Stop

1. Output -

Enter the number of rows and columns of the first matrix

3

3

Enter the elements of the matrix

1

2

3

4

5

6

7

8

9

First Matrix:

1 2 3

4 5 6

7 8 9

Enter the number of rows and columns of the second matrix

3

3

Enter the elements of the matrix

0

1

2

3

4

5

6

7

8

Second Matrix:

0 1 2

3 4 5

6 7 8

Subtracted Matrix:

1 1 1

1 1 1

1 1 1

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| arr[][] | int | Stores the matrix. |
| m | int | Stores the number of rows of the matrix. |
| n | int | Stores the number of columns of the matrix. |
| mm | int | Stores the number of rows of the matrix and assigns it to m. |
| nn | int | Stores the number of columns of the matrix and assigns it to n. |
| i | int | Counter variable |
| j | int | Counter variable. |
| m1 | int | Stores the number of rows of the first matrix input by the user. |
| n1 | int | Stores the number of columns of the first matrix input by the user. |
| m2 | int | Stores the number of rows of the second matrix input by the user. |
| n2 | int | Stores the number of columns of the second matrix input by the user. |

Program - 6

1. Question -

Write a program to declare a square matrix A[ ] [ ] of order (M x M) where ‘M’ is the number of rows and the number of columns such that M must be greater than 2 and less than 10. Accept the value of M as user input. Display an appropriate message for an invalid input. Allow the user to input integers into this matrix. Perform the following tasks:

1. Display the original matrix.  
   (b) Check if the given matrix is Symmetric or not.  
   A square matrix is said to be Symmetric, if the element of the ith row and jth column is equal to the element of the jth row and ith column.  
   (c) Find the sum of the elements of left diagonal and the sum of the elements of right diagonal of the matrix and display them.
2. Program -

import java.util.\*;

public class prog06

{

int arr[][];

int m;

prog06(int m1) //parameterized constructor

{

m = m1;

arr = new int[m][m];

}

void input() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

void originalmatrix() //prints original matrix

{

System.out.println("Original Matrix");

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

{

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

{

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

}

System.out.println();

}

}

boolean symmetrical() //finds if matrix is symmetrical or not

{

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

{

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

{

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

{

return false;

}

}

}

return true;

}

void diagonals() //finds the sum of the 2 diagonals

{

int left = 0;

int right = 0;

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

{

left += arr[i][i];

}

for(int i = 0, j = m-1; i<m; i++,j--)

{

right += arr[i][j];

}

System.out.println("The sum of the left diagonal = " + left);

System.out.println("The sum of the right diagonal = " + right);

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the row and column length");

int l = sc.nextInt();

if(l<=2 || l>=10)

{

System.out.println("The matrix size is out of range");

System.exit(0);

}

prog06 obj = new prog06(l);

obj.input();

obj.originalmatrix();

if(obj.symmetrical())

{

System.out.println("The given matrix is symmetric");

}

else

{

System.out.println("The given matrix is not symmetric");

}

obj.diagonals();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int arr[][], int m

Step 3 - prog06(int m1)

{

m = m1;

arr = new int[m][m];

}

Step 4 - void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Step 5 - void originalmatrix()

{

System.out.println("Original Matrix");

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

{

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

{

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

}

System.out.println();

}

}

Step 6 - boolean symmetrical()

{

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

{

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

{

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

{

return false;

}

}

}

return true;

}

Step 7 - void diagonals()

{

int left = 0;

int right = 0;

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

{

left += arr[i][i];

}

for(int i = 0, j = m-1; i<m; i++,j--)

{

right += arr[i][j];

}

System.out.println("The sum of the left diagonal = " + left);

System.out.println("The sum of the right diagonal = " + right);

}

Step 8 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the row and column length");

int l = sc.nextInt();

if(l<=2 || l>=10)

{

System.out.println("The matrix size is out of range");

System.exit(0);

}

prog06 obj = new prog06(l);

obj.input();

obj.originalmatrix();

if(obj.symmetrical())

{

System.out.println("The given matrix is symmetric");

}

else

{

System.out.println("The given matrix is not symmetric");

}

obj.diagonals();

}

Step 9 - Stop

1. Output -

Enter the row and column length

3

Enter the elements of the matrix

1

2

3

2

4

5

3

5

6

Original Matrix

1 2 3

2 4 5

3 5 6

The given matrix is symmetric

The sum of the left diagonal = 11

The sum of the right diagonal = 10

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| arr[][] | int | Stores the matrix. |
| m | int | Stores the number of rows and columns of the matrix. |
| i | int | Counter variable. |
| j | int | Counter variable. |
| left | int | Stores the sum of the left diagonal. |
| right | int | Stores the sum of the right diagonal. |
| l | int | Stores the length of the rows and columns of the matrix. |

Program - 7

1. Question -

Write a Program in Java to input a 2-D array of size ‘m\*n’ and print its boundary (border) elements.

1. Program -

import java.util.\*;

public class prog07

{

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns"); //input

int m = sc.nextInt();

int n = sc.nextInt();

int arr[][] = new int[m][n];

System.out.println("Enter the elements of the matrix");

for(int i = 0; i<m; i++) //input the elements of the matrix

{

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

{

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

}

}

System.out.println("Original Matrix:"); //printing the original matrix

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

{

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

{

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

}

System.out.println();

}

System.out.println("Boundary elements:"); //printing the boundary elements

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

{

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

{

if((i>=1)&&(i<m-1)&&(j>=1)&&(j<n-1))

{

System.out.print(" " + "\t");

}

else

{

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

}

}

System.out.println();

}

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of rows and columns");

int m = sc.nextInt();

int n = sc.nextInt();

int arr[][] = new int[m][n];

System.out.println("Enter the elements of the matrix");

Step 3 - for(int i = 0; i<m; i++)

{

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

{

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

}

}

Step 4 - System.out.println("Original Matrix:");

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

{

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

{

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

}

System.out.println();

}

Step 5 - System.out.println("Boundary elements:");

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

{

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

{

if((i>=1)&&(i<m-1)&&(j>=1)&&(j<n-1))

{

System.out.print(" " + "\t");

}

else

{

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

}

}

System.out.println();

}

Step 6 - Stop

1. Output -

Enter the number of rows and columns

4

5

Enter the elements of the matrix

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

Original Matrix:

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

Boundary elements:

1 2 3 4 5

6 10

11 15

16 17 18 19 20

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| m | int | Stores the number of rows of the matrix. |
| n | int | Stores the number of columns of the matrix. |
| arr[][] | int | Stores the matrix containing the numbers. |
| i | int | Counter variable. |
| j | int | Counter variable. |

Program - 8

1. Question -

A square matrix is said to be a Magic Square, if the sum of each row, each column and each diagonal is same. Write a program to enter an integer number ‘n’. Create a magic square of size ‘n\*n’. Finally, print the elements of the matrix as Magic Square.

Note: n <= 5

1. Program -

import java.util.\*;

public class prog08

{

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix"); //input

int size = sc.nextInt();

int arr1[][] = {{1},{}}; //1x1 magic matrix

int arr3[][] = {{8,1,6},{3,5,7},{4,9,2}}; //3x3 magic matrix

int arr4[][] = {{16,2,3,13},{5,11,10,8},{9,7,6,12},{4,14,15,1}}; //4x4 magix matrix

int arr5[][] = {{17,24,1,8,15},{23,5,7,14,16},{4,6,13,20,22},{10,12,19,21,3},{11,18,25,2,9}}; //5x5 magic matrix

if(size == 2)

{

System.out.println("A magic matrix of size 2x2 does not exist");

}

else if(size<1 || size>5)

{

System.out.println("Matrix size is out of range");

}

else if(size == 1) //1x1 magic matrix

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr1[i][j]);

}

}

}

else if(size == 3) //3x3 magic matrix

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr3[i][j] + "\t");

}

System.out.println();

}

}

else if(size == 4) //4x4 magic matrix

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr4[i][j] + "\t");

}

System.out.println();

}

}

else if(size == 5) //5x5 magic matrix

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr5[i][j] + "\t");

}

System.out.println();

}

}

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix");

int size = sc.nextInt();

int arr1[][] = {{1},{}};

int arr3[][] = {{8,1,6},{3,5,7},{4,9,2}};

int arr4[][] = {{16,2,3,13},{5,11,10,8},{9,7,6,12},{4,14,15,1}};

int arr5[][] = {{17,24,1,8,15},{23,5,7,14,16},{4,6,13,20,22},{10,12,19,21,3},{11,18,25,2,9}};

Step 3 - if(size == 2)

{

System.out.println("A magic matrix of size 2x2 does not exist");

}

else if(size<1 || size>5)

{

System.out.println("Matrix size is out of range");

}

else if(size == 1)

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr1[i][j]);

}

}

}

else if(size == 3)

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr3[i][j] + "\t");

}

System.out.println();

}

}

else if(size == 4)

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr4[i][j] + "\t");

}

System.out.println();

}

}

else if(size == 5)

{

System.out.println("Magic Matrix of size " + size);

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

{

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

{

System.out.print(arr5[i][j] + "\t");

}

System.out.println();

}

}

Step 4 - Stop

1. Output -

Enter the size of the matrix

3

Magic Matrix of size 3

8 1 6

3 5 7

4 9 2

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| size | int | Stores the size of the matrix. |
| arr1[][] | int | Stores the 1x1 magic matrix. |
| arr3[][] | int | Stores the 3x3 magic matrix. |
| arr4[][] | int | Stores the 4x4 magic matrix. |
| arr5[][] | int | Stores the 5x5 magic matrix. |
| i | int | Counter variable. |
| j | int | Counter variable. |

Program - 9

1. Question -

Write a program to declare a square matrix A[ ][ ] of order ‘n’. Allow the user to input positive integers into this matrix. Perform the following tasks on the matrix:

(i) Output the original matrix.  
(ii) Find the SADDLE POINT for the matrix. If the matrix has no saddle point, output the message “NO SADDLE POINT”.

[Note: A saddle point is an element of the matrix such that it is the minimum element for the row to which it belongs and the maximum element for the column to which it belongs. Saddle point for a given matrix is always unique.]

1. Program -

import java.util.\*;

public class prog09

{

int n;

int arr[][];

prog09(int n1) //parameterized constructor

{

n=n1;

arr= new int[n][n];

}

void input() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

void display() //displays the original matrix

{

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

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

{

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

{

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

}

System.out.println();

}

}

void saddlepoint() //finds the saddle point

{

int sp = 0;

int min;

int max;

int minarr[] = new int[n]; //array of minimum elements from all rows

int maxarr[] = new int[n]; //array of maximum elements from all columns

int counter = 0;

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

{

min = arr[i][0];

max = arr[0][i];

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

{

if(arr[i][j]<min)

{

min = arr[i][j];

}

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

{

max = arr[j][i];

}

}

minarr[i] = min;

maxarr[i] = max;

}

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

{

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

{

if(minarr[i] == maxarr[j])

{

sp = minarr[i]; //common element of minarr and maxarr is the saddle point

counter++;

}

}

}

if(counter == 0)

{

System.out.println("No Saddle Point");

}

else

{

System.out.println("Saddle Point = " + sp);

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix");

int size = sc.nextInt();

prog09 obj = new prog09(size);

obj.input();

obj.display();

obj.saddlepoint();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int n, int arr[][]

Step 3 - prog09(int n1)

{

n=n1;

arr= new int[n][n];

}

Step 4 - void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Step 5 - void display()

{

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

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

{

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

{

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

}

System.out.println();

}

}

Step 6 - void saddlepoint()

{

int sp = 0;

int min;

int max;

int minarr[] = new int[n];

int maxarr[] = new int[n];

int counter = 0;

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

{

min = arr[i][0];

max = arr[0][i];

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

{

if(arr[i][j]<min)

{

min = arr[i][j];

}

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

{

max = arr[j][i];

}

}

minarr[i] = min;

maxarr[i] = max;

}

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

{

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

{

if(minarr[i] == maxarr[j])

{

sp = minarr[i];

counter++;

}

}

}

if(counter == 0)

{

System.out.println("No Saddle Point");

}

else

{

System.out.println("Saddle Point = " + sp);

}

}

Step 7 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix");

int size = sc.nextInt();

prog09 obj = new prog09(size);

obj.input();

obj.display();

obj.saddlepoint();

}

Step 8 - Stop

1. Output -

Enter the size of the matrix

3

Enter the elements of the matrix

4

5

6

7

8

9

5

1

3

Matrix:

4 5 6

7 8 9

5 1 3

Saddle Point = 7

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| n | int | Stores the size of the matrix. |
| arr[][] | int | Stores the matrix. |
| n1 | int | Stores the size of the matrix and assigns it to n. |
| i | int | Counter variable. |
| j | int | Counter variable. |
| sp | int | Stores the saddle point. |
| min | int | Stores the minimum element from each row. |
| max | int | Stores the maximum element from each column. |
| minarr[] | int | Stores all the minimum elements from all the rows. |
| maxarr[] | int | Stores all the maximum elements from all the columns. |
| counter | int | It is used to show whether a saddle point was found or not in the matrix. |
| size | int | Stores the size of the matrix input by the user. |

Program - 10

1. Question -

Write a Program in Java to input a 2-D square matrix and check whether it is an Upper Triangular Matrix or not.

Upper Triangular Matrix : An Upper Triangular matrix is a square matrix in which all the entries below the main diagonal are zero. The entries above or on the main diagonal themselves may or may not be zero.

1. Program -

import java.util.\*;

public class prog10

{

int n;

int arr[][];

prog10(int n1) //parameterized constructor

{

n=n1;

arr= new int[n][n];

}

void input() //input

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

void display() //displays the original matrix

{

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

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

{

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

{

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

}

System.out.println();

}

}

void UTMatrix() //checks if the matrix is an upper triangular matrix or not

{

int counter = 0;

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

{

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

{

if(i<=j)

{

continue;

}

if(arr[i][j]!=0)

{

counter++;

}

}

}

if(counter == 0)

{

System.out.println("It is an Upper Triangular Matrix");

}

else

{

System.out.println("It is not an Upper Triangular Matrix");

}

}

public static void main() //main method

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix");

int size = sc.nextInt();

prog10 obj = new prog10(size);

obj.input();

obj.display();

obj.UTMatrix();

}

}

1. Algorithm -

Step 1 - Start

Step 2 - Declare int n, int arr[][]

Step 3 - prog10(int n1)

{

n=n1;

arr= new int[n][n];

}

Step 4 - void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the elements of the matrix");

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

{

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

{

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

}

}

}

Step 5 - void display()

{

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

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

{

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

{

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

}

System.out.println();

}

}

Step 6 - void UTMatrix()

{

int counter = 0;

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

{

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

{

if(i<=j)

{

continue;

}

if(arr[i][j]!=0)

{

counter++;

}

}

}

if(counter == 0)

{

System.out.println("It is an Upper Triangular Matrix");

}

else

{

System.out.println("It is not an Upper Triangular Matrix");

}

}

Step 7 - public static void main()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the matrix");

int size = sc.nextInt();

prog10 obj = new prog10(size);

obj.input();

obj.display();

obj.UTMatrix();

}

Step 8 - Stop

1. Output -

Enter the size of the matrix

4

Enter the elements of the matrix

1

2

3

4

0

5

6

7

0

0

8

9

0

0

0

10

Matrix:

1 2 3 4

0 5 6 7

0 0 8 9

0 0 0 10

It is an Upper Triangular Matrix

1. VDT -

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| n | int | Stores the size of the matrix. |
| arr[][] | int | Stores the matrix. |
| n1 | int | Stores the size of the matrix and assigns it to n. |
| i | int | Counter variable. |
| j | int | Counter variable. |
| counter | int | Stores the value 0 if the matrix is an upper triangular matrix, and stores a value greater than 0 if the matrix is not an upper triangular matrix. |
| size | int | Stores the size of the matrix input by the user. |