**PROGRAM NO : 1**

**AIM:** Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name Product and members as pcode,pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

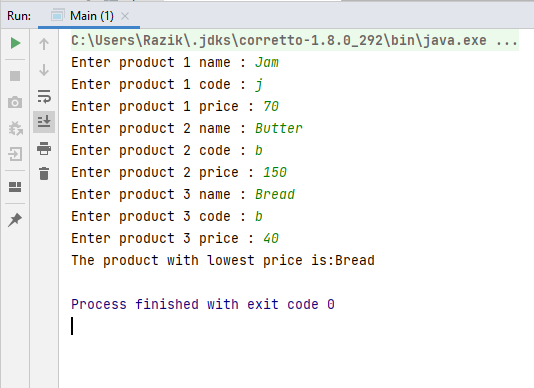
Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  }  package com.lab\_cycles.co1.q1;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  } |
| Product.java | public class Product {  public String pcode;  public String pname;  public int price;  } |

**OUTPUT:**

****

**PROGRAM NO : 2**

**AIM:** Read 2 matrices from the console and perform matrix addition.

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class having name AddMatrix.

Step 3: Read row number(m),column number (n) and initialize the  double dimensional arrays mat1[][],mat2[][],res[][] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.

for i=0 to i<m

for j=0 to j<n

mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

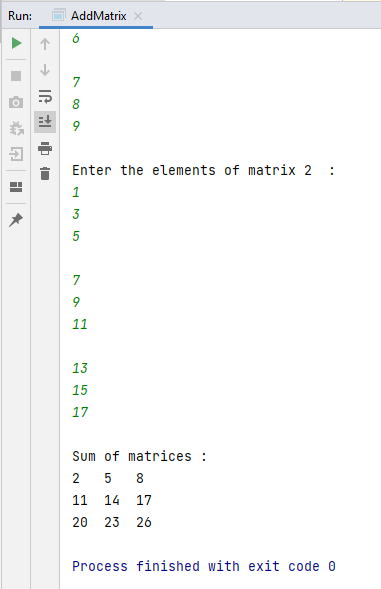
Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| AddMatrix  .java | import java.util.Scanner;  public class AddMatrix {  public static void main(String args[])  {  int row, col,i,j;  Scanner in = new Scanner(System.in);  System.out.println("Enter the number of rows : ");  row = in.nextInt();  System.out.println("Enter the number columns : ");  col = in.nextInt();  int mat1[][] = new int[row][col];  int mat2[][] = new int[row][col];  int res[][] = new int[row][col];  System.out.println("Enter the elements of matrix 1 : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat1[i][j] = in.nextInt();  System.out.println();  }  System.out.println("Enter the elements of matrix 2 : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat2[i][j] = in.nextInt();  System.out.println();  }  for ( i= 0 ; i < row ; i++ )  for ( j= 0 ; j < col ;j++ )  res[i][j] = mat1[i][j] + mat2[i][j] ;  System.out.println("Sum of matrices : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  System.out.print(res[i][j]+"\t");  System.out.println();  }  }  } |

**OUTPUT:**

****

**PROGRAM NO : 3**

**AIM:** Add complex numbers.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and imaginary number.

Step 3: Define a function ComplexNumber and add values to variables.

Step 4 : Define a function ComplexNumber sum to add complex number using 3rd ComplexNumber object and return the value.

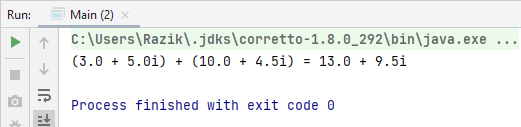
Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| ComplexNumber. java | public class ComplexNumber {  double real,imaginary;  ComplexNumber(double real,double imaginary)  {  this.real = real;  this.imaginary = imaginary;  }  ComplexNumber addComplexNumber(ComplexNumber complexNumber1,ComplexNumber complexNumber2)  {  double real = complexNumber1.real+ complexNumber2.real;  double imaginary = complexNumber1.imaginary+complexNumber2.imaginary;  ComplexNumber complexNumber = new ComplexNumber(real,imaginary);  return complexNumber;  }  String displayComplexNumber()  {  return this.real+" + "+this.imaginary+"i";  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  ComplexNumber complexNumber1 = new ComplexNumber(3,5);  ComplexNumber complexNumber2 = new ComplexNumber(10,4.5);  ComplexNumber complexNumber3 = new ComplexNumber(0,0);  complexNumber3 = complexNumber3.addComplexNumber(complexNumber1,complexNumber2);  System.out.println("("+complexNumber1.displayComplexNumber()+") + ("+complexNumber2.displayComplexNumber()+") = "+complexNumber3.displayComplexNumber());  }  } |

**OUTPUT:**



**PROGRAM NO : 4**

**AIM:** Read a matrix from the console and check whether it is symmetric or not.

**ALGORITHM:**

Step 1: Start.

Step 2 : Read row number,column number and initialize the  double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

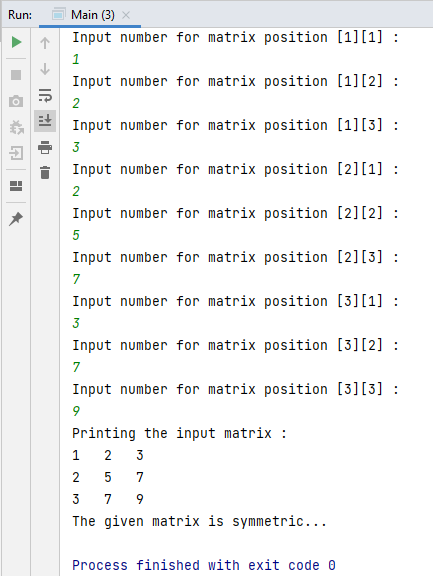
Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main  {  public static void main(String[] args)  {  Scanner sc = new Scanner(System.in);  System.out.println("Enter the no. of rows : ");  int rows = sc.nextInt();  System.out.println("Enter the no. of columns : ")  int cols = sc.nextInt();  int matrix[][] = new int[rows][cols];  System.out.println("Enter the elements :");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  System.out.println("Input number for matrix position ["+(i+1)+"]["+(j+1)+"] : ");  matrix[i][j] = sc.nextInt();  }  }  System.out.println("Printing the input matrix :");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  System.out.print(matrix[i][j]+"\t");  }  System.out.println();  }  //Checking the input matrix for symmetric  if(rows != cols)  {  System.out.println("The given matrix is not a square matrix, so it can't be symmetric.");  }  else  {  boolean symmetric = true;  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  if(matrix[i][j] != matrix[j][i])  {  symmetric = false;  break;  }  }  }  if(symmetric)  {  System.out.println("The given matrix is symmetric...");  }  else  {  System.out.println("The given matrix is not symmetric...");  }  }  sc.close();  }  } |

**OUTPUT:**



**PROGRAM NO : 5**

**AIM:** Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members no\_cores,manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

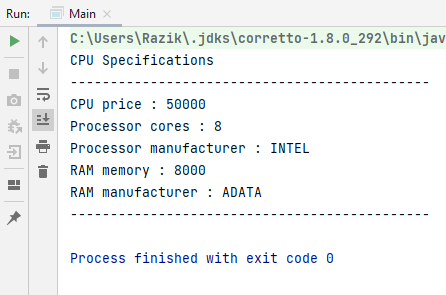
Step 5: Create objects in corresponding classes and display it’s details.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CPU.java | public class CPU {  int price;  Processor processor = new Processor();  RAM ram = new RAM();  static class Processor  {  int cores;  String manufacturer;  }  static class RAM  {  int memory;  String manufacturer;  }  CPU(int cpuPrice,int cores,String cpuManufacturer,int memory,String ramManufacturer){  this.price = cpuPrice;  this.processor.cores = cores;  this.processor.manufacturer = cpuManufacturer;  this.ram.memory = memory;  this.ram.manufacturer = ramManufacturer;  }  void printCPUSpecifications()  {  System.out.println("CPU Specifications");  System.out.println("---------------------------------------------");  System.out.println("CPU price : "+this.price);  System.out.println("Processor cores : "+this.processor.cores);  System.out.println("Processor manufacturer : "+this.processor.manufacturer);  System.out.println("RAM memory : "+this.ram.memory);  System.out.println("RAM manufacturer : "+this.ram.manufacturer);  System.out.println("---------------------------------------------");  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  CPU cpu1 = new CPU(50000, 8, "INTEL", 8000, "ADATA");  cpu1.printCPUSpecifications();  }  } |

**OUTPUT:**

****

**PROGRAM NO : 6**

**AIM:** Program to Sort strings.

**ALGORITHM :**

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparision, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

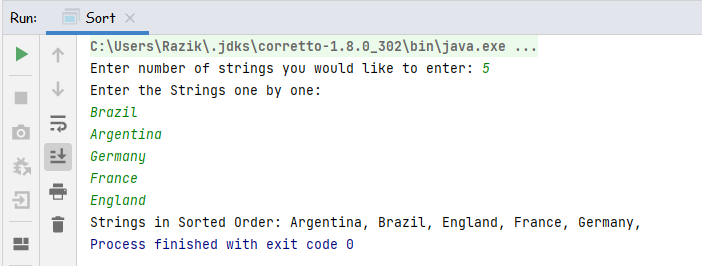
Step 6: Stop

.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Sort.java | import java.util.Scanner;  public class Sort {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  Scanner scan2 = new Scanner(System.in);  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  scan.close();  scan2.close();  //Sorting the strings  for (int i = 0; i < count; i++)  {  for (int j = i + 1; j < count; j++) {  if (str[i].compareTo(str[j])>0)  {  temp = str[i];  str[i] = str[j];  str[j] = temp;  }  }  }  //Displaying the strings after sorting them based on alphabetical order  System.out.print("Strings in Sorted Order: ");  for (int i = 0; i <= count - 1; i++)  {  System.out.print(str[i] + ", ");  }  }  } |

**OUTPUT:**



**PROGRAM NO : 7**

**AIM:** Search an element in an array.

**ALGORITHM :**

Step 1: Start

Step 2: Check each element in the given list with the string provided by the user.

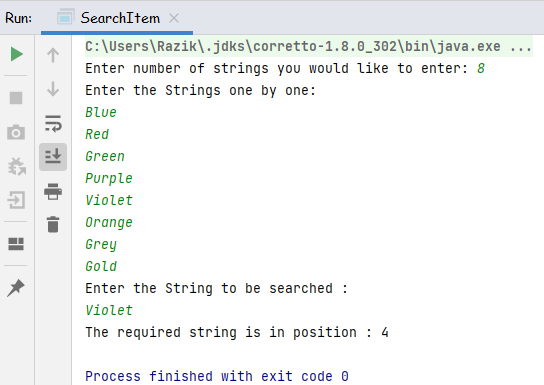
Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| SearchItem.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



**PROGRAM NO : 8**

**AIM:** Perform string manipulations

**ALGORITHM :**

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

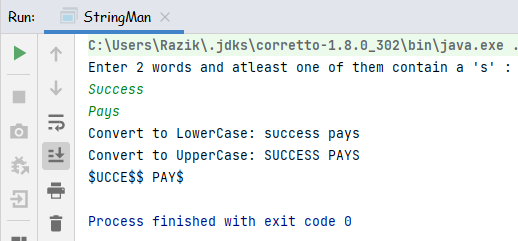
Step 4: Display the combined string after replacing all the ‘s’ & ‘S’ characters with ‘$’ character.

Step 5: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| StringMan.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



**PROGRAM NO : 9**

**AIM:** Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

**ALGORITHM :**

Step 1: Start

Step 2: Search the ‘eNo’ attribute of the list of Employee Objects for the ‘eNo’ provided by the user.

Step 3: If user provided ‘eNo’ is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee {  public String eNo;  public String eName;  public int eSalary;  public Employee(String eNo,String eName,int eSalary)  {  this.eNo = eNo;  this.eName = eName;  this.eSalary = eSalary;  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  System.out.print("Enter count of employee list : ");  int count = Integer.parseInt(scan.nextLine());  Employee[] employees = new Employee[count];  for (int i=0;i<count;i++)  {  System.out.print("Enter Number of employee : ");  String eNo = scan.nextLine();  System.out.print("Enter Name of employee : ");  String eName = scan.nextLine();  System.out.print("Enter Salary of employee : ");  int eSalary = Integer.parseInt(scan.nextLine());  employees[i] = new Employee(eNo,eName,eSalary);  }  System.out.print("Enter the number of employee to be filtered out : ");  String search = scan.nextLine();  for (int i=0;i<count;i++){  if(employees[i].eNo.equals(search))  {  System.out.println("Employee Number : "+employees[i].eNo);  System.out.println("Employee Name : "+employees[i].eName);  System.out.println("Employee Salary : "+employees[i].eSalary);  }  }  }  } |

**OUTPUT:**

