using System;

using System.Xml.Serialization;

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programming I - Test 3 (Version R)

\* Name: Abdallah Fardin Divkar

\* Date: 19/4/2024

\* Student #: 301302441

\* Solution: VersionR.exe

\* Test 3 - 25% of final grade

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

namespace VersionR

{

class VerR

{

static void Main(string[] args)

{

char choice;

do

{

DisplayMenu();

choice = Convert.ToChar(Console.ReadLine());

switch (choice)

{

case 'a':

DemoQuestion3();

break;

case 'w':

DemoQuestion4();

break;

case 'p':

DemoQuestion5();

break;

case 't':

DemoQuestion6();

break;

default:

Console.WriteLine("Invalid choice : Please enter a valid letter");

break;

}

} while (choice != 'x');

}

#region Question 1 - 11 marks

/\*

\* Write a method that does not take any argument nor does it

\* return a value. The method only displays the following text.

\*

\* YOU MUST INSERT YOUR NAME IN THE TOP LINE OF THE MENU

\*

\* +---------------------COMP100------------------------+

\* | a) Working with an integer array |

\* | w) Weight Conversion |

\* | p) Planetary Motion |

\* | t) CRA Calculation |

\* | |

\* | x) To exit the program |

\* +----------------------------------------------------+

\* Press the letter corresponding to your choice-> ");

\*

\*/

static void DisplayMenu()

{

Console.WriteLine("+---------------------COMP100------------------------+");

Console.WriteLine("| a) Working with an integer array |");

Console.WriteLine("| w) Weight Conversion |");

Console.WriteLine("| p) Planetary Motion |");

Console.WriteLine("| t) CRA Calculation |");

Console.WriteLine("| |");

Console.WriteLine("| x) To exit the program |");

Console.WriteLine("+----------------------------------------------------+");

Console.Write("Press the number corresponding to your choice-> ");

}

#endregion

#region Question 2 - 20 marks

/\*

\* In your Main() method, using a suitable looping structure, write the

\* code to call this method repeatedly.

\* Using a suitable branching structure, write the code to implement the

\* following required functionality for all valid responses.

\* Valid responses includes both upper and lower case of the input. The

\* following must be implemented:

\* a will call the DemoQuestion3() method

\* v will call the DemoQuestion4() method

\* p will call the DemoQuestion5() method

\* t will call the DemoQuestion6() method

\* x will terminate the program

\* Any other key will produce an error message

\*

\* YOU DON'T HAVE TO CREATE A NEW METHOD FOR THIS QUESTION

\*/

#endregion

#region Question 3 - 36 marks

/\* Question 3c (10 marks)

\*

\* Write a method with the following specification

\* name : DisplaySmallInt

\* arguments: array of int

\* returns : nothing

\* display : only the items that are less than 15

\* task : use a suitable looping structure to send only the

\* appropriate items to the screen.

\*

\* YOU MUST USE A FOR loop

\*/

static void DisplaySmallInt(int[] num)

{

for(int i = 0; i < num.Length; i++)

{

if (num[i] < 15)

{

Console.Write($"{num[i]} ");

}

}

}

/\* Question 3b (8 marks)

\*

\* Write a method with the following specification

\* name : DisplayIntArray

\* arguments: array of int

\* returns : nothing

\* display : all the items in the array separated by a single space

\* task : use a suitable looping structure to send each

\* item to the screen.

\*

\* YOU MUST USE A FOREACH loop

\*/

static void DisplayIntArray(int[] num)

{

foreach(int i in num)

{

Console.Write($"{i} ");

}

}

/\* Question 3a (11 marks)

\*

\* Write a method with the following specification

\* name : GenerateIntArray

\* arguments: int representing the number of ints to generate

\* i.e. the size of the resulting array

\* returns : int array

\* display : nothing

\* task : 1) create the random object to generate random numbers

\* 2) declare and allocate sufficient storage for the array

\* 3) using a suitable looping structure get a random value

\* and assign it the element of the array

\* use random.Next(10, 31) to obtain a random int in the range 10 to 30

\* 4) return the array

\*/

static int[] GenerateIntArray(int arraySize)

{

Random random = new Random();

int[] num = new int[arraySize];

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

{

num[i] = random.Next(10,31);

}

return num;

}

/\* The driver for question 3 (7 marks)

\*

\* 1) Call GenerateIntArray with argument 35 and assign the

\* returned value to a suitable variable

\* 2) Use DisplayIntArray to print the above stored value

\* 3) Print an empty line

\* 4) Use DisplayOnlyEvenInt to print only the even items in the

\* above stored value

\*/

static void DemoQuestion3()

{

Console.WriteLine("\n\n\*\*\*\*\*\*\*\*\*\*Begin Question 3 \*\*\*\*\*\*\*\*\*\*");

//code for invoking question 3 goes here

int[] num = GenerateIntArray(35);

DisplayIntArray(num);

Console.WriteLine();

DisplaySmallInt(num);

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*End Question 3 \*\*\*\*\*\*\*\*\*\*\n\n");

}

#endregion

#region Question 4 - 19 marks

/\* Question 4 (12 marks)

\*

\* Write a method with the following specification

\* name : WeightConversion

\* arguments: an double representing the liter value to convert

\* an double to store the kg

\* an double to store the gram

\* an double to store the milligram

\* returns : nothing

\* display : nothing

\* task : converts the first argument (grams amount) into

\* kg, g and mg

\*

\* hint : the last three parameters are decorated so that the

\* method receives the actual variable so it is able to

\* modify the actual parameters it receives.

\*

\* : 1000 gram = 1 kg

\* 1 gram = 1000 mg

\*/

static void WeightConversion(double g, out double kiloGram, out double gram, out double milliGram)

{

kiloGram = g \* 1.0;

gram = kiloGram \* 1000.0;

milliGram = gram \* 1000.0;

}

/\* The driver for question 4 (7 marks)

\*

\* 1) Declare an int variable (this will be your first argument)

\* and initialized it to 1\_234.567

\* 2) Declare three other variables to store the kilograms, grams and milligrams

\* 3) Call the WeightConversion with the appropriate arguments.

\* 4) Print all of the arguments after the method call

\* 5) Now change the first argument to 12\_345.678 and repeat step 3 and 4.

\*/

static void DemoQuestion4()

{

Console.WriteLine("\n\n\*\*\*\*\*\*\*\*\*\*Begin Question 4 \*\*\*\*\*\*\*\*\*\*");

//code for invoking question 4 goes here

double liter = 1\_234.567;

double kiloGram, gram, milliGram;

WeightConversion(liter, out kiloGram, out gram, out milliGram);

Console.WriteLine($"{liter}g is equivalent to {kiloGram:F0}kg {gram:F0}g and {milliGram:F0}mg");

liter = 12\_345.678;

WeightConversion(liter, out kiloGram, out gram, out milliGram);

Console.WriteLine($"{liter}g is equivalent to {kiloGram}kg {gram}g and {milliGram}mg");

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*End Question 4 \*\*\*\*\*\*\*\*\*\*\n\n");

}

#endregion

#region Question 5 - 16 marks

/\* Question 5 (10 marks)

\*

\* Kepler found that there is a relationship between the distance

\* of a planet from the Sun and their orbital period.

\* See attached word document for the formulae

\*

\* Write a method that calculates and returns the orbital period

\* of a planet.

\* The specifications is as follows:

\*

\* Name:

\* CalculateOrbitalPeriod

\*

\* Argument:

\* a double representing the distance of the planet from

\* the Sun.

\*

\* Returns:

\* a double indicating the orbital period.

\*

\* Action:

\* Calculates and return the orbital period. The equation is

\* given in the attached word document.

\*

\* Displays:

\* Nothing.

\*

\*/

static double CalculateOrbitalPeriod(double distance)

{

const double k = 1.334 \* (10 \* 10 \* 10 \* 10 \* 10);

double a = (distance \* distance \* distance);

double orbitalPeriod = Math.Sqrt(k \* a);

return orbitalPeriod;

}

/\* The driver for question 5 (6 mark)

\*

\* In the DemoQuestion5() method, write the code to call this method three

\* times with arguments 1, 0.723 and 0.387 respectively and display the

\* return value after each call

\*

\*/

static void DemoQuestion5()

{

Console.WriteLine("\n\n\*\*\*\*\*\*\*\*\*\*Begin Question 5 \*\*\*\*\*\*\*\*\*\*");

//code for invoking question 5 goes here

double distance = 1;

Console.WriteLine($"A planet of distance {distance:F3}AU will have a year of {CalculateOrbitalPeriod(1):F1} days");

distance = 0.723;

Console.WriteLine($"A planet of distance {distance:F3}AU will have a year of {CalculateOrbitalPeriod(0.723):F1} days");

distance = 0.387;

Console.WriteLine($"A planet of distance {distance:F3}AU will have a year of {CalculateOrbitalPeriod(0.387):F1} days");

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*End Question 5 \*\*\*\*\*\*\*\*\*\*\n\n");

}

#endregion

#region Question 6 - 22 marks

/\* Question 6 (14 marks)

\*

\* All Canadian are required to pay federal and provincial taxes. This question deals

\* with federal tax calculation. The question specifications are as follows:

\*

\* Name:

\* CalculateFederalTax

\*

\* Argument:

\* a double representing the annual taxable income

\*

\* Returns:

\* a double indicating the federal tax owned to the government

\*

\* Action:

\* Calculates and return the tax based on the following abridge table:

\* 15% on the first $45,282 of taxable income, +

\* 20.5% on the next $45,281 of taxable income (on the portion of taxable income over $45,282 up to $90,563), +

\* 33% of taxable income over $90,563.

\*

\* Displays: nothing.

\*

\*/

static double CalculateFederalTax(double annualIncome)

{

double federalTax;

if(annualIncome <= 45282)

{

federalTax = (0.15 \* annualIncome);

}

else if(annualIncome >= 90563)

{

federalTax = 45282 \* 0.15 + 45281 \* 0.205 + (annualIncome - 90563) \* 0.33;

}

else

{

federalTax = 45282 \* 0.15 + (annualIncome - 45282) \* 0.205;

}

return federalTax;

}

/\* The driver for question 6 (8 marks)

\*

\* 1) Declare a double variable and initialized it to 30,000

\* 2) Invoke the CalculateFederalTax method with this variable and print the returned value

\* 3) Change the variable to 80,000 and repeat step 2

\* 4) Change the variable to 130,000 and repeat step 2

\*/

static void DemoQuestion6()

{

Console.WriteLine("\n\n\*\*\*\*\*\*\*\*\*\*Begin Question 6 \*\*\*\*\*\*\*\*\*\*");

//code for invoking question 6 goes here

double annualIncome = 30000;

Console.WriteLine($" {annualIncome:C} will result in {CalculateFederalTax(annualIncome):C} of tax");

annualIncome = 80000;

Console.WriteLine($" {annualIncome:C} will result in {CalculateFederalTax(annualIncome):C} of tax");

annualIncome = 130000;

Console.WriteLine($" {annualIncome:C} will result in {CalculateFederalTax(annualIncome):C} of tax");

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*End Question 6 \*\*\*\*\*\*\*\*\*\*\n\n");

}

#endregion

}

}