

Detyra te shtepise

Hyrje ne Struktura e te te Dhenave

Studenti/ja: Gerti Gonxhi

Kampusi Prishtine

Viti I-Grupi II

Ligjeruesi: Laberion Zebica

1.    Convert the numbers **151**, **35**, **43**, **251**, **1023**and **1024** to the **binary numeral system**.

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX1: Convert the numbers 151, 35, 43, 251, 1023 and 1024 to the binary numeral system.");

int[] numbersToConvert = { 151, 35, 43, 251, 1023, 1024 };

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

{

int numberToConvert = numbersToConvert[i];

string result = "";

while (numberToConvert > 0)

{

result = result.Insert(0, (numberToConvert % 2).ToString());

numberToConvert = numberToConvert / 2;

}

Console.WriteLine("{0} = {1}", numbersToConvert[i], result);

}

}

}

}

2.    Convert the number **1111010110011110(2)** to **hexadecimal** and **decimal** numeral systems.

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX2: Convert the number 1111010110011110(2) to hexadecimal and decimal numeral systems.");

string[] conversionTable = ("0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F").Split(new char[] { ',' });

string binNumber = "1111010110011110";

string resultHex = "";

double resultDec = 0;

char[] binNumberArray = binNumber.ToCharArray();

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

{

resultDec += int.Parse(binNumberArray[i].ToString()) \* Math.Pow(2, binNumberArray.Length - i - 1);

}

Console.WriteLine("Bin number {0} equal {1} in dec.", binNumber, resultDec);

double resultDecTmp = resultDec;

while (resultDecTmp > 0)

{

resultHex = resultHex.Insert(0, conversionTable[(int)(resultDecTmp % 16)]);

resultDecTmp = Math.Floor(resultDecTmp / 16);

}

Console.WriteLine("Bin number {0} equal {1} in hex.", binNumber, resultHex);

}

}

}

3.    Convert the hexadecimal numbers **FA**, **2A3E**, **FFFF**, **5A0E9** to **binary** and **decimal** numeral systems.

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX3: Convert the hexadecimal numbers FA, 2A3E, FFFF, 5A0E9 to binary and decimal numeral systems.");

string[] inputHexaNumbers = { "FA", "2A3E", "FFFF", "5A0E9" };

string[] conversionTableHexToDec = "0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F".Split(new char[] { ',' });

string[] conversionTableHexToBin = "0000,0001,0010,0011,0100,0101,0110,0111,1000,1001,1010,1011,1100,1101,1110,1111".Split(new char[] { ',' });

foreach (var inputHexNumber in inputHexaNumbers)

{

double resultDecEx3 = 0;

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

{

resultDecEx3 +=

Array.IndexOf(conversionTableHexToDec, inputHexNumber.Substring(i, 1)) \*

Math.Pow(16, inputHexNumber.Length - i - 1);

}

Console.WriteLine("Hex number {0} equal {1} in dec.", inputHexNumber, resultDecEx3);

string resultBinEx3 = "";

string hexStrTmp = "";

int hexStrTmpIndex = 0;

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

{

hexStrTmp = inputHexNumber.Substring(i, 1);

hexStrTmpIndex = Array.IndexOf(conversionTableHexToDec, hexStrTmp);

resultBinEx3 += conversionTableHexToBin[hexStrTmpIndex];

}

Console.WriteLine("Hex number {0} equal {1} in bin.", inputHexNumber, resultBinEx3);

}

}

} }

4.    Write a program that converts a **decimal number to binary** one.

using System;

namespace ConsoleApp4

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX4: Convert a positive decimal number to binary.");

Console.Write("Enter positive decimal number: ");

string consoleInputEx4 = Console.ReadLine();

bool consoleInputIsDec = decimal.TryParse(consoleInputEx4, out decimal consoleInputDec);

if (consoleInputIsDec)

{

decimal decTmp = consoleInputDec;

string decToBinResult = decTmp == 0 ? "0" : "";

while (decTmp > 0)

{

decToBinResult = decToBinResult.Insert(0, (decTmp % 2).ToString());

decTmp = Math.Floor(decTmp / 2);

}

Console.WriteLine("{0} in dec = {1} in bin", consoleInputEx4, decToBinResult);

}

else

{

Console.WriteLine("{0} is not a positive decimal number!!!", consoleInputEx4);

}

}

}

}

5.    Write a program that converts a **binary number to decimal** one.

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX5: Converts a binary number to a decimal.");

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

string consoleInputEx5 = Console.ReadLine();

ulong binToDecResult = 0;

bool isNumber = false;

for (int i = 0; i < consoleInputEx5.Length; i++)

{

isNumber = int.TryParse(consoleInputEx5.Substring(i, 1), out int consoleInputParseNumber);

if (isNumber && (consoleInputParseNumber == 0 || consoleInputParseNumber == 1))

{

binToDecResult += (ulong)(consoleInputParseNumber \* Math.Pow(2, consoleInputEx5.Length - i - 1));

}

else

{

Console.WriteLine("{0} is not binary number!!!", consoleInputEx5);

break;

}

}

Console.WriteLine("{0} in bin = {1} in dec", consoleInputEx5, binToDecResult);

}

}

}

6.    Write a program that converts a **decimal number to hexadecimal** one.

using System;

namespace ConsoleApp6

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX6: Converts a decimal number to hexadecimal.");

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

string consoleInputEx6 = Console.ReadLine();

bool consoleInputEx6IsNumber = decimal.TryParse(consoleInputEx6, out decimal consoleInputParseToDecimal);

if (consoleInputEx6IsNumber)

{

string decToHexResult = "";

string[] conversionTableEx6 = ("0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F").Split(new char[] { ',' });

decimal decToHexTmp = consoleInputParseToDecimal;

while (decToHexTmp > 0)

{

decToHexResult = decToHexResult.Insert(0, conversionTableEx6[(int)(decToHexTmp % 16)]);

decToHexTmp = Math.Floor(decToHexTmp / 16);

}

Console.WriteLine("{0} in dec = {1} in hex", consoleInputEx6, decToHexResult);

}

else

{

Console.WriteLine("{0} is not a hex number!!!", consoleInputEx6);

}

}

}

}

7.    Write a program that converts a **hexadecimal number to decimal** one.

using System;

namespace ConsoleApp7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX7: Converts a hexadecimal number to a decimal.");

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

string inputConsoleHex = (Console.ReadLine()).ToUpper();

ulong hexToDecResult = 0;

bool isHexNumber = true;

string invalidCharacter = "";

int invalidCharacterIndex = 0;

string[] conversionTableEx7 = ("0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F").Split(new char[] { ',' });

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

{

int indexOfString = Array.IndexOf(conversionTableEx7, inputConsoleHex.Substring(i, 1));

if (indexOfString >= 0 && indexOfString < 16)

{

hexToDecResult += (ulong)(indexOfString \* Math.Pow(16, inputConsoleHex.Length - i - 1));

}

else

{

isHexNumber = false;

invalidCharacter = inputConsoleHex.Substring(i, 1);

invalidCharacterIndex = i + 1;

break;

}

}

if (isHexNumber)

{

Console.WriteLine("{0} in hex = {1} in dec", inputConsoleHex, hexToDecResult);

}

else

{

Console.WriteLine("{0} is not a hexadecimal number!!! Invalid char {1} at position {2} " +

"from the left side.", inputConsoleHex, invalidCharacter, invalidCharacterIndex);

}

} } }

8.    Write a program that converts a **hexadecimal number to binary** one.

using System;

namespace ConsoleApp8

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX8: Converts a hexadecimal number to binary.");

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

string consoleInputHexNumber = Console.ReadLine().ToUpper();

string binResult = "";

bool inputIsHexNumber = true;

string[] arrayHexChars = "0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F".Split(new char[] { ',' });

string[] arrayHexToBin = ("0000,0001,0010,0011,0100,0101,0110,0111," +

"1000,1001,1010,1011,1100,1101,1110,1111").Split(new char[] { ',' });

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

{

int indexOfHexChar = Array.IndexOf(arrayHexChars, consoleInputHexNumber.Substring(i, 1));

if (indexOfHexChar >= 0)

{

binResult += arrayHexToBin[indexOfHexChar];

}

else

{

inputIsHexNumber = false;

Console.WriteLine("{0} is not hex number!!! Value {1} at position {2} " +

"from left side is invalid", consoleInputHexNumber, consoleInputHexNumber.Substring(i, 1), i + 1);

break;

}

}

if (inputIsHexNumber)

{

Console.WriteLine("{0} in hex = {1} in bin", consoleInputHexNumber, binResult);

}

}

}

}

9.    Write a program that converts a **binary number to hexadecimal** one.

using System;

namespace ConsoleApp9

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX9: Converts a binary number to hexadecimal.");

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

string consoleInputBinNumber = Console.ReadLine();

string hexResult = "";

bool inputIsBinNumber = true;

string[] arrayHexCharsEx9 = "0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F".Split(new char[] { ',' });

string[] arrayBinStringsEx9 = ("0000,0001,0010,0011,0100,0101,0110,0111," +

"1000,1001,1010,1011,1100,1101,1110,1111").Split(new char[] { ',' });

if (consoleInputBinNumber.Length < 4)

{

consoleInputBinNumber = consoleInputBinNumber.Insert(0, new String('0', 4 - consoleInputBinNumber.Length));

}

if (consoleInputBinNumber.Length % 4 != 0 && consoleInputBinNumber.Length > 4)

{

consoleInputBinNumber = consoleInputBinNumber.Insert(0, new String('0', consoleInputBinNumber.Length % 4));

}

for (int i = 0; i < consoleInputBinNumber.Length; i += 4)

{

int indexOfBinString = Array.IndexOf(arrayBinStringsEx9, consoleInputBinNumber.Substring(i, 4));

if (indexOfBinString >= 0)

{

hexResult += arrayHexCharsEx9[indexOfBinString];

}

else

{

inputIsBinNumber = false;

Console.WriteLine("{0} is not hex number!!! 4-bits {1} at position {2} " +

"from left side are invalid", consoleInputBinNumber, consoleInputBinNumber.Substring(i, 4), i + 1);

break;

}

}

if (inputIsBinNumber)

{

Console.WriteLine("{0} in bin = {1} in hex", consoleInputBinNumber, hexResult);

}

}

}

}

10.   Write a program that converts a **binary number to decimal** using the Horner scheme.

using System;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX10: Converts a binary number to decimal using the Horner scheme / algorithm.");

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

string inputBinConsole = Console.ReadLine();

ulong binToDecResultEx10 = 1;

for (int i = 0; i < inputBinConsole.Length - 1; i++)

{

int nextBinNumber = int.Parse(inputBinConsole.Substring(i + 1, 1));

binToDecResultEx10 = binToDecResultEx10 \* 2 + (ulong)nextBinNumber;

}

Console.WriteLine("{0} in bin = {1} in dec", inputBinConsole, binToDecResultEx10);

}

}

}

11.   Write a program that converts **Roman digits to Arabic** ones.

using System;

namespace ConsoleApp11

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX11: Program that converts Roman digits to Arabic.");

Console.Write("Enter number in Roman number system: ");

string consoleInputRomanNumber = Console.ReadLine().ToUpper();

int romanToArabicResult = 0;

bool isRomanNumber = true;

int[] romanToArabicNumbers = new int[consoleInputRomanNumber.Length];

int consoleInputLengthReverse = 0;

while (consoleInputLengthReverse < consoleInputRomanNumber.Length && isRomanNumber)

{

switch (consoleInputRomanNumber.Substring(consoleInputLengthReverse, 1))

{

case "I":

romanToArabicNumbers[consoleInputLengthReverse] = 1;

break;

case "V":

romanToArabicNumbers[consoleInputLengthReverse] = 5;

break;

case "X":

romanToArabicNumbers[consoleInputLengthReverse] = 10;

break;

case "L":

romanToArabicNumbers[consoleInputLengthReverse] = 50;

break;

case "C":

romanToArabicNumbers[consoleInputLengthReverse] = 100;

break;

case "D":

romanToArabicNumbers[consoleInputLengthReverse] = 500;

break;

case "M":

romanToArabicNumbers[consoleInputLengthReverse] = 1000;

break;

default:

isRomanNumber = false;

break;

}

consoleInputLengthReverse++;

}

if (isRomanNumber)

{

for (int i = 0; i < romanToArabicNumbers.Length - 1; i++)

{

if (romanToArabicNumbers[i] >= romanToArabicNumbers[i + 1])

{

romanToArabicResult += romanToArabicNumbers[i];

}

else

{

romanToArabicResult -= romanToArabicNumbers[i];

}

}

romanToArabicResult += romanToArabicNumbers[romanToArabicNumbers.Length - 1];

Console.WriteLine("{0} in Roman = {1} in dec (Arabic)", consoleInputRomanNumber, romanToArabicResult);

}

else

{

Console.WriteLine("{0} is NOT a Roman number!!!", consoleInputRomanNumber);

}

}

}

}

12.   Write a program that converts **Arabic digits to Roman** ones.

using System;

namespace ConsoleApp12

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX12: Program that converts Arabic digits to Roman ones.");

Console.Write("Enter number from 1 to 3999: ");

string inputConsole = Console.ReadLine();

string[][] romanNumbers = {

new string[] { "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX" },

new string[] { "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC" },

new string[] { "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM" },

new string[] { "M", "MM", "MMM" }

};

string resultInRoman = "";

for (int i = 0, j = inputConsole.Length - 1; i < inputConsole.Length; i++, j--)

{

int inputNumber = int.Parse(inputConsole.Substring(j, 1));

resultInRoman = resultInRoman.Insert(0, romanNumbers[i][inputNumber - 1]);

}

Console.WriteLine("{0} in Arabic = {1} in Roman", inputConsole, resultInRoman);

}

}

}

13.   Write a program that by given **N**, **S**, **D** (**2** **≤** **S**, **D** **≤** **16**) converts the number **N** from an **S**-based numeral system to a **D**based numeral system.

using System;

namespace ConsoleApp13

{

class Program

{

static void Main(string[] args)

{

int s, d;

Console.Write("Enter N: ");

string n = Console.ReadLine();

do

{

Console.Write("Enter S (S == 2, 8, 10 or 16): ");

s = Int32.Parse(Console.ReadLine());

} while (s != 2 && s != 8 && s != 10 && s != 16);

do

{

Console.Write("Enter D (D == 2, 8, 10 or 16): ");

d = Int32.Parse(Console.ReadLine());

} while (d != 2 && d != 8 && d != 10 && d != 16);

n = Convert.ToString(Convert.ToInt32(n, s), d);

Console.WriteLine("Result is {0}.", n);

}

}

}

14.   Try **adding up 50,000,000 times the number 0.000001**. Use a loop and addition (not direct multiplication). Try it with **float** and **double** and after that with **decimal**. Do you notice the **huge difference in the results** and speed of calculation? Explain what happens.

using System;

namespace ConsoleApp14

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EX:14 compare result for float, double and decimal");

Console.WriteLine("Add real number 0.000001 50 000 000 times using float, double and decimal type:");

float numberFloat = 0.0f;

for (int i = 0; i < 50000000; i++)

{

numberFloat += 0.000001f;

}

Console.WriteLine($"Float: {numberFloat} not accurate result, but the high speed.");

double numberDouble = 0.0d;

for (int i = 0; i < 50000000; i++)

{

numberDouble += 0.000001d;

}

Console.WriteLine($"Double: {numberDouble} not accurate result, but the high speed.");

decimal numberDecimal = 0.0M;

for (int i = 0; i < 50000000; i++)

{

numberDecimal += 0.000001M;

}

Console.WriteLine($"Decimal: {numberDecimal} the resault is accurate, but the speed is slowest.");

}

}

}

15. Write a program that prints the value of the **mantissa**, the **sign of the mantissa** and **exponent** in **float** numbers (32-bit numbers with a floating-point according to the **IEEE 754** standard). Example: for the number **-27.25** should be printed: **sign** = **1**, **exponent** = **10000011**, **mantissa** = **10110100000000000000000**.

using System;

namespace ConsoleApp15

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(new String('#', 80));

Console.WriteLine();

}

}

}