|  |  |
| --- | --- |
|  | |
| **Semestrální projekt č. 2** | |
|  | |
| David Papaj | |
|  | |
|  |  |
| Semestrální projekt  2021 |  |
|  |  |
|  | |

OBSAH

[zadání 3](#_Toc65583390)

[1 popis programu a jeho funkcí 4](#_Toc65583391)

[2.1 Převod binární soustavy do dekadické 5](#_Toc65583393)

[2.2 Převod dekadické soustavy do binární 5](#_Toc65583394)

[2.3 Převod hexadecimální soustavy do binární 6](#_Toc65583395)

[2.4 Převod binární soustavy do hexadecimální 7](#_Toc65583396)

[2.5 Převod Oktálové soustavy do dekadické 8](#_Toc65583397)

[2.6 Převod dekadické soustavy do oktálové 9](#_Toc65583398)

[2.7 Převod číslic do římských čísel 10](#_Toc65583399)

[2.8 Kód pro zapisování vstupů pro převod soustav 13](#_Toc65583400)

zadání

Sestavte program (funkce) pro převod čísel mezi různými soustavami bez použití vestavěných funkcí prostředí Mathematica. Dále sestavte funkci pro převod čísel do soustavy římských čísel a výpočet počtu míst při převodu čísla v libovolných soustavách.

popis programu a jeho funkcí

Tento program provádí převod soustav 2 na 10 a zpět, 8 na 10 a zpět, 2 na 16 a zpět a převod čísla do římských číslic.

První částí programu je kód, který vypočítává v různých způsobech převod daných soustav a zde je také popsáni jak byl daný převod soustav vypočítán.

Druhá část je kód, kde můžeme vložit dané číslo v soustavě, které chceme přeložit do jiné.

# Demonstrace konkrétního příkladu

Převod binární soustavy do dekadické

Každé binární číslo vynásobíme vzorcem e 2^0, která se pak vsadí do proměnné dek..

Když např. budeme chtít převézt binární číslo 101, tak poté vynásobíme každé číslo smaostatně.1\*(2^2)+0\*(2^1)+1\*(2^0)=5

static int BinarniNaDekadicka(String a)

{

String bin = a;

int dek = 0;

// Startovací hodnota je 1, kde používáme vzorec i.e 2^0.

int x = 1;

int len = bin.Length;

for (int i = len - 1; i >= 0; i--)

{

if (bin[i] == '1')

dek += x;

x = x \* 2;

}

return dek;

}

Převod dekadické soustavy do binární

Každé číslo je vyděleno 2 a zbytek je uložen. Opakujeme dokud není číslo 0

Poté daný výsledek "vytiskneme" v opačném pořadí

static int DekadickaNaBinarni(int b)

{

// Kod je použit pro uložení binarního čísla

int Bin = 0;

int y = 0;

while (b != 0)

{

int rem = b % 2;

int c = (int)Math.Pow(10, y);

Bin += rem \* c;

b /= 2;

// Ukládání hodnoty exponentu

y++;

}

return Bin;

}

Převod hexadecimální soustavy do binární

Převedme každé číslo samostatně podle číslové abecedy a poté se nakonci všechny hodnoty spojí

static void HexDoBinarni(char[] hexdec)

{

int i = 0;

while (hexdec[i] != '\u0000')

{

//Číslová abeceda, protože hexadecimlaní soustava použivá abecedu pro tato čísla 10-16 .Kde musíme musíme napsat kod pro rozpoznání daných čísel a pote je přeložit.

switch (hexdec[i])

{

case '0':

System.Console.Write("0000");

break;

case '1':

System.Console.Write("0001");

break;

case '2':

System.Console.Write("0010");

break;

case '3':

System.Console.Write("0011");

break;

case '4':

System.Console.Write("0100");

break;

case '5':

System.Console.Write("0101");

break;

case '6':

System.Console.Write("0110");

break;

case '7':

System.Console.Write("0111");

break;

case '8':

System.Console.Write("1000");

break;

case '9':

System.Console.Write("1001");

break;

case 'A':

case 'a':

System.Console.Write("1010");

break;

case 'B':

case 'b':

System.Console.Write("1011");

break;

case 'C':

case 'c':

System.Console.Write("1100");

break;

case 'D':

case 'd':

System.Console.Write("1101");

break;

case 'E':

case 'e':

System.Console.Write("1110");

break;

case 'F':

case 'f':

System.Console.Write("1111");

break;

}

i++;

}

}

Převod binární soustavy do hexadecimální

Rozdělíme dané binární číslo do skupin po 4 čísel a začneme počítat hex. formu

Poté tuto hodnotu uložíme do vektoru. Toto opakujeme pro všechny číslice daného binárního čísla. Vytiskneme číslo uložené ve vektoru v opačném pořadí

static void bcdToHexaDecimal(char[] s)

{

int len = s.Length, kontrola = 0;

int sum = 0, mul = 1;

List<char> list = new List<char>();

//Opakuje skrz bity opačně(začínýme ze zadu)

for (int i = len - 1; i >= 0; i--)

{

sum += (s[i] - '0') \* mul;

mul \*= 2;

kontrola++;

// Vypočítání hexadecimalních soustav, z dosud vytvořených čísle a poté dané hodnoty uložíme do vektoru.

if (kontrola == 4 || i == 0)

{

if (sum <= 9)

list.Add((char)(sum + '0'));

else

list.Add((char)(sum + 55));

// Resetování hodnot pro další skupinu.

kontrola = 0;

sum = 0;

mul = 1;

}

}

len = list.Count;

// "Tisk" hexy. soustavy z dousd vytvořených čísel.

Console.Write($"2 do 16:Hexadecimální soustava má hodnotu = ");

for (int i = len - 1; i >= 0; i--)

Console.Write(list[i]);

}

Převod Oktálové soustavy do dekadické

Každé oktálové číslo vynásobíme vzorcem e 2^0, která se pak vsadí do proměnné dek.

Když např. budeme chtít převézt oktálové číslo 55, tak poté vynásobíme každé čislo smaostatně. 5\*(8^1)+5\*(8^0)=45

static int OktalovaDoDekadicke(int n1)

{

int cislo\_a = n1;

int dek\_hod = 0;

// Kod začne s základní hodnotou 1, vzorec i.e 8^0

int b\_a = 1;

int t = cislo\_a;

while (t > 0)

{

//Veme poslední hodnotu

int posledni\_c = t % 10;

t = t / 10;

// násobíme poslední hodnotu s vhodným základem dané hodnoty a přidání do dek\_hod

dek\_hod += posledni\_c \* b\_a;

b\_a = b\_a \* 8;

}

return dek\_hod;

}

Převod dekadické soustavy do oktálové

Vydělíme číslo v poli 8 a zbytek uložíme. Opakujeme dokud se číslo nerovná 0.

Poté vytiskneme daný výsledek v opačném pořadí

static void DekadickaNaOktalova(int n)

{

// array, kde ukládáme oktálové hodnoty

int[] oktalN = new int[100];

// Čítač pro oktálové hodnoty v daném array

int i = 0;

while (n != 0)

{

// Ukladáme zbytek do určeného oktálového array(pole)

oktalN[i] = n % 8;

n = n / 8;

i++;

}

// "Vytisknutí" oktálového čísla v poli v obráceném pořadí

Console.Write("10 do 8:Oktálová soustava má hodnotu = ");

for (int j = i - 1; j >= 0; j--)

Console.Write(oktalN[j]);

Console.WriteLine();

}

static int sub\_digit(char num\_1, char num\_2,

int i, char[] w)

{

w[i++] = num\_1;

w[i++] = num\_2;

return i;

}

// Přidání symbolu 'ch' \* několikrát po indexu i v poli w[]

static int digit(char ch, int n, int i, char[] w)

{

for (int j = 0; j < n; j++)

{

w[i++] = ch;

}

return i;

}

Převod číslic do římských čísel

Program funguje tak že samostatně převedeme desítky, stovky, tisíce.A také pro převod čísel jako 4 a 9 ,které vyžadují více symbolů v daném pořadí.

Např. číslo 1549:

Napřeď 1549 je větší jak 1000, tak můžeme dané číslo rozdělit jenom jednou ,takže zapíšeme M.

Poté dané číslo je 549 , které je větší jak 500, tak dan= číslo můžeme rozdělit zase jenom jednou a zapíšeme D.

Číslo je 49 : Dále číslo je větší jak 40, číslo můžeme rozdělit zase jenom jednou a zbytek je 9. Daný symbol je poté XL.

Číslo je 9 : Dále je 9 menší jak 10 a dané číslo odpovídá symbolu IX.

Poté spojíme dané hodnoty :MDXLIX

// Pro správné převedení daných čísel do správných symbolů

static int dil\_cis(char num\_1, char num\_2,

int i, char[] w)

{

w[i++] = num\_1;

w[i++] = num\_2;

return i;

}

// Pro přidání symbolu ch \*za index i v poli w

static int cislo\_c(char ch, int n, int i, char[] w)

{

for (int j = 0; j < n; j++)

{

w[i++] = ch;

}

return i;

}

// Funkce pro převod čísel do římských čísel

static void CisloToRimsky(int cislo)

{

char[] w = new char[10001];

int i = 0;

// Převedeme dekadické číslo do římských číslic

while (cislo != 0)

{

// Jestli hodnota čísla je větší nebo stejná jak 1000, tak ---

if (cislo >= 1000)

{

// ,tak přidáme M za index i

i = cislo\_c('M', cislo / 1000, i, w);

cislo = cislo % 1000;

}

// Jestli hodnota čísla je větší nebo stejná jak 500, tak ---

else if (cislo >= 500)

{

// ,tak přidáme shodný symbol :

if (cislo < 900)

{

// ,tak přidáme M za index i

i = cislo\_c('D', cislo / 500, i, w);

cislo = cislo % 500;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 9 a přidáním vhodného symbolu

else

{

// ,tak přidáme C a M za index i

i = dil\_cis('C', 'M', i, w);

cislo = cislo % 100;

}

}

// Jestli hodnota čísla je větší nebo stejná jak 100, tak ---

else if (cislo >= 100)

{

// ,tak přidáme shodný symbol :

if (cislo < 400)

{

i = cislo\_c('C', cislo / 100, i, w);

cislo = cislo % 100;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 4 a přidáním vhodného symbolu

else

{

i = dil\_cis('C', 'D', i, w);

cislo = cislo % 100;

}

}

// Jestli hodnota čísla je větší nebo stejná jak 50, tak ---

else if (cislo >= 50)

{

// ,tak přidáme shodný symbol :

if (cislo < 90)

{

i = cislo\_c('L', cislo / 50, i, w);

cislo = cislo % 50;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 9 a přidáním vhodného symbolu

else

{

i = dil\_cis('X', 'C', i, w);

cislo = cislo % 10;

}

}

// Jestli hodnota čísla je větší nebo stejná jak 10, tak ---

else if (cislo >= 10)

{

// ,tak přidáme shodný symbol :

if (cislo < 40)

{

i = cislo\_c('X', cislo / 10, i, w);

cislo = cislo % 10;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 4 a přidáním vhodného symbolu

else

{

i = dil\_cis('X', 'L', i, w);

cislo = cislo % 10;

}

}

// Jestli hodnota čísla je větší nebo stejná jak 5, tak ---

else if (cislo >= 5)

{

if (cislo < 9)

{

i = cislo\_c('V', cislo / 5, i, w);

cislo = cislo % 5;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 9 a přidáním vhodného symbolu

else

{

i = sub\_digit('I', 'X', i, w);

cislo = 0;

}

}

// Jestli hodnota čísla je větší nebo stejná jak 1, tak ---

else if (cislo >= 1)

{

if (cislo < 4)

{

i = cislo\_c('I', cislo, i, w);

cislo = 0;

}

// Pro odčítání hodnot, kdy čísla mají hodnotu 4 a přidáním vhodného symbolu

else

{

i = dil\_cis('I', 'V', i, w);

cislo = 0;

}

}

}

// "Tisk" přeložených čísel do římských čísel

Console.Write("Číslo převedené do římského čísla je = ");

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

{

Console.Write("{0}", w[b]);

}

}

Kód pro zapisování vstupů pro převod soustav

Kód, kde zapisujme hodnoty, které poté chceme převádět do jiných soustav a také obsahuje i kód pro "tisk" daných funkcí.

public static void Main()

{

//2to10

String bin = "101";

Console.WriteLine($"2 do 10:Dekadícká soustava má hodnotu = {BinarniNaDekadicka(bin)}");

//10to2

int b = 5;

Console.WriteLine($"10 do 2:Binární soutava má hodnotu = {DekadickaNaBinarni(b)}");

//2to16

String s = "100000101111";

bcdToHexaDecimal(s.ToCharArray());

//16to2

string q = "1AC5";

char[] hexdec = new char[100];

hexdec = q.ToCharArray();

Console.WriteLine(" ");

// Převádění hexadecimální soustavy do binarní

System.Console.Write("16 do 2:Binární soutava má hodnotu = ");

try

{

HexDoBinarni(hexdec);

}

catch (System.IndexOutOfRangeException)

{

System.Console.WriteLine("");

}

//8to10

int num = 55;

Console.WriteLine($"8 do 10:Dekadická soutava má hodnotu = {OktalovaDoDekadicke(num)}");

//10to8

int n = 8;

DekadickaNaOktalova(n);

//Číslo převedené na římské číslo

int cislo = 4;

CisloToRimsky(cislo);

}

