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

using System;

namespace detyra1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter a random number: ");

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

bool even = num % 2 == 0 ? true : false;

if(even == true)

{

Console.WriteLine("Your number is even");

}

else

{

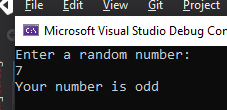
Console.WriteLine("Your number is odd");

}

}

}

}



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

using System;

class Program

{

static void Main()

{

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

bool check = num % 5 == 0 && num % 7 == 0 ? true : false;

if(check == true)

{

Console.WriteLine("Ky numer plotpjestohet me 7 dhe 5");

}

else

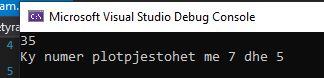
{

Console.WriteLine("Ky numer nuk plotpjestohet me 5 dhe 7");

}

}

}



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

using System;

class Program

{

static void Main()

{

Console.Write("Write a number: ");

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

int thirdNum = (num / 100) % 10;

if (thirdNum == 7)

{

Console.WriteLine("The third digits IS seven!");

}

else

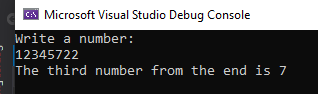
{

Console.WriteLine("The third digit IS NOT seven.");

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Detyra4

{

class Program

{

static void Main()

{

Console.WriteLine("Enter a number: ");

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

bool bit3 = (num & 8) != 0;

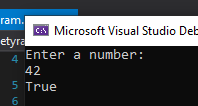
Console.WriteLine(bit3);

Console.ReadKey();

}

}

}



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

using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter Base a");

double a = double.Parse(Console.ReadLine());

Console.WriteLine("Enter Base b");

double b = double.Parse(Console.ReadLine());

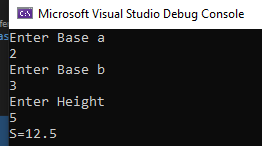
Console.WriteLine("Enter Height");

double h = double.Parse(Console.ReadLine());

Console.WriteLine("S={0}",( (a + b) / 2) \* h);

}

}



6.    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 Area\_Perimeter

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a");

double a = double.Parse(Console.ReadLine());

Console.WriteLine("Enter h");

double h = double.Parse(Console.ReadLine());

double perimetri = 2 \* (a + h);

double siperfaqja = a \* h;

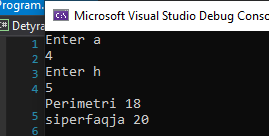
Console.WriteLine("Perimetri {0} \nsiperfaqja {1}", perimetri, siperfaqja);

Console.ReadKey();

}

}

}



7.    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;

class Program

{

static void Main()

{

Console.WriteLine("Please enter your current weight: ");

double weightEarth = double.Parse(Console.ReadLine());

double weightMoon = weightEarth \* 0.17;

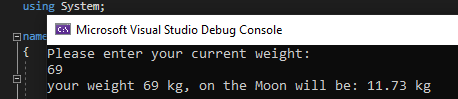
Console.WriteLine("your weight {0} kg, on the Moon will be: {1} kg",

weightEarth, weightMoon);

Console.ReadKey();

}

}



8.    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 Detyra8

{

class Program

{

static void Main()

{

Console.Write("Enter x: ");

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

Console.Write("Enter y: ");

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

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

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

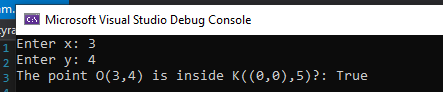
x, y, inside);

Console.ReadKey();

}

}

}



9.    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;

public class Program

{

public static void Main()

{

Console.Write("Enter x: ");

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

Console.Write("Enter y: ");

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

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

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

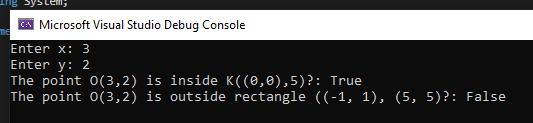
Console.WriteLine("The point O({0},{1}) is inside K((0,0),5)?: {2}", x, y,

inside);

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

}

}



10.   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;

class Program

{

static void Main()

{

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

int firstNumber = ( num / 1000) % 10;

int secondNumber = ( num / 100) % 10;

int thirdNumber = ( num / 10) % 10;

int fourNumber = ( num % 10);

Console.WriteLine("The sum of the digits is: {0}", firstNumber + secondNumber

+ thirdNumber + fourNumber);

Console.WriteLine("The digits in revers are: {0}{1}{2}{3}", fourNumber,

thirdNumber, secondNumber, firstNumber);

Console.WriteLine("The last digit in the first position {0}{1}{2}{3}",

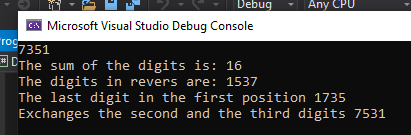
fourNumber, firstNumber, secondNumber, thirdNumber);

Console.WriteLine("Exchanges the second and the third digits {0}{1}{2}{3} ",

firstNumber, thirdNumber, secondNumber, fourNumber);

}

}



11.   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 Detyra11

{

class Program

{

static void Main()

{

Console.WriteLine("Enter a number: ");

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

Console.WriteLine("Enter the position of bit: ");

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

int i = 1;

int mask = i << poz;

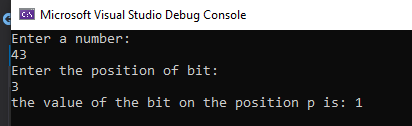
Console.Write("the value of the bit on the position p is: ");

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

}

}

}



12.   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 Detyra12

{

class Program

{

static void Main()

{

Console.Write("Enter a number: ");

int val1 = int.Parse(Console.ReadLine());

Console.Write("Enter a number bit pozition: ");

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

int vlera = 1 << poz;

bool correct = (val1 & vlera) != 0 ? true : false;

if (correct == true)

{

Console.WriteLine("The bit at position {0} of number {1} is 1",

poz, val1);

}

else

{

Console.WriteLine("The bit at position {0} of number {1} is 0",

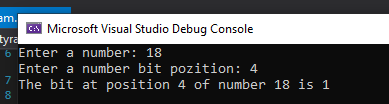
poz, val1);

}

}

}

}



13.   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 Detyra13

{

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);

}

}

}



14.   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 Detyra14

{

class Program

{

static void Main(string[] args)

{

int num = 23;

bool isPrime = true;

if (num > 2)

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

{

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

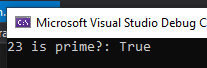
}

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

}

}

}



15.   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 Detyra15

{

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);

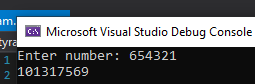
v = (bitAt26 == 0) ? v = v & (~(1 << 5)) : v = v | (1 << 5);

Console.WriteLine(v);

}

}

}



16.   \* 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 Detyra16

{

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);

}

}

}

