// 1. WAP to check given number is palindrome or not.

/\* Get the number from user

Hold the number in temporary variable

Reverse the number

Compare the temporary number with reversed number

If both numbers are same, print palindrome number

Else print not palindrome number \*/

using System;

namespace SDM

{

class Palindrome

{

static void Main(string[] args)

{

int n, r, sum = 0, temp;

Console.Write("Enter the Number: ");

n = int.Parse(Console.ReadLine());

temp = n;

while (n > 0)

{

r = n % 10;

sum = (sum \* 10) + r;

n = n / 10;

}

if (temp == sum)

Console.Write("Number is Palindrome.");

else

Console.Write("Number is not Palindrome");

Console.ReadKey();

}

}

}

// 2. WAP to demonstrate command line arguments. Addition of two numbers

// goto "Developer command prompt" in Visual studio

// goto file path where file is located

// use CSC followed by file name i.e CSC add.cs

// call file name with parameters i.e "add 10 20 "

// o/p sum=30;

using System;

namespace SDC

{

class cmdLnArgs

{

static void Main(string[] args)

{

int oprnd1, oprnd2;

oprnd1 = Convert.ToInt32(args[0]);

oprnd2 = Convert.ToInt32(args[1]);

Console.WriteLine("Sum = " + (oprnd1 + oprnd2));

Console.ReadKey();

}

}

}

// 3. WAP to find roots of quadratic equation.

/\*Calculate root of Quadratic Equation :

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

Input the value of a : 1

Input the value of b : 2

Input the value of c : 1

O/p

Both roots are equal.

First Root Root1= -1

Second Root Root2= -1 \*/

using System;

namespace QuadraticEqn

{

class Program

{

static void Main(string[] args)

{

int a, b, c;

double d, x1, x2;

Console.Write("\n\n");

Console.Write("Calculate root of Quadratic Equation :\n");

Console.Write("----------------------------------------");

Console.Write("\n\n");

Console.Write("Input the value of a : ");

a = Convert.ToInt32(Console.ReadLine());

Console.Write("Input the value of b : ");

b = Convert.ToInt32(Console.ReadLine());

Console.Write("Input the value of c : ");

c = Convert.ToInt32(Console.ReadLine());

d = b \* b - 4 \* a \* c;

if (d == 0)

{

Console.Write("Both roots are equal.\n");

x1 = -b / (2.0 \* a);

x2 = x1;

Console.Write("First Root Root1= {0}\n", x1);

Console.Write("Second Root Root2= {0}\n", x2);

}

else if (d > 0)

{

Console.Write("Both roots are real and diff-2\n");

x1 = (-b + Math.Sqrt(d)) / (2 \* a);

x2 = (-b - Math.Sqrt(d)) / (2 \* a);

Console.Write("First Root Root1= {0}\n", x1);

Console.Write("Second Root root2= {0}\n", x2);

}

else

Console.Write("Root are imeainary;\nNo Solution. \n\n");

Console.ReadKey();

}

}

}

// 4. WAP to demonstrate boxing and unboxing.

// Demonstration of boxing and unboxing

/\* o/p

Boxing

Before boxing object value

After boxing object value 10

Unboxing

Before unboxing value is 0

After unboxing value is 10

\*/

using System;

namespace boxingUnboxing

{

class Program

{

static void Main(string[] args)

{

// Boxing

int a = 10;

object obj =null;

Console.WriteLine("Boxing");

Console.WriteLine("Before boxing object value " + obj);

// converting value type to reference type

obj = a;

Console.WriteLine("After boxing object value " + obj);

// Unboxing

int x=0;

Console.WriteLine("\nUnboxing");

Console.WriteLine("Before unboxing value is " + x);

// converting refernce type to value type

x =(int) obj;

Console.WriteLine("After unboxing value is " + x);

Console.ReadKey();

}

}

}

// 5. WAP to implement stack operations.

/\* stack operations push,pop,peek, count \*/

using System;

using System.Collections;

namespace StackOperation

{

class Program

{

static void Main(string[] args)

{

// Creating a Stack

int selected,length;

Stack myStack = new Stack();

while (true)

{

Console.WriteLine("\n\nStack operations");

Console.WriteLine("1. Push");

Console.WriteLine("2. Display");

Console.WriteLine("3. Pop");

Console.WriteLine("4. Peek");

Console.WriteLine("5. Count");

Console.WriteLine("6. Exit\n\n");

selected = Convert.ToInt32(Console.ReadLine());

switch (selected)

{

case 1:

Console.WriteLine("Enter how many elements");

length = Convert.ToInt32(Console.ReadLine());

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

{

myStack.Push(Console.ReadLine());

}

break;

case 2:

if (myStack.Count == 0)

Console.WriteLine("stack is empty");

else

foreach (var item in myStack)

{

Console.WriteLine(item);

}

break;

case 3:

if (myStack.Count == 0)

Console.WriteLine("Stack is empty");

else

myStack.Pop();

break;

case 4:

if (myStack.Count == 0)

Console.WriteLine("Stack is empty");

else

Console.WriteLine(myStack.Peek());

break;

case 5:

Console.WriteLine("number of elements in stack is "+myStack.Count);

break;

default:

goto exit;

}

}

exit:

Console.WriteLine("");

}

}

}

// 6. WAP to demonstrate operator overloading.

/\* operator overloading demonstration using + operator\*/

using System;

namespace OperatorOverloading

{

class Line

{

public int Length;

public static Line operator +(Line l1, Line l2)

{

Line objLine = new Line();

objLine.Length = l1.Length + l2.Length;

return objLine;

}

}

class Program

{

static void Main(string[] args)

{

Line l1 = new Line();

Line l2 = new Line();

Console.WriteLine("Enter line1 length");

l1.Length =Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter line2 length");

l2.Length =Convert.ToInt32(Console.ReadLine());

Line l3 = l1 + l2;

Console.WriteLine("sum is " + l3.Length );

Console.ReadKey();

}

}

}

// 7. WAP to find second largest element in single dimensional array.

//7. Program to find second largest number in single dimensional array

using System;

namespace SecondLargest

{

class Program

{

static void Main(string[] args)

{

int[] arr = new int[10];

int i,j, first, second,n;

Console.WriteLine("Enter how many elments");

n = Convert.ToInt32(Console.ReadLine());

//There should be atleast two elements

if (n < 2)

{

Console.WriteLine("There should be atleast two elements ");

}

else

{

// Read the elements from keyboard

Console.WriteLine("Enter {0} elements", n);

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

{

arr[i] = Convert.ToInt32(Console.ReadLine());

}

first = second = 0;

for (i = 0; i < arr.Length; i++)

{

// If current element is smaller than

// first then update both first and second

if (arr[i] > first)

{

second = first;

first = arr[i];

}

// If arr[i] is in between first

// and second then update second

else if (arr[i] > second && arr[i] != first)

second = arr[i];

}

if (second == 0)

Console.Write("There is no second largest element\n");

else

Console.Write("The second largest element " + second);

}

Console.ReadKey();

}

}

}

// 8. WAP to reverse a given string using c#.

// 8. Program to reverse a given string

using System;

namespace stringReverse

{

class Program

{

static void Main(string[] args)

{

string str = "", reverse = "";

int Length = 0;

Console.WriteLine("Enter a Word");

//Getting String(word) from Console

str = Console.ReadLine();

//Calculate length of string str

Length = str.Length - 1;

while (Length >= 0)

{

reverse = reverse + str[Length];

Length--;

}

//Displaying the reverse word

Console.WriteLine("Reverse word is {0}", reverse);

Console.ReadLine();

}

}

}