C# Chapter Three

1. Write an expression that checks whether an integer is **odd or even**.

using System;

namespace oddOreven

{

class Program

{

static void Main(string[] args)

{

int num1 = 24;

bool odd = num1 % 2 == 0;

Console.WriteLine(odd);

}

}

}



1. Write a Boolean expression that checks whether a given integer is **divisible by both 5 and 7**, without a remainder.

using System;

namespace checkingDevision

{

class Program

{

static void Main(string[] args)

{

int num1 = 37;

bool check = num1 % 7 == 0;

bool check2 = num1 % 5 == 0;

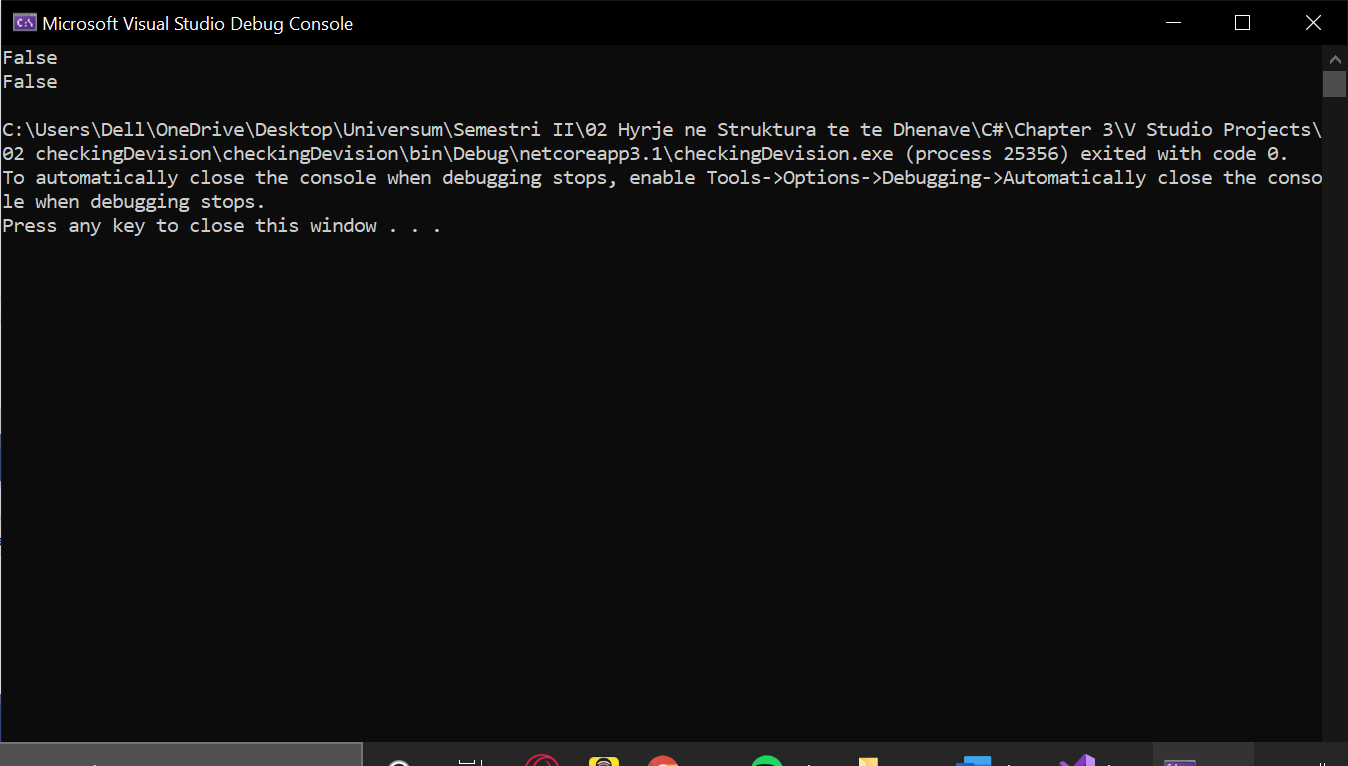
Console.WriteLine(check);

Console.WriteLine(check2);

}

}

}



1. Write an expression that checks for a given integer if its **third digit** (right to left) is 7.

using System;

namespace isitEqual

{

class Program

{

static void Main(string[] args)

{

Console.Write("Write a 3 diget number to check if the 3rd digit is 7: ");

int number = Convert.ToInt32(Console.ReadLine());

int devider = (number / 100);

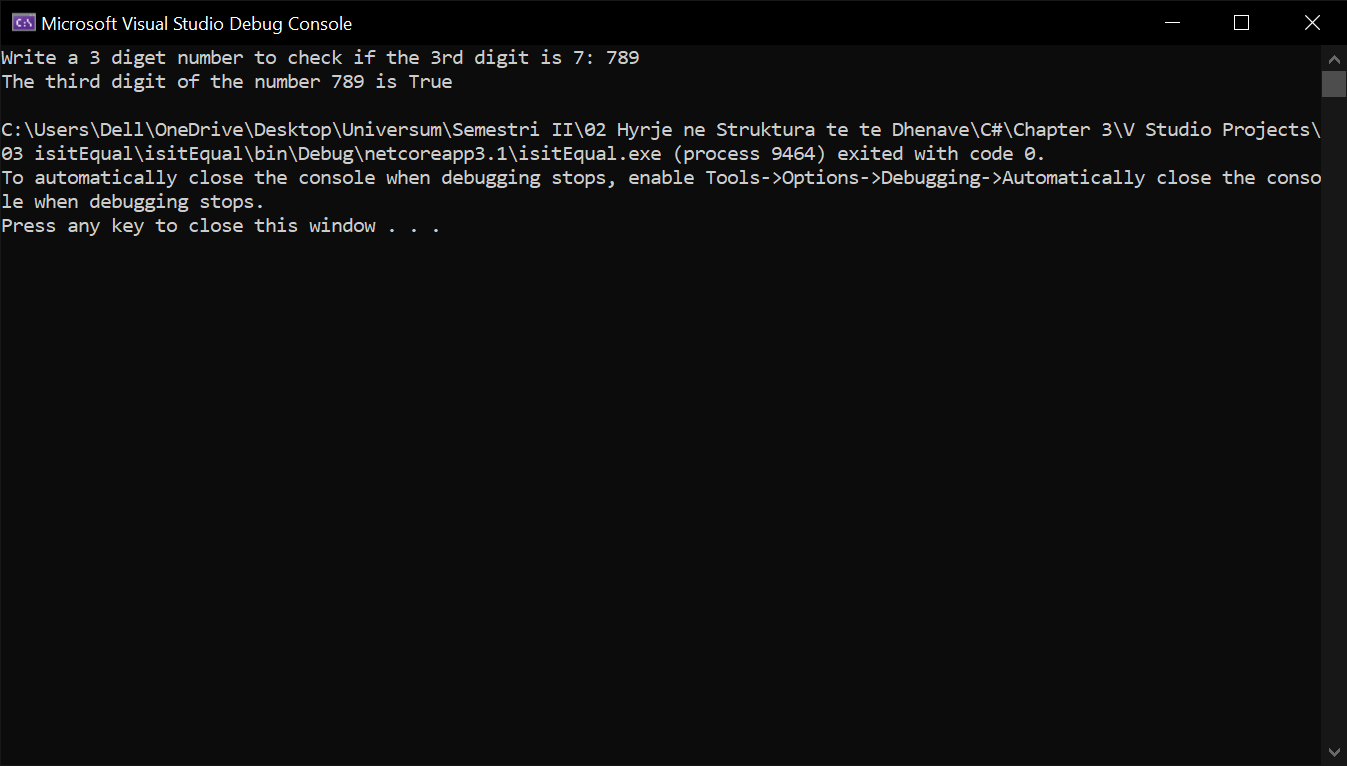
bool checking = devider == 7;

Console.WriteLine("The third digit of the number {0} is {1} ", number, checking);

}

}

}



1. Write an expression that checks whether the **third bit** in a given integer is 1 or 0.

using NetTools;

using System;

namespace checkingBits

{

class Program

{

static void Main(string[] args)

{

Console.Write("Write a number to check the Third bit if it is 1 or 0 ");

int number = Convert.ToInt32(Console.ReadLine());

bool bits;

if (bits = ((number >> 2)& 1) == 1) {

Console.WriteLine("The 3rd bit is 1");

}

else

{

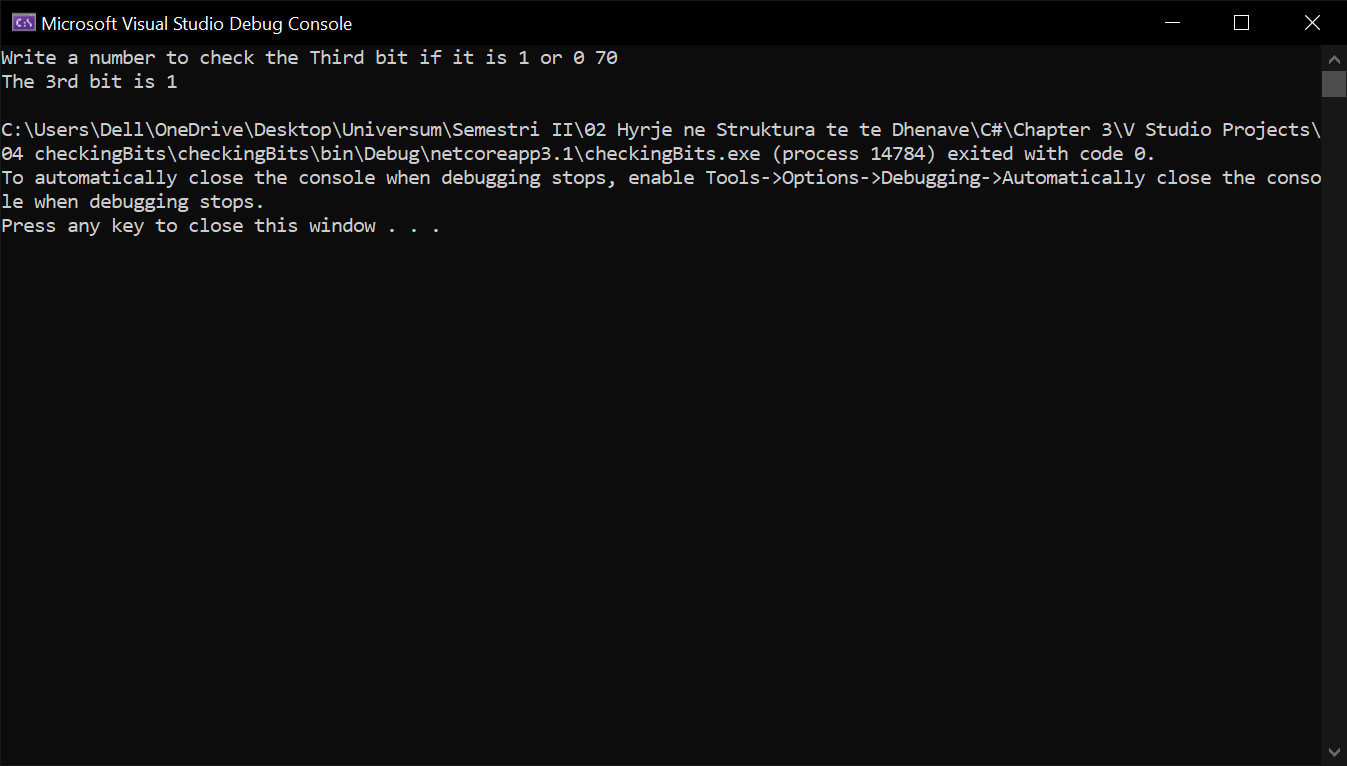
Console.WriteLine("The 3rd bit is 0");

}

}

}

}



1. Write an expression that calculates the **area of a trapezoid** by given sides **a**, **b** and height **h**.

using System;

namespace trapezoidSurface

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Shkruaj informatat e Trapezit tuaj:");

Console.Write("Shkruaj Brinjen a: ");

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

Console.Write("Shkruaj Brinjen b: ");

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

Console.Write("Shkruaj Brinjen h: ");

int h = Convert.ToInt32(Console.ReadLine());

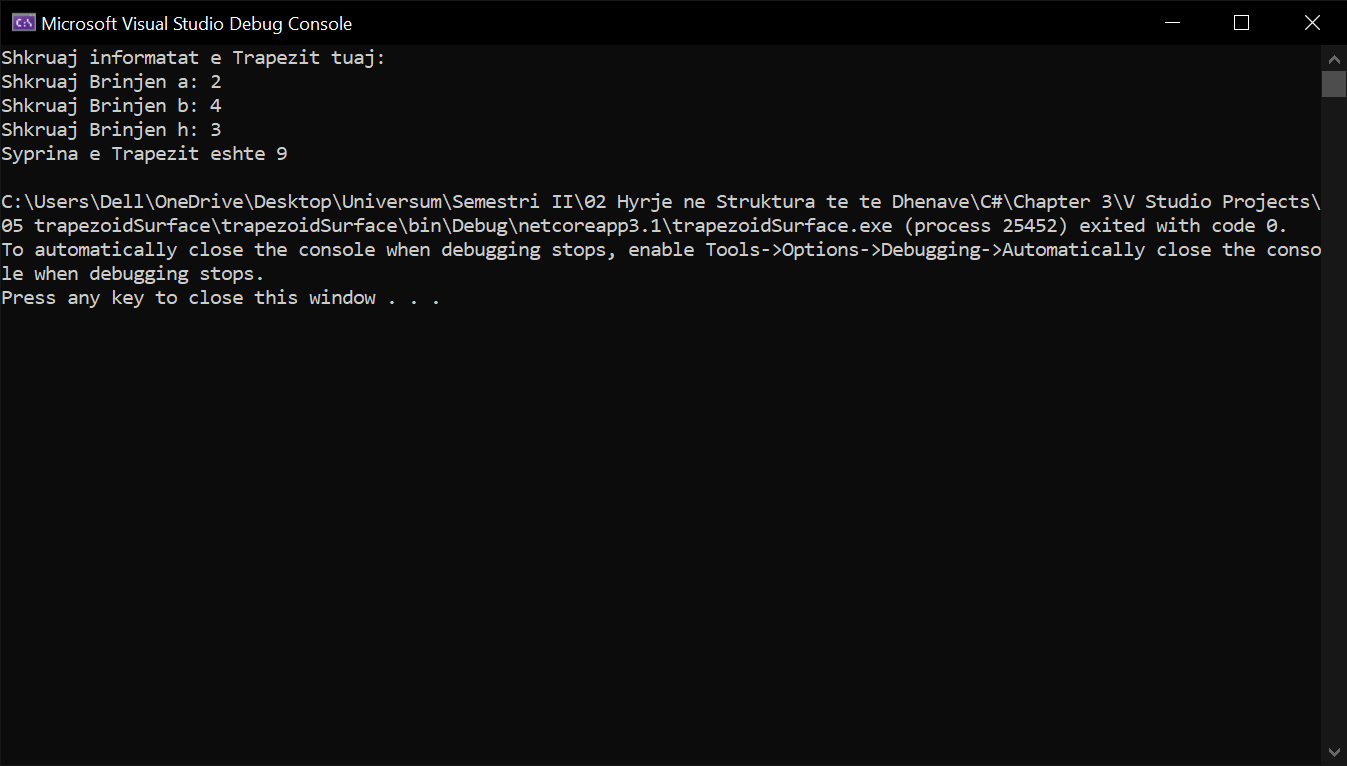
int formula = (a + b) \* h / 2;

Console.WriteLine("Syprina e Trapezit eshte {0}",formula);

}

}

}



1. Write a program that prints on the console the **perimeter and the area of a rectangle**by given side and height entered by the user.

using System;

namespace racktangelPandS

{

class Program

{

static void Main(string[] args)

{

int a = 20;

Console.Write("Shkruaj Brinjen b; ");

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

int perimetri = (2 \* a) + (2 \* b);

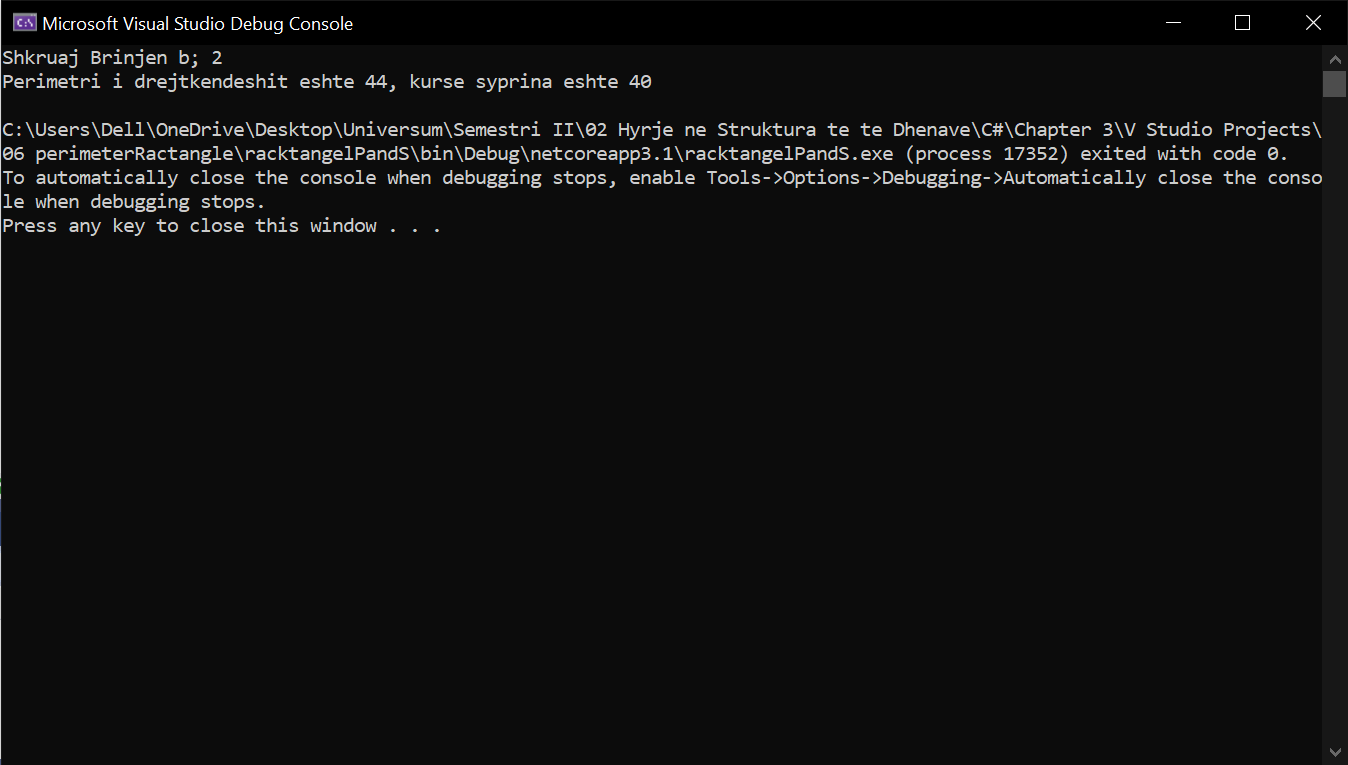
int syprina = a \* b;

Console.WriteLine("Perimetri i drejtkendeshit eshte {0}, kurse syprina eshte {1}", perimetri,syprina);

}

}

}



1. The gravitational field of the Moon is approximately 17% of that on the Earth. Write a program that calculates the **weight of a man on the moon** by a given weight on the Earth.

using System;

namespace calculatingWeight

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter your weight: ");

int weight = Convert.ToInt32(Console.ReadLine());

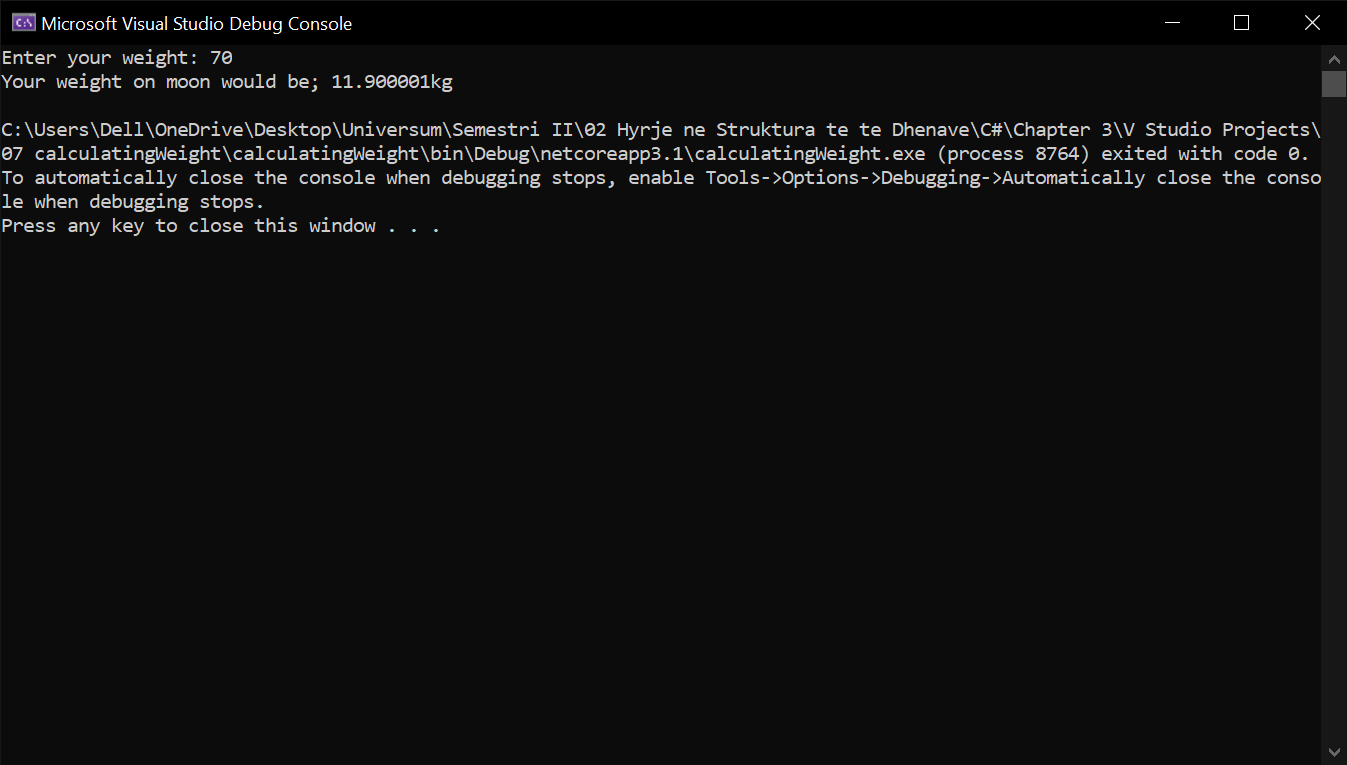
float formula = weight \* 0.17f;

Console.WriteLine("Your weight on moon would be; {0}kg", formula);

}

}

}



1. Write an expression that checks for a given point {x, y} if it is **within the circle**K[{0, 0}, R=5]. Explanation: the point {0, 0} is the center of the circle and 5 is the radius.

using System;

namespace calculatingCirclepoints

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Shkruaj x: ");

int x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Shkruaj y: ");

int y = Convert.ToInt32(Console.ReadLine());

bool formula;

if (formula = (x \* x) + (y \* y) <= 5)

{

Console.WriteLine("For an expression that checks for given point {x, y} it is with" +

"in the circle");

}

else

{

Console.WriteLine("For an expression that checks for given point {x, y} it is not with" +

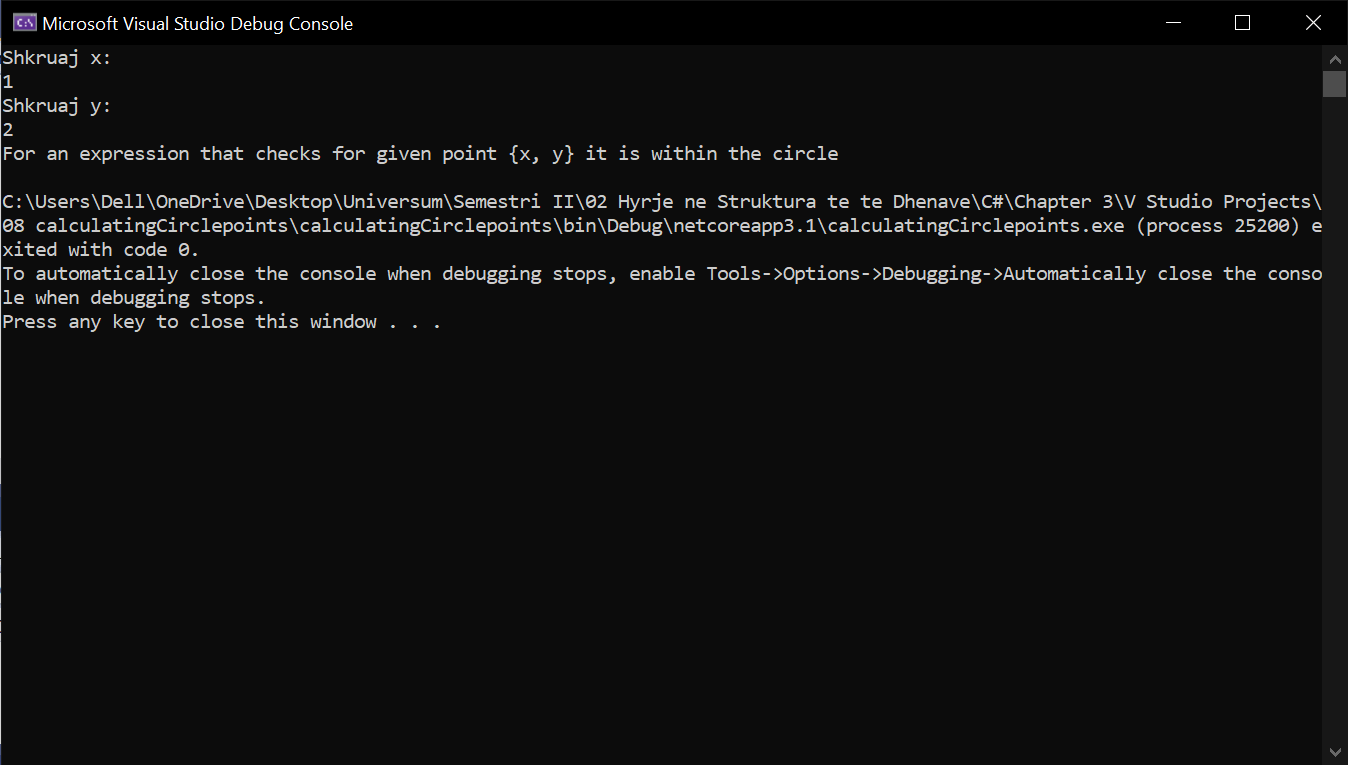
"in the circle");

}

}

}

}



1. Write an expression that checks for given point {x, y} if it is **within the circle** K[{0, 0}, R=5] and **out of the rectangle** [{-1, 1}, {5, 5}]. Clarification: for the rectangle the lower left and the upper right corners are given.

using System;

namespace \_09\_isinsideRecktangle

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter x: ");

int x = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter y: ");

int y = Convert.ToInt32(Console.ReadLine());

bool isInsideCircle = (x \* x + y \* y <= 5) ? true : false;

bool isOutsideRectangle = (x < -1 && x > 5 && y < 1 && y > 5) ? true : false;

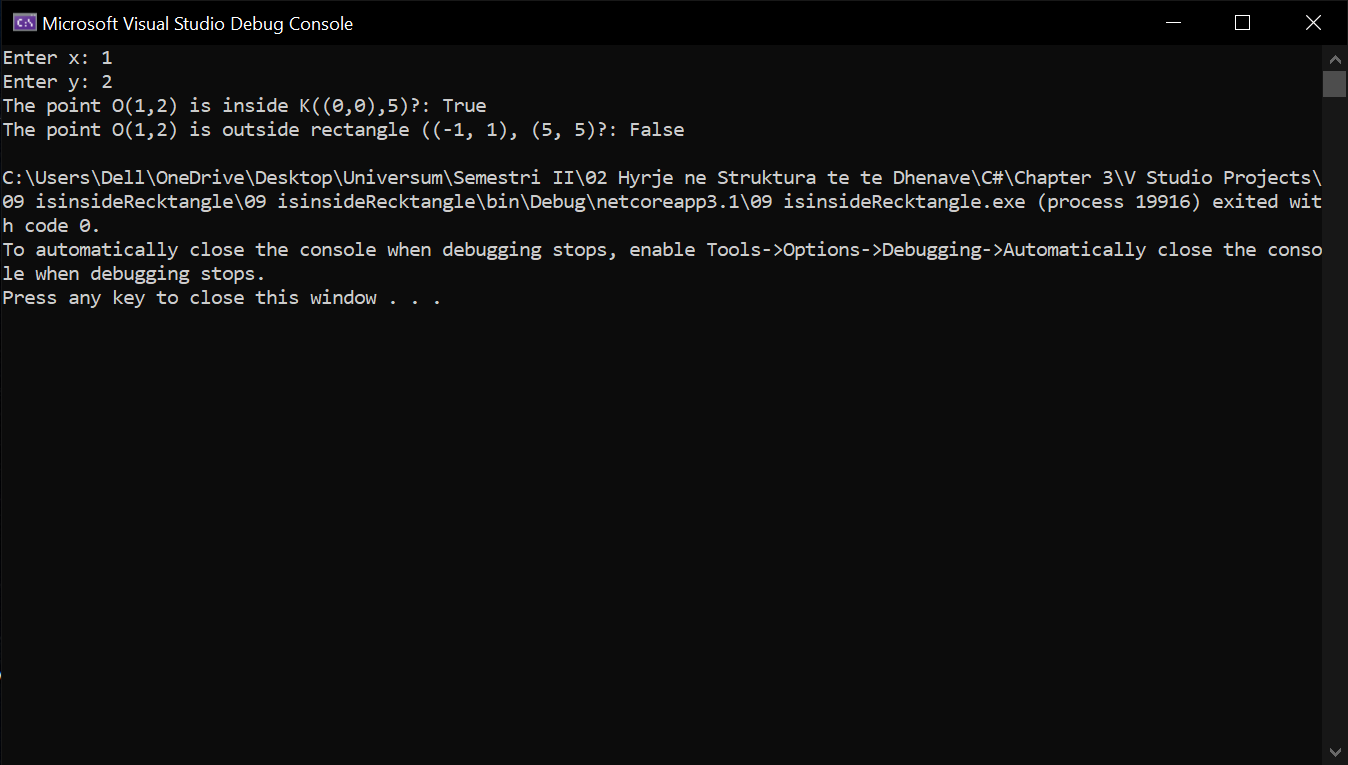
Console.WriteLine("The point O({0},{1}) is inside K((0,0),5)?: {2}", x, y, isInsideCircle);

Console.WriteLine("The point O({0},{1}) is outside rectangle ((-1, 1), (5, 5)?: {2}", x, y, isOutsideRectangle);

}

}

}



1. Write a program that takes as input a **four-digit number** in format **abcd** (e.g. 2011) and performs the following actions:
   * Calculates the sum of the digits (in our example 2+0+1+1 = 4).
   * Prints on the console the number in reversed order: **dcba** (in our example 1102).
   * Puts the last digit in the first position: **dabc** (in our example 1201).
   * Exchanges the second and the third digits: **acbd** (in our example 2101).

using System;

namespace \_10\_switchingPositions

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter number: ");

int number = Convert.ToInt32(Console.ReadLine());

int a = number / 1000;

int b = (number / 100) % 10;

int c = (number / 10) % 10;

int d = number % 10;

Console.WriteLine("1.Sum of digits = {0}", a + b + c + d);

Console.WriteLine("2.Digits backwards = {3}{2}{1}{0}", a, b, c, d);

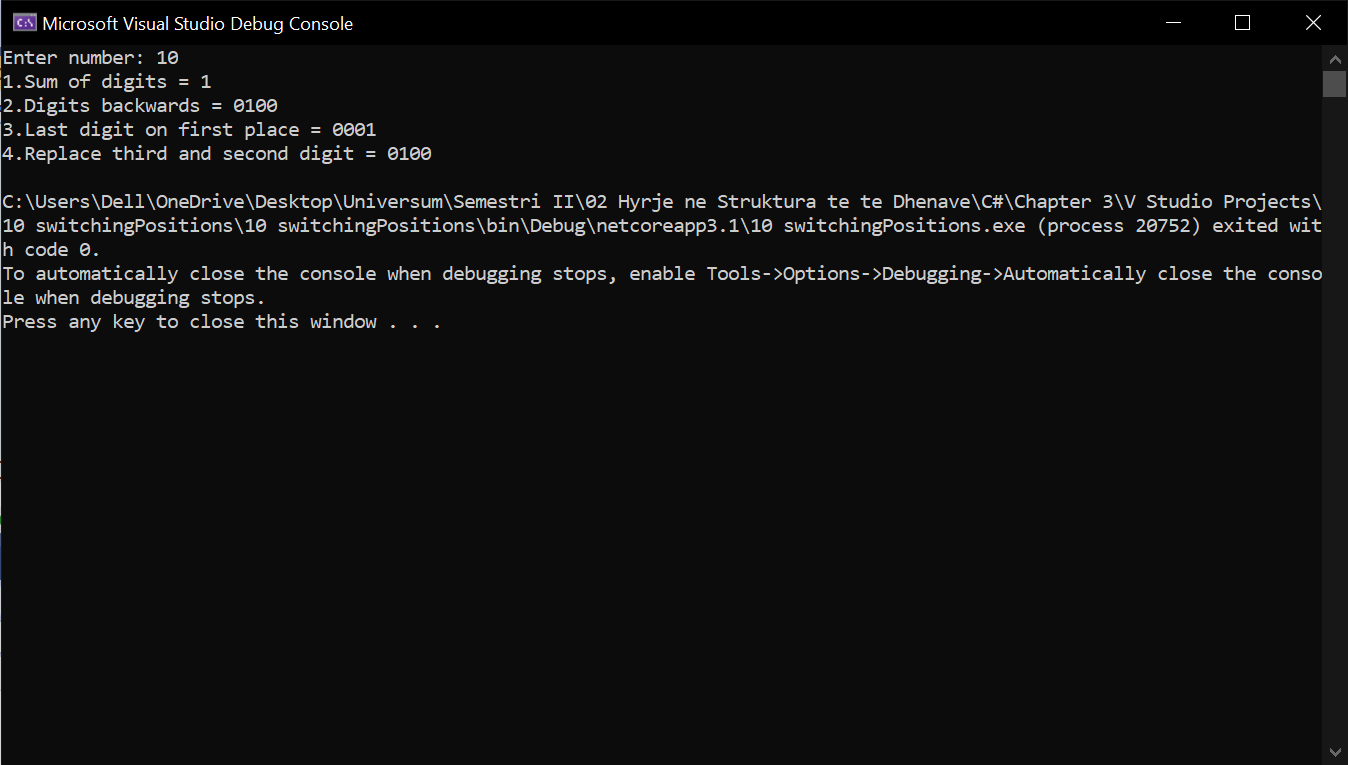
Console.WriteLine("3.Last digit on first place = {3}{0}{1}{2}", a, b, c, d);

Console.WriteLine("4.Replace third and second digit = {0}{2}{1}{3}", a, b, c, d);

}

}

}



1. We are given number **n** and position **p**. Write a sequence of operations that prints the value of **the bit on the position** **p** in the number (0 or 1). Example: **n**=35, **p**=5 -> 1. Another example: n=35, **p**=6 -> 0.

using System;

namespace \_11\_bitWise

{

class Program

{

static void Main(string[] args)

{

int n = 25;

int p = 3;

int i = 1;

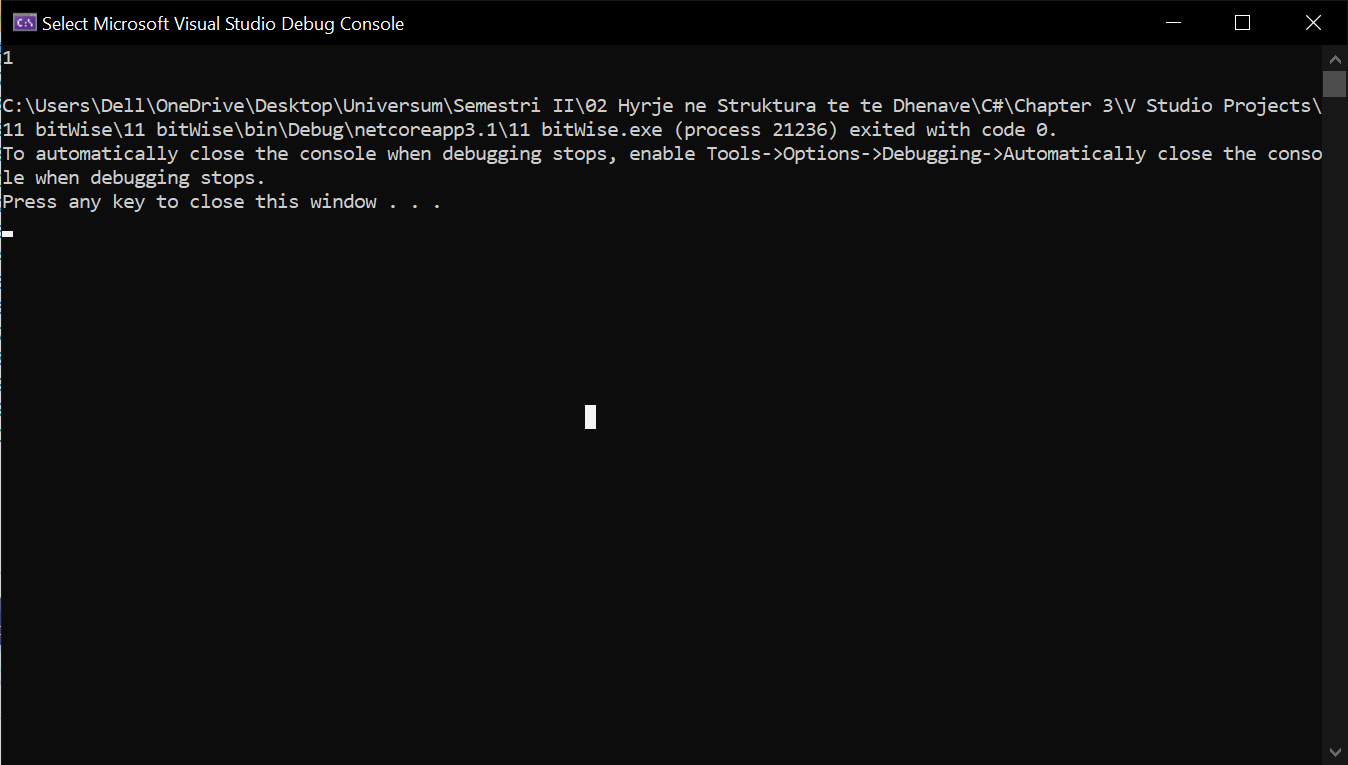
int mask = i << p;

Console.WriteLine((n & mask) != 0 ? 1 : 0);

}

}

}



1. Write a Boolean expression that checks if the bit on position **p** in the integer **v** has the value 1. Example v=5, **p**=1 -> **false**.

using System;

namespace \_12\_bitBool

{

class Program

{

static void Main(string[] args)

{

int v = 170;

int p = 70;

int mask = 1 << p;

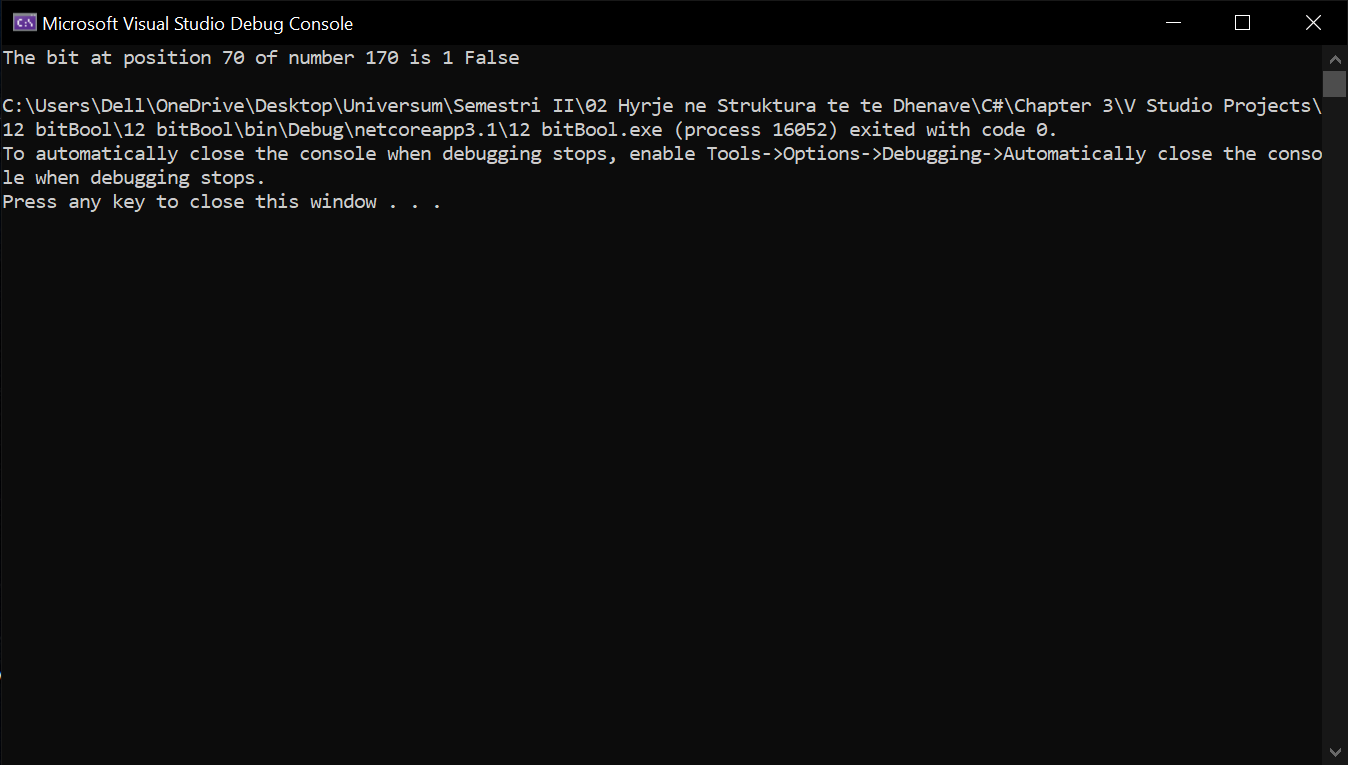
bool isOne = (v & mask) != 0;

Console.WriteLine("The bit at position {0} of number {1} is 1 {2}", p, v, isOne);

}

}

}



1. We are given the number **n**, the value **v** (**v** = 0 or 1) and the position **p**. write a sequence of operations that changes the value of **n**, so the bit on the position **p** has the value of **v**. Example: n=35, p=5, v=0 -> n=3. Another example: n=35, p=2, v=1 -> n=39.

using System;

namespace \_13\_changeNvalue

{

class Program

{

static void Main(string[] args)

{

int n = 350;

int v = 0;

int p = 3;

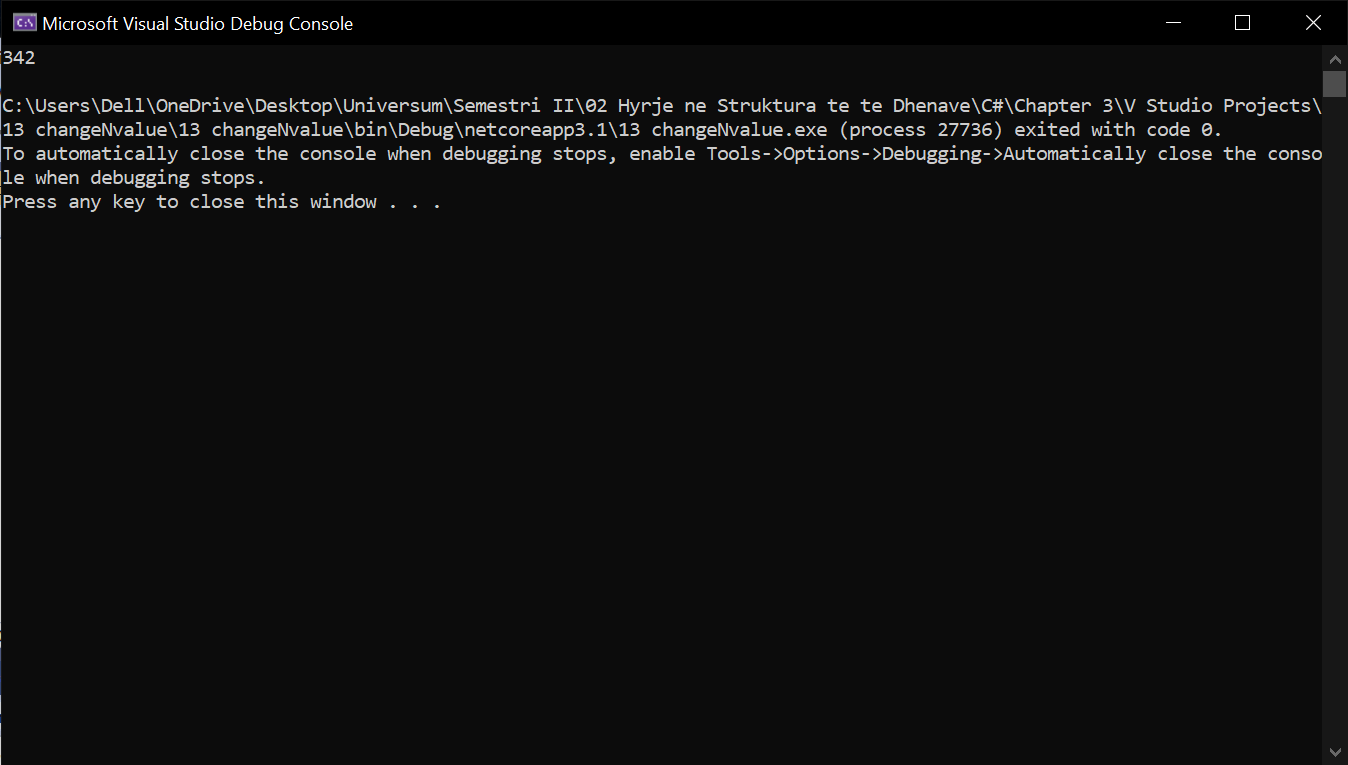
n = (v == 0) ? n = n & (~(1 << p)) : n = n | (1 << p);

Console.WriteLine(n);

}

}

}



1. Write a program that checks if a given number **n** (1 <**n** < 100) is a **prime number** (i.e. it is divisible without remainder only to itself and 1).

using System;

namespace \_14\_writenNumberPrime

{

class Program

{

static void Main(string[] args)

{

int number = 72;

bool isPrime = true;

if (number > 2)

for (int i = 2; i <= Math.Ceiling(Math.Sqrt(number)); ++i)

{

if (number % i == 0) isPrime = false;

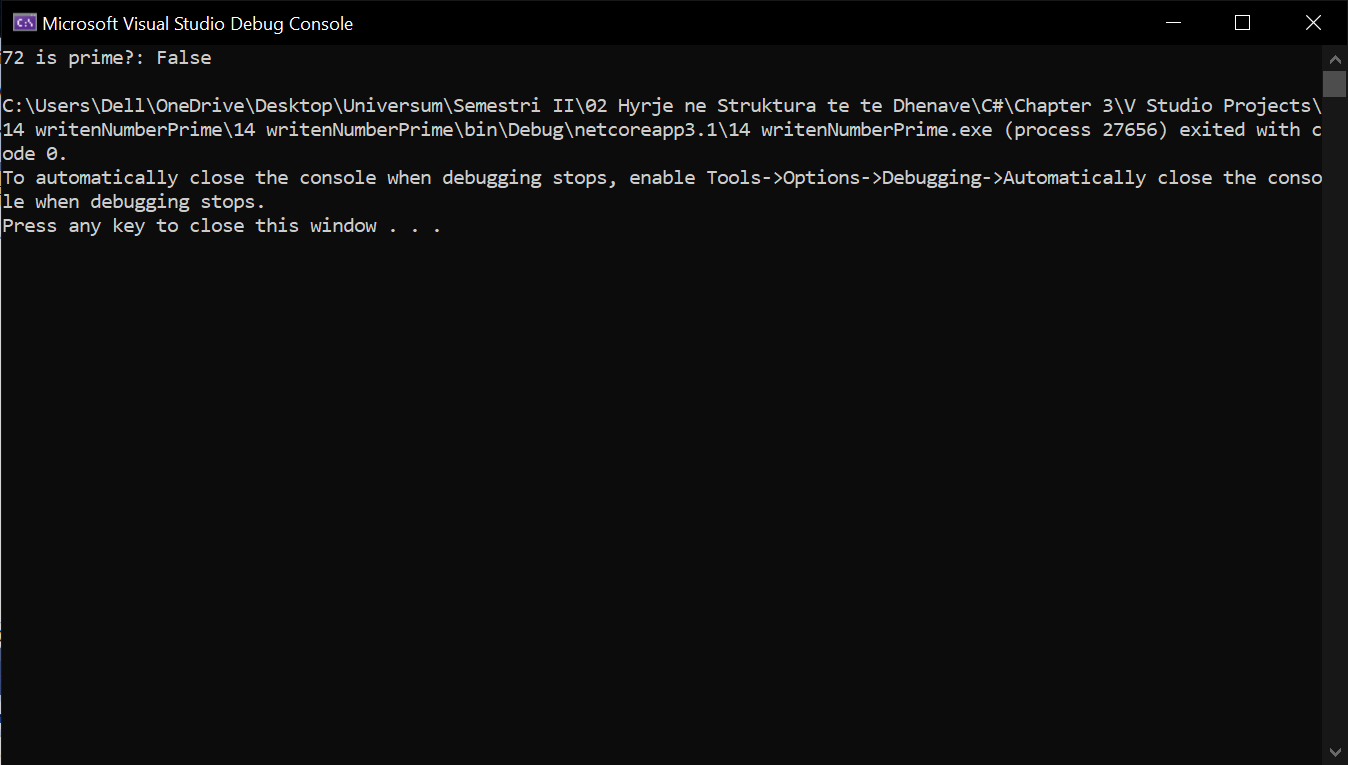
}

Console.WriteLine("{0} is prime?: {1}", number, isPrime);

}

}

}



1. Write a program that **exchanges the values of the bits** on positions 3, 4 and 5 with bits on positions 24, 25 and 26 of a given 32-bit unsigned integer.

using System;

namespace \_15\_changingBITPoistion

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter number: ");

int v = Convert.ToInt32(Console.ReadLine());

int mask = 1 << 3;

int bitAt3 = (v & mask) != 0 ? 1 : 0;

mask = 1 << 4;

int bitAt4 = (v & mask) != 0 ? 1 : 0;

mask = 1 << 5;

int bitAt5 = (v & mask) != 0 ? 1 : 0;

mask = 1 << 24;

int bitAt24 = (v & mask) != 0 ? 1 : 0;

mask = 1 << 25;

int bitAt25 = (v & mask) != 0 ? 1 : 0;

mask = 1 << 26;

int bitAt26 = (v & mask) != 0 ? 1 : 0;

v = (bitAt3 == 0) ? v = v & (~(1 << 24)) : v = v | (1 << 24);

v = (bitAt4 == 0) ? v = v & (~(1 << 25)) : v = v | (1 << 25);

v = (bitAt5 == 0) ? v = v & (~(1 << 26)) : v = v | (1 << 26);

v = (bitAt24 == 0) ? v = v & (~(1 << 3)) : v = v | (1 << 3);

v = (bitAt25 == 0) ? v = v & (~(1 << 4)) : v = v | (1 << 4);

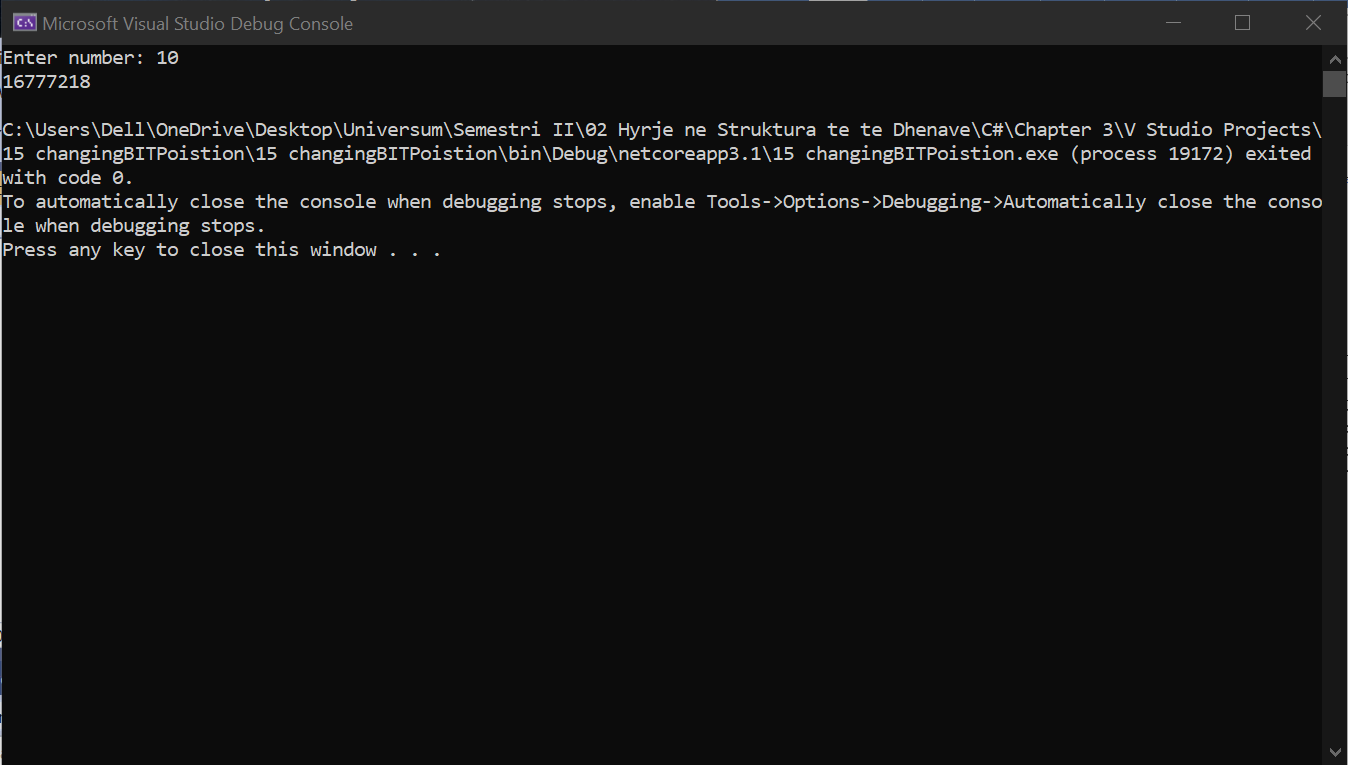
v = (bitAt26 == 0) ? v = v & (~(1 << 5)) : v = v | (1 << 5);

Console.WriteLine(v);

}

}

}



1. Write a program that **exchanges bits** {p, p+1, …, p+k-1} with bits {q, q+1, …, q+k-1} of a given 32-bit unsigned integer.

using System;

namespace \_16\_exchaningBits

{

class Program

{

private static uint ModifyNumber(uint number, int p, int q, int k)

{

int[] pBits = new int[k];

int[] qBits = new int[k];

for (int position = p, i = 0; i < pBits.Length; position++, i++)

{

pBits[i] = PthBit(number, position);

}

for (int position = q, i = 0; i < qBits.Length; position++, i++)

{

qBits[i] = PthBit(number, position);

}

for (int position = p, i = 0; i < qBits.Length; position++, i++)

{

number = ModifiedNumber(number, position, qBits[i]);

}

for (int position = q, i = 0; i < pBits.Length; position++, i++)

{

number = ModifiedNumber(number, position, pBits[i]);

}

return number;

}

private static int PthBit(uint number, int position)

{

uint pthBit = (number >> position) & 1;

return (int)pthBit;

}

private static uint ModifiedNumber(uint number, int position, int bitValue)

{

uint actualP = (uint)bitValue << position;

number = number & (~((uint)1 << position));

uint result = number | actualP;

return result;

}

static void Main(string[] args)

{

Console.Write("Enter number: ");

uint number = uint.Parse(Console.ReadLine());

Console.Write("Enter p: ");

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

Console.Write("Enter q: ");

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

Console.Write("Enter k: ");

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

if (p > q)

{

int oldValue = p;

p = q;

q = oldValue;

}

if (p + k >= q)

{

k += p - q - 1;

q += p + k + 1;

}

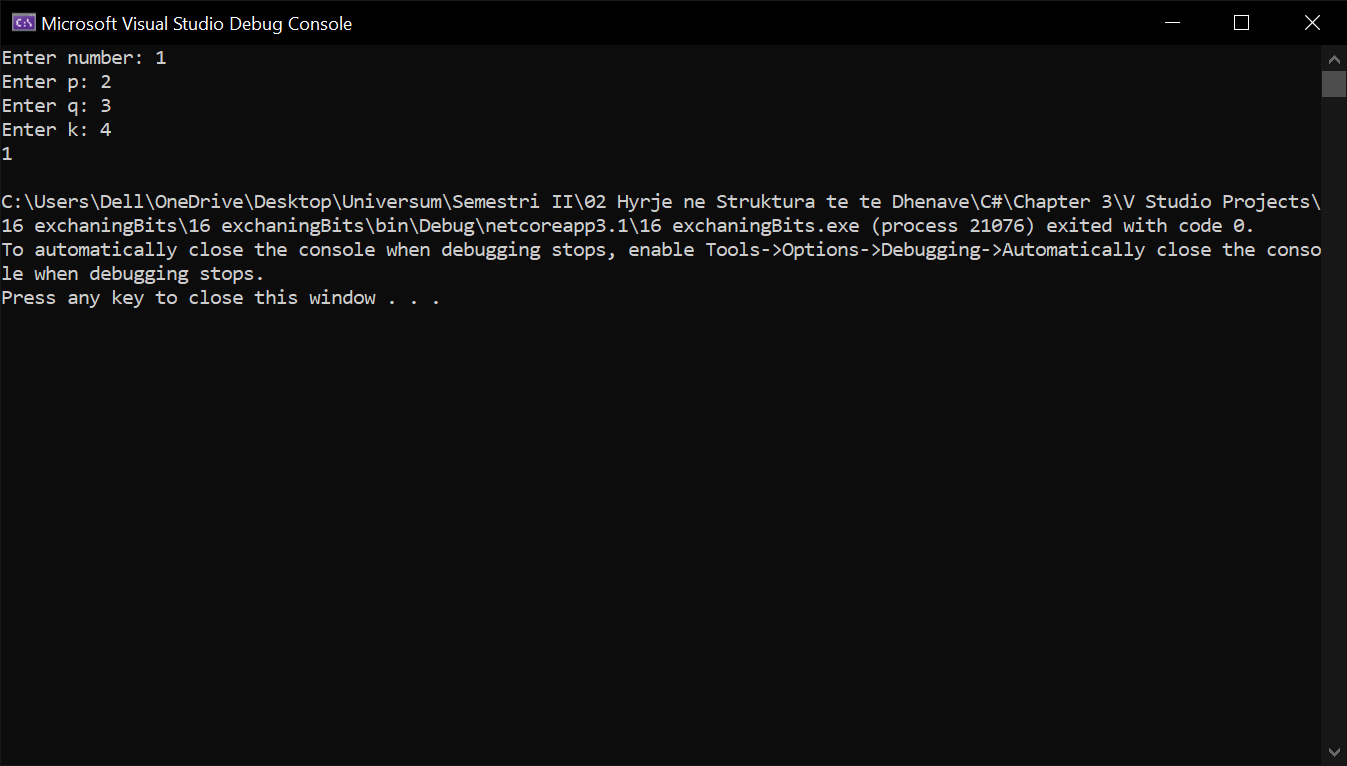
number = ModifyNumber(number, p, q, k);

Console.WriteLine(number);

}

}

}



Enis Neziri, Ferizaj 2021, SHK.