An array stores multiple values in a single variable. Arrays are a kind of data structure that can store a fixed-size sequential collection of elements of the same type (homogeneous) values. In Java, all arrays are dynamically allocated. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables. the actual values are stored in contiguous memory locations. In the case of class objects, the actual objects are stored in a heap segment.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index number

**Arrays are of two types:**
1) One-dimensional arrays
2) Multidimensional arrays

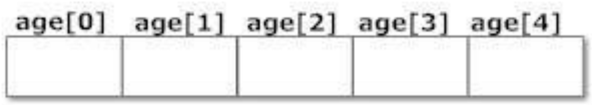
**One dimensional array:**
Declaration of one-dimensional array:
To declare an array in Java, a programmer specifies the type of the elements and the number of elements required by an array as follows;

**Syntax:** **data_type[]** ArrayName **= new data_type[size]** ;

**Example:** **int []** age = **new int[5];**
Here, the name of array is age. The size of array is 5, i.e., there are 5 items (elements) of array age. All elements in an array are of the same type (**int,** in this case).

## Array elements

Size of array defines the number of elements in an array. Each element of array can be accessed and used by user according to the need of program. For example: **int** age[5];



Array elements

***Note*** that, the first element is numbered 0, second element 1 and so on

**Initialization of one-dimensional array:**
Arrays can be initialized at declaration time in this source code as:

**Int[]** age =**{**12,14,34,13,19**};**

It is not necessary to define the size of arrays during declare with initialization but data not initialize declare time in array then array size must be define.

**int** age[ ] =**{**12,14,34,13,19**};** // array declare with initialization

**or**

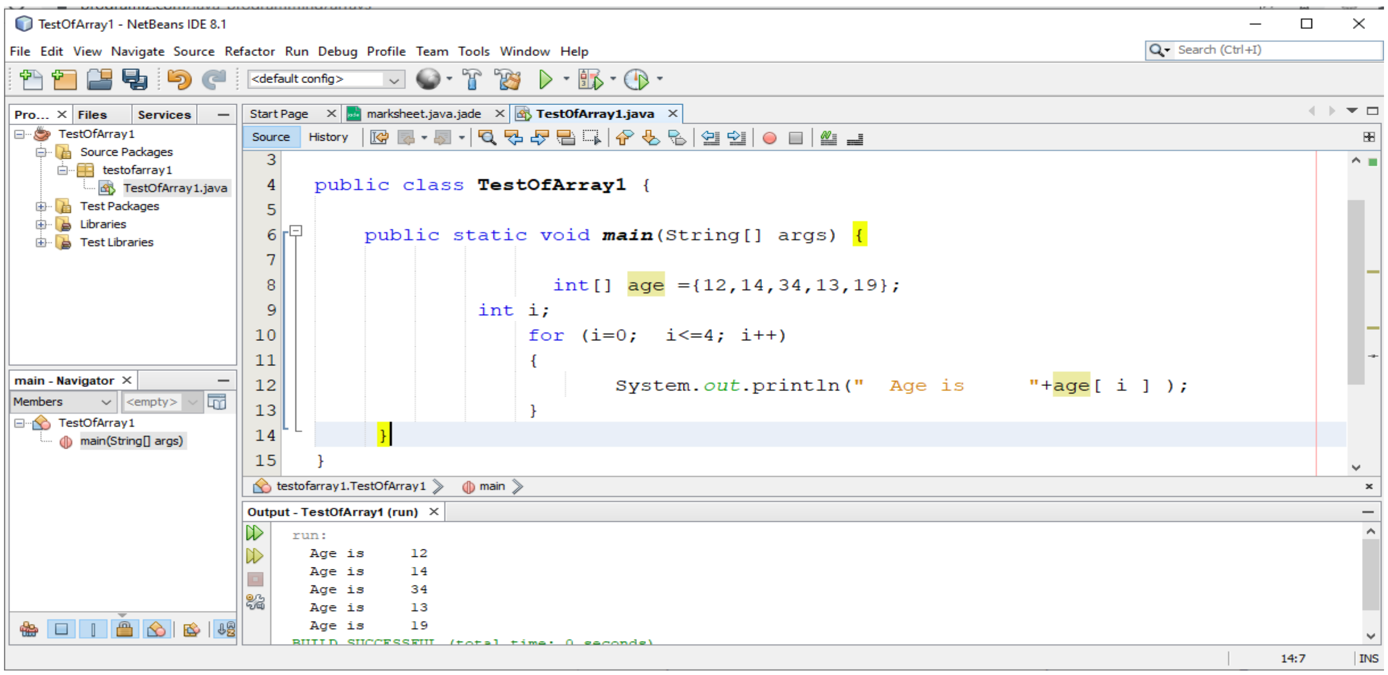
**int[]** age **= new int[5];** **//** array declare without initialization

## Accessing array elements

In Java programming, arrays can be accessed and treated like variables in Java.

**Example:**

```
int[] age ={12,14,34,13,19};
int i;
for (i=0;  i<=4; i++)
{
    System.out.println(“  Age is    “+age[ i ] );
}
```

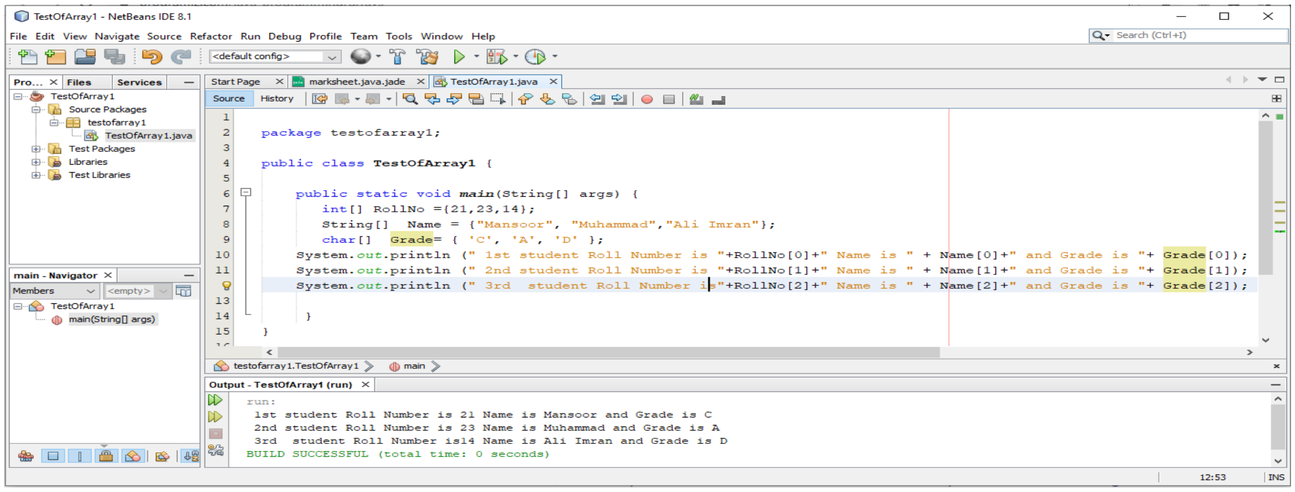


```
public class TestOfArray1 {

    public static void main(String[] args) {

                 int[] age ={12,14,34,13,19};
             int i;
                 for (i=0;  i<=4;  i++)
                 {
                        System.out.println("  Age is     "+age[ i ] );
                 }
    }
}
```

```
run:
  Age is     12
  Age is     14
  Age is     34
  Age is     13
  Age is     19
BUILD SUCCESSFUL (total time: 0 seconds)
```

The following example creates an indexed array If you have a list of students record items (a list of Roll Number, Name and Grade, storing the RollNo , Name and Grade in three array variables to it, and then prints containing the array values.

```
int[] RollNo ={21,23,14};
String[] Name = { "Mansoor ", "Muhammad ","Ali Imran"};
char[] Grade= { 'C', 'A', 'D' };
System.out.println (" 1st student Roll Number is "+RollNo[0]+ " Name is "+ Name[0]+ " and Grade is "+ Grade[0]);
System.out.println (" 2nd student Roll Number is "+RollNo[1]+ " Name is "+ Name[1]+ " and Grade is "+ Grade[1]);
System.out.println (" 3rd  student Roll Number is"+RollNo[2]+ "Name is " + Name[2]+ " and Grade is "+ Grade[2]);
```



```
package testofarray1;

public class TestOfArray1 {

    public static void main(String[] args) {
       int[] RollNo ={21,23,14};
       String[]  Name = {"Mansoor", "Muhammad","Ali Imran"};
       char[]  Grade= { 'C', 'A', 'D' };
    System.out.println (" 1st student Roll Number is "+RollNo[0]+" Name is " + Name[0]+" and Grade is "+ Grade[0]);
    System.out.println (" 2nd student Roll Number is "+RollNo[1]+" Name is " + Name[1]+" and Grade is "+ Grade[1]);
    System.out.println (" 3rd  student Roll Number is"+RollNo[2]+" Name is " + Name[2]+" and Grade is "+ Grade[2]);

      }
}
```

Output - TestOfArray1 (run)

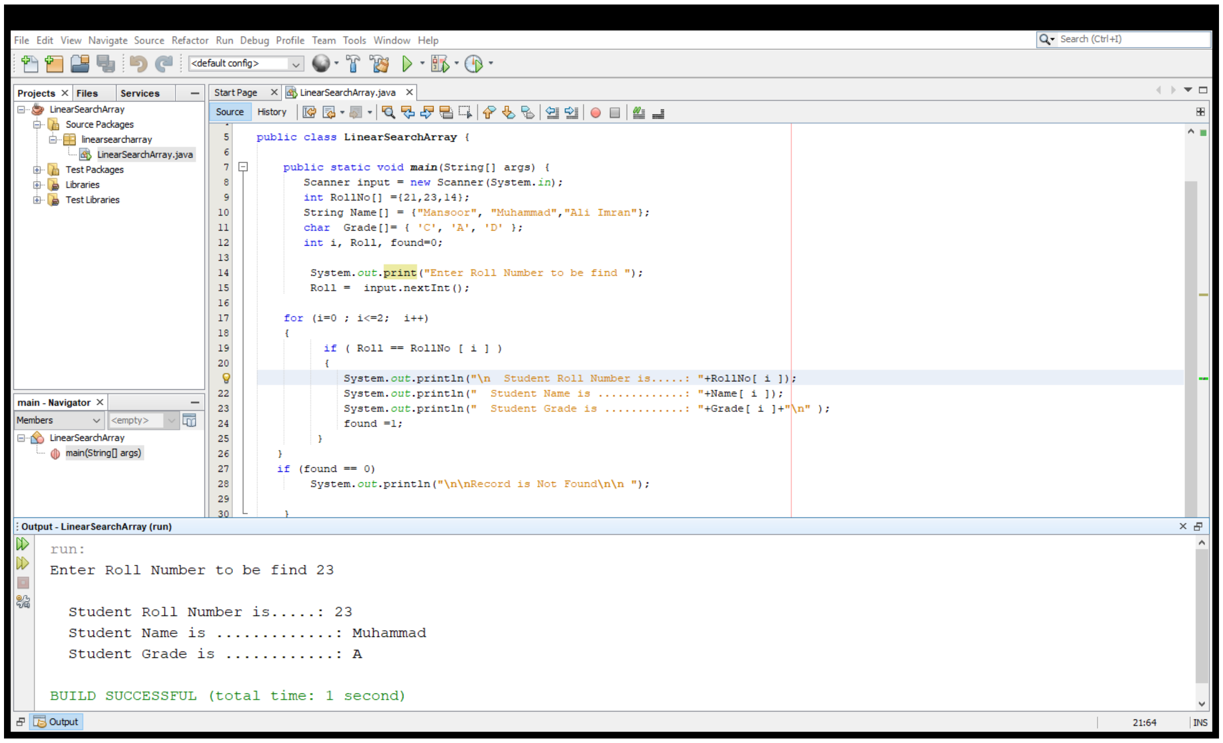
```
run:
 1st student Roll Number is 21 Name is Mansoor and Grade is C
 2nd student Roll Number is 23 Name is Muhammad and Grade is A
 3rd  student Roll Number is14 Name is Ali Imran and Grade is D
BUILD SUCCESSFUL (total time: 0 seconds)
```

The index can be assigned values manually like this:

```
RollNo[2] = 67;
Grade[2] = 'B';
Name[2] ="Farhan";
```

Here you will get program for linear search in Java. In linear search Program, we compare targeted element with each element of the array. If the element is found then its print student record is displayed.

```
package linearsearcharray;

import java.util.Scanner;

public class LinearSearchArray {

  public static void main(String[] args) {
     Scanner input = new Scanner(System.in);
     int RollNo[] ={21,23,14};
     String Name[] = {"Mansoor", “Muhammad","Ali Imran"};
     char  Grade[]= { 'C', 'A', 'D' };
     int i, Roll, found=0;

      System.out.print(“Enter Roll Number to be find “);
      Roll =  input.nextInt();

  for (i=0 ; i<=2;  i++)
  {
          if ( Roll == RollNo [ i ] )
          {
             System.out.println("\n  Student Roll Number is.....: "+RollNo[ i ]);
             System.out.println("  Student Name is .............: "+Name[ i ]);
             System.out.println("  Student Grade is ............: "+Grade[ i ]+ "\n" );
             found =1;
          }
  }
  if (found == 0)
        System.out.println("\n\nRecord is Not Found\n\n ");

   }
}
```



```
public class LinearSearchArray {

    public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      int RollNo[] ={21,23,14};
      String Name[] = {"Mansoor", "Muhammad","Ali Imran"};
      char  Grade[]= { 'C', 'A', 'D' };
      int i, Roll, found=0;

       System.out.print("Enter Roll Number to be find ");
       Roll =  input.nextInt();

    for (i=0 ; i<=2;  i++)
    {
         if ( Roll == RollNo [ i ] )
         {
            System.out.println("\n  Student Roll Number is.....: "+RollNo[ i ]);
            System.out.println("  Student Name is ............: "+Name[ i ]);
            System.out.println("  Student Grade is ...........: "+Grade[ i ]+"\n" );
            found =1;
         }
    }
   if (found == 0)
       System.out.println("\n\nRecord is Not Found\n\n ");

    }
```

```
run:
Enter Roll Number to be find 23

  Student Roll Number is.....: 23
  Student Name is ............: Muhammad
  Student Grade is ...........: A

BUILD SUCCESSFUL (total time: 1 second)
```

### Multi-Dimensional array

In Java, we can create an array of an array, known as a multidimensional array. Data in multidimensional arrays are stored in tabular form There are two Types Dimensional array double **(**2D**)** dimensional and three **(**3D**)** dimensional array.

### Two (2D) dimensional Array

The two dimensional arrays are such type of arrays which stores an multiple column at each row and column index number instead of single element. It can be created using nested array. These type of arrays can be used to store similar type of elements, but the index is always a number.

## Declaration of Two-Dimensional array:

**Syntax:**

*Data_type [num_of_rows][num_of_column]* ***arrayName****;*

**OR**

*Data_type [ y ][ x ]* ***arrayName****;*

Where data type can be any valid Java data type and **arrayName** will be a valid Java identifier/Variable.

**Example of Two dimensional array:**

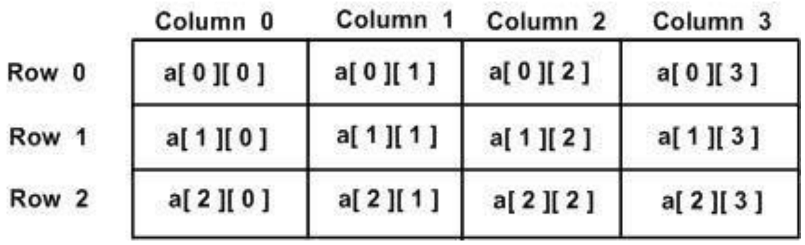
int[][] twoDarr = new int[10][20];

Size of multidimensional arrays: The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

The array int[][] x = new int[10][20] can store a total of **(**10*20**)** = 200 elements.

The Ary two dimensional array can be considered as a table which will have y number of rows and x number of columns. A two-dimensional array Ary, which contains three rows and four columns can be shown as follows.

int Ary[3][4]



| | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | a[ 0 ][ 0 ] | a[ 0 ][ 1 ] | a[ 0 ][ 2 ] | a[ 0 ][ 3 ] |
| Row 1 | a[ 1 ][ 0 ] | a[ 1 ][ 1 ] | a[ 1 ][ 2 ] | a[ 1 ][ 3 ] |
| Row 2 | a[ 2 ][ 0 ] | a[ 2 ][ 1 ] | a[ 2 ][ 2 ] | a[ 2 ][ 3 ] |

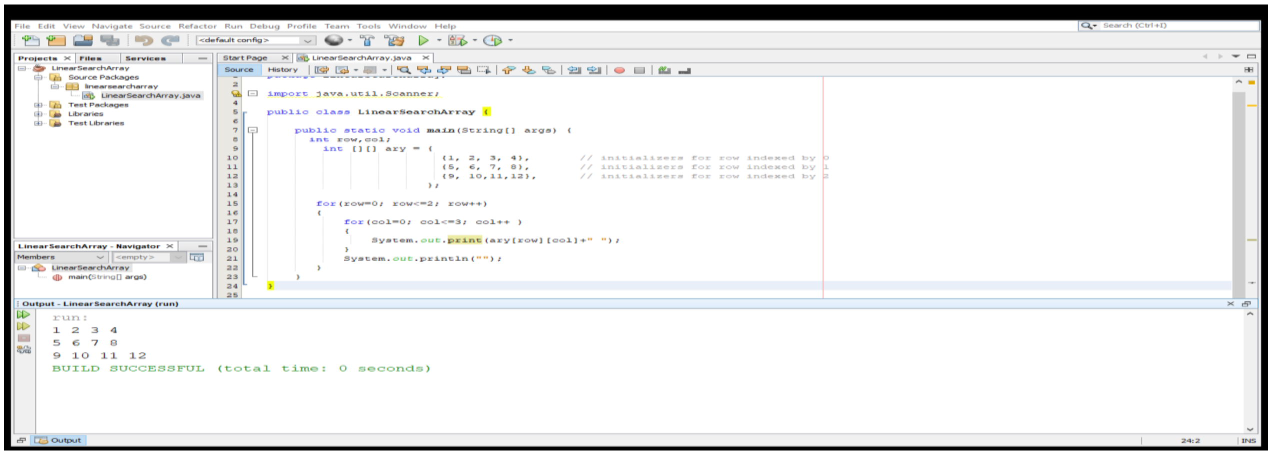
**Initializing two-dimensional arrays:**

Two dimensional arrays may be initialized by specifying bracketed values for each row. Following is an array with 3 rows and each row has 4 columns.

**Example:**

```
int [][] ary = {
  {1, 2, 3, 4},          // initializers for row indexed by 0
  {5,6, 7, 8},           // initializers for row indexed by 1
  {9,10, 11, 12},         // initializers for row indexed by 2
  };
```

| | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | 1 | 2 | 3 | 4 |
| Row 1 | 5 | 6 | 7 | 8 |
| Row 2 | 9 | 10 | 11 | 12 |



```
import java.util.Scanner;

public class LinearSearchArray {

    public static void main(String[] args) {
      int row,col;
        int [][] ary = {
                              {1, 2, 3, 4},        // initializers for row indexed by 0
                              {5, 6, 7, 8},        // initializers for row indexed by 1
                              {9, 10,11,12},       // initializers for row indexed by 2
                             };

       for(row=0; row<=2; row++)
       {
           for(col=0; col<=3; col++ )
           {
               System.out.print(ary[row][col]+" ");
           }
           System.out.println("");
       }
    }
}
```

```
run:
1 2 3 4
5 6 7 8
9 10 11 12
BUILD SUCCESSFUL (total time: 0 seconds)
```

The nested braces are optional. The following initialization is equivalent to it

```
        int ary[][] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

**Accessing Two-Dimensional Array Elements:**

An element in a two-dimensional array is accessed by using the subscripts, i.e., row index and column index of the array.

```
int val1,val2;

   val1 = a[0][0];     // value is 1
   val2 = a[1][2];     // value is 7
```

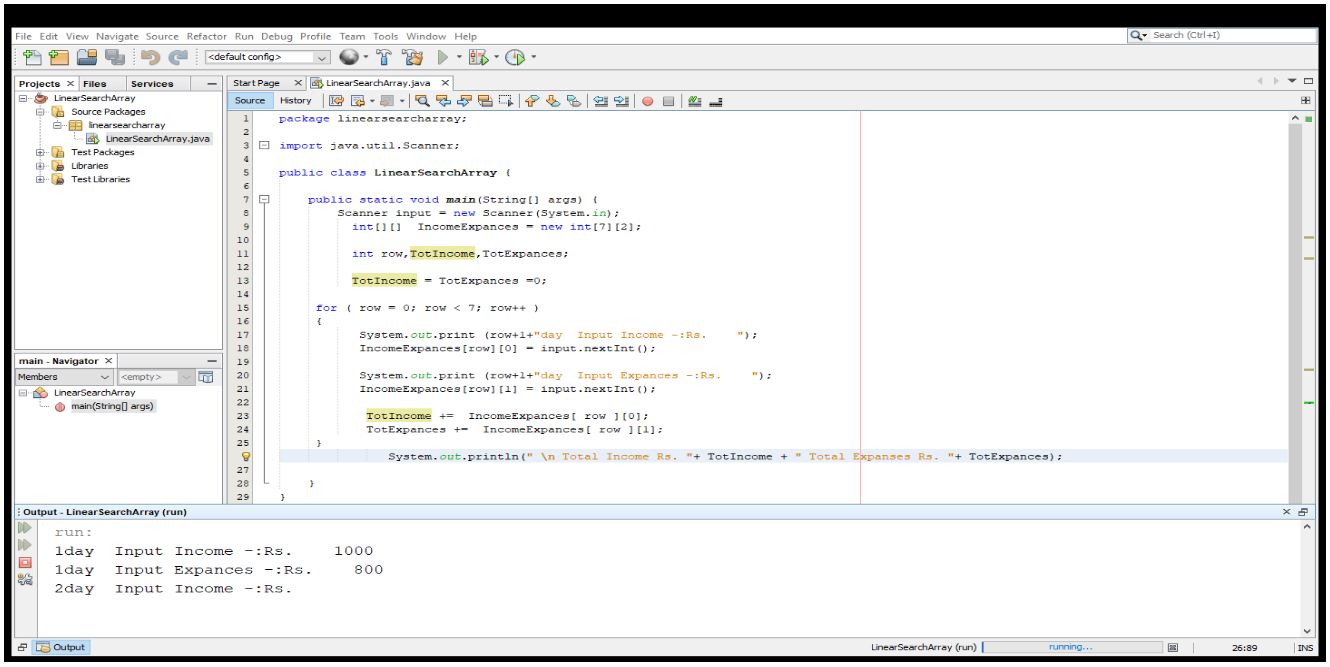
| | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | 1 | 2 | 3 | 4 |
| Row 1 | 5 | 6 | 7 | 8 |
| Row 2 | 9 | 10 | 11 | 12 |

The above 1st statement will take the element from the 0 row and 0 column access value is 1 of the array value put in ***val1*** variable.

The above 2nd statement will take the element from the 1 row and 2 column access value is 7 of the array value put in ***val2*** variable

The following program example to input income and expanses of seven days and calculate sum of total income and total expanses with the help of double dimension array.

```
 Scanner input = new Scanner(System.in);
  int[][]  IncomeExpances = new int[7][2];

  int row,TotIncome,TotExpances;

  TotIncome = TotExpances =0;

 for ( row = 0; row < 7; row++ )
{
   System.out.print (row+1+"day  Input Income -:  Rs.  ");
   IncomeExpances[row][0] = input.nextInt();

   System.out.print (row+1+"day  Input Expances -:  Rs.  ");
   IncomeExpances[row][1] = input.nextInt();

    TotIncome +=  IncomeExpances[ row ][0];
    TotExpances +=  IncomeExpances[ row ][1];

}
     System.out.println(" \n Total Income Rs. "+ TotIncome + " Total Expanses Rs. "+ TotExpances);
```



```java
package linearsearcharray;

import java.util.Scanner;

public class LinearSearchArray {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
          int[][]  IncomeExpances = new int[7][2];

          int row,TotIncome,TotExpances;

          TotIncome = TotExpances =0;

     for ( row = 0; row < 7; row++ )
     {
           System.out.print (row+1+"day  Input Income -:Rs.    ");
           IncomeExpances[row][0] = input.nextInt();

           System.out.print (row+1+"day  Input Expances -:Rs.    ");
           IncomeExpances[row][1] = input.nextInt();

            TotIncome +=  IncomeExpances[ row ][0];
            TotExpances +=  IncomeExpances[ row ][1];
     }
               System.out.println(" \n Total Income Rs. "+ TotIncome + " Total Expanses Rs. "+ TotExpances);

    }
}
```

Output - LinearSearchArray (run)

```
run:
1day  Input Income -:Rs.    1000
1day  Input Expances -:Rs.    800
2day  Input Income -:Rs.
```

## Exercise

### Theory Questions

1. What is array?
2. What is the difference between a single dimensional array and a multidimensional array?
3. Describe the two **(**2D**)** dimension array.

### Practical Questions

1. What would be the output of the following Program?

```
     int salary[ ] = {50000,40000,20000,30000,60000};
    for (int i=4; i>=0; i--)
   {
     System.out.println( " Employee "+i + " and Salary is  "+ salary[ i ]);
    }
 }
```

2. Write a program that takes 10 integers as input and prints their sum by using array.
3. Create a MarkSheet for 10 students using arrays which contains the following;

   Roll Number

   Name

   Marks of 3 subjects **(** Maths, English and Urdu**)** and display following form

```
                         Student List Of Mark Sheet
--------------------------------------------------------------------------------------------------------------
Roll # | Student Name | Math's | English | Urdu | Obtain Marks | Percentage | Grade |
______________________________________________________________________________________________________________
99    | xxxxxxxxxxxx  | 99     | 99      | 99  |    999       |    99.99   | xx    |
99    | xxxxxxxxxxxx  | 99     | 99      | 99  |    999       |    99.99   | xx    |
99    | xxxxxxxxxxxx  | 99     | 99      | 99  |    999       |    99.99   | xx    |
:
```

4. Write a program to sort an integer array in ascending order.
5. Write a program that adds up two 2x2 arrays A and B stores the sum in third array C.

## Objective MCQ's

1. Which declaration of array is correct?
   a) int Number[ ];
   b) Number int[ ];
   c) int[] Number = new int[5];
   d) int Number(5);

2. The first element number start always from ___ index number.
   a) 1
   b) 0
   c) 2
   d) NULL

3. What is the correct syntax for declaring and initializing an associative array?
   a) char Names[ ][ ] = array{“Asif” - “Muhammad” - “Kamran” - “Umer Ahmed”};
   b) char Names[4][10 ] = {Asif , Muhammad , Kamran , Umer Ahmed};
   c) char Names[4][10 ] = (“Asif” , “Muhammad” , “Kamran” , “Umer Ahmed”);
   d) char Names[4][10 ] = {“Asif”, “Muhammad”, “Kamran”, “Umer Ahmed”};

4. When you pass an array as an argument to a function, what is actually passed?
   a) The values of all elements in array
   b) The address of the array
   c) The address of the first element in the array
   d) The value of first element in the array

5. The following declares a two dimensional float array name ***No*** with 5 rows and 3 column
   a) float No[3,5];
   b) float No[5,3];
   c) float No[3][5];
   d) float No[5][3];

6. An array is a collection of variable's in C/C++
   a) Different data types scattered throughout memory.
   b) Similar data types scattered throughout memory.
   c) Similar data types place continuously in memory.
   d) Different data types place continuously in memory.

7. Which is last element of array, if we declared integer array int Salary [10]; ?
   a) Salary[0];
   b) Salary[1];
   c) Salary[10 ];
   d) Salary[9];