Implementation

For

Perfect Number && Great Common Divisor

Supervised By

ENG: Mahmoud Abd El Salam DR: Samir El Mougy

Teaching Assistant Professor of computer science

Faculty of computer and information science

Mansoura University

Submitted By

Mahmoud El Gohary Nabil Mohamed

ID: 805647069 ID: 805486489

Major: Software Engineering

Faculty of computer and information science

Mansoura University

***/\****

***PROGRAMMERS  : Mahmoud El Gohary && Nabil Salah***

***DATE CREATED: 21/12/2021***

***PURPOSE :***

***GCD:***

***to find the great common divisor***

***the largest intger d that***

***d|a && d|b***

***Used to Simplify Fraction Numbers***

***Perfect Number:***

***To find the perfect numbers from 2 to 1000***

***the perfect number is a positive integer that the summation of its positive factors***

***excluding the number itself***

***is equal to the number itself***

***VERY IMPORTANT:***

***using the definition of the perfect number we will find no odd perfect numbers***

***The reason we didn't implement that in our code is to keep the code as simple as possible***

***we didn't want to use classes or arrays and useless functions as we can***

***as shown the perfect number program shows the factors and the summation of the factor and display them to the user***

***and if we try to make a condition for an odd number it will be a must to repeat the code again so we didn't make that***

***\*/***

using System;

namespace College1

{

    internal class Program

    {

***/\*The Euclidean Algorithm for finding GCD(A,B) is as follows:***

***If A = 0 then GCD(A,B)=B, since the GCD(0,B)=B, and we can stop.***

***If B = 0 then GCD(A,B)=A, since the GCD(A,0)=A, and we can stop.***

***Write A in quotient remainder form (A = B⋅Q + R)***

***Find GCD(B,R) using the Euclidean Algorithm since GCD(A,B) = GCD(B,R)\*/***

        public static int Gcd(int a, int b)

        {

***//initialize temp***

            int temp = 0;

            while (a != 0 && b != 0)

            {

***//Make a Temporary copy of a***

                temp = a;

                a = b; ***//assign a to b to put the new greatest number in a again***

                if (temp % b == 0) ***//to prevent b from modification in case its the divisor***

                    break;

                b = temp % b; ***//new value to try as a divisor***

            }

            return b;

        }

***// the recursive Method of gcd finding***

***/\*The Euclidean Algorithm for finding GCD(A,B) is as follows:***

***If A = 0 then GCD(A,B)=B, since the GCD(0,B)=B, and we can stop.***

***If B = 0 then GCD(A,B)=A, since the GCD(A,0)=A, and we can stop.***

***Write A in quotient remainder form (A = B⋅Q + R)***

***Find GCD(B,R) using the Euclidean Algorithm since GCD(A,B) = GCD(B,R)\*/***

        public static int gCDRe(int a, int b)

        {

***//my base case that involves finding b = 0 after last reminder operation***

            if (b == 0)

            {

                return a; ***//returning my gcd***

            }

            else

            {

***//trying new value as a divisor***

                return gCDRe(b, a % b);

            }

        }

        public static void GDRProgram()

        {

            Console.WriteLine("Enter Two Numbers Two Get Their GCD");

            int in1 = 0;

            int in2 = 0;

            bool isValid = false; ***//become true when you enter valid int***

            bool isValid2 = false; ***//become true when you enter valid int***

***//to confirm that the input is an integer***

            do

            {

                Console.Write("The first Number : ");

***//return true in case the input is int and assign it to in1***

                if (int.TryParse(Console.ReadLine(), out in1))

                {

                    if (in1 != 0)

                        isValid = true;

                    else

                    {

                        Console.WriteLine("Input should be non-zero integer");

                        continue;

                    }

                }

                else

                {

                    Console.WriteLine("not a valid input -- Please enter an intger");

                    continue;

                }

                Console.Write("The Second Number: ");

***//return true in case the input is int and assign it to in2***

                if (int.TryParse(Console.ReadLine(), out in2))

                {

                    if (in2 != 0)

                        isValid2 = true;

                    else

                    {

                        Console.WriteLine("Input should be non-zero integer");

                        continue;

                    }

                }

                else

                {

                    Console.WriteLine("not a valid input -- Please enter an intger");

                    continue;

                }

            } while (!(isValid && isValid2));

            int ans = Gcd(in2, in1);

***//running my method and return ans to my console in iterative way***

            Console.WriteLine("Their GCD (by iteration) Is: " + ans);

***//running my method and return ans to my console in recursive***

            Console.WriteLine("Their GCD (by recursion) Is: " + gCDRe(in1, in2));

            if (ans == 1)

                Console.WriteLine("Look They Are Relatively Prime");

        }

***/\*The PerfectNumbCheckProgram method is created to prompt the user to enter an integer from 2 to 1000***

***as defined in the problem and check if that number is perfect or not***

***by using the perfect number definition (The sum of the number factor must == the number itself)\*/***

        public static void PerfectNumbCheckProgram()

        {

            while (true)

            {

***// Prompt user for data***

Console.Write("Please enter an inger betwenn 2 and 1000 to check if it is perfect ot not: ");

***// Declaraing a local variable and read the data given by the user to check if it is valid or not***

                var checkValidation = Console.ReadLine();

***// Declaraing a variable as an intger to pass the user data to it if it is valid***

                int number;

***// check if the input is an intger to proceed***

                bool parseSuccess = int.TryParse(checkValidation, out number);

***// using the logical operator and between the expression because we want all expression to be true***

                if (parseSuccess && number >= 2 && number <= 1000)

                {

                    int sumOfFactors = 0;

***// Display the message outside the loop so it would be executed once so, the output would me more friendly***

                    Console.Write("[+] {0} Factors are:", number);

***/\*Caution We divided number by two because a perfect number is a number***

***that is half the sum of all of its positive divisors including itself\*/***

                    for (int possiableFactors = 1; possiableFactors <= number / 2; possiableFactors++)

                    {

***/\* using the factor definition of a factor to check for all real factors as if we divide the number by***

***its factor there will be no reminder \****/

                        if (number % possiableFactors == 0)

                        {

***// displaying the factors to screen***

                            Console.Write(" {0} ", possiableFactors);

***// adding all factors and storing them int the sumOfFactor variable***

                            sumOfFactors += possiableFactors;

                        }

                    }

***// displaying a new line so the output will be more friendly***

                    Console.WriteLine();

***// displaying the summation of the factors***

                    Console.WriteLine("[+] Sum of its factors is {0}", sumOfFactors);

***/\* using the definition of the perfect number by checking if the summation of the factors is equal to the***

***Number \*/***

                    if (sumOfFactors == number)

                    {

***// if the number is perfect this message will be displayed to the user***

                        Console.WriteLine("[+] so {0} is a perfect number\n", number);

***// the program will get out of the while loop and stop***

                        break;

                    }

                    else

                    {

***// If the number is not perfect this message will be displayed to the user***

                        Console.WriteLine("[-] This number is not perfect.\n");

***// the program will get out of the while loop and stop***

                        break;

                    }

                }

                else

                {

***/\* these message will be shown to the user if he entered a not valid input so it will help him to***

***enter a valid input \****/

                    Console.WriteLine("that is not a valid input");

                    Console.WriteLine("HINT for a valid input");

                    Console.WriteLine("    [0] Input must be an intger.");

                    Console.WriteLine("    [1] The intger must be between 2 and 1000.");

                    Console.WriteLine(".....PLEASE TRY AGIN.....");

                }

            }

        }

***/\*The PerfectNumProgram method is created to display all perfect number for 2 to 1000\*/***

        public static void PerfectNumProgram()

        {

***// looping to get all numbers in the given range***

            for (int Number = 2; Number <= 1000; Number++)

            {

***// Declaraing a variable to store the sum of factors and resting it in each loop for the outer loop***

                int sum = 0;

***// getting the all possible factors to each number in the outer loop***

***/\*Caution We divided number by two because a perfect number is a number***

***that is half the sum of all of its positive divisors including itself\*/***

                for (int possibleFactor = 1; possibleFactor <= Number / 2; possibleFactor++)

                {

***// searching for real factors***

                    if (Number % possibleFactor == 0)

                    {

***// adding all real factors together***

                        sum += possibleFactor;

                    }

                }

***// checking if the number is perfect or not by using the definition on the perfect number***

                if (Number == sum)

                {

***// Displaying the perfect numbers to the user***

                    Console.Write("[+] {0} is a perfect Number \n", Number);

***// Diaplaying the factors to screen***

                    Console.Write("[+] Factors: ");

***// Display the factors to show the user***

                    for (int factor = 1; factor < Number; factor++)

                    {

***// checking for real factors***

                        if (Number % factor == 0)

                        {

***// printing the real factors***

                            Console.Write("{0} ", factor);

                        }

                    }

***// proving that the number is perfect to user by printing the sum of the factors and the number***

                    Console.WriteLine();

                    Console.WriteLine("[+] Sum of its factors {0} = Number {1}", sum, Number);

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

                }

            }

        }

        static void Main(string[] args)

        {

            do

            {

                Console.WriteLine("Choose Operation From 1-3");

                Console.WriteLine("[1] Printing Perfect Numbers From 2 - 1000");

                Console.WriteLine("[2] Test If A Number Is Perfect Or Not");

                Console.WriteLine("[3] Find Greatest Common Divisor (GCD) Of Two Numbers");

                bool isValid = false;

                int op;

                do

                {

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

                    if (int.TryParse(Console.ReadLine(), out op))

                        if (op > 0 && op <= 3)

                            isValid = true;

                        else

                            Console.WriteLine("please enter an intger between 1 and 3");

                } while (!isValid);

                switch (op)

                {

                    case 1:

                        Console.WriteLine();

                        Console.WriteLine("Executing the programe [Printing Perfect Numbers From 2 -1000]\n");

                        PerfectNumProgram();

                        break;

                    case 2:

                        Console.WriteLine();

                        Console.WriteLine("Executing the programe [Test If A Number Is Perfect Or Not]\n");

                        PerfectNumbCheckProgram();

                        break;

                    case 3:

                        Console.WriteLine();

                        Console.WriteLine("Executing the programe [Find Greatest Common Divisor (GCD) Of Two Numbers]\n");

                        GDRProgram();

                        break;

                }

                Console.Write("Do You Want To Try Again (y to tyr agin || any other character to exit) ");

                string pass = Console.ReadLine();

                if (!(pass == "y" || pass == "Y"))

                    break;

            } while (true);

            Console.WriteLine(".....Have A Good Day.....");

        }

    }

}