Homework №1. Numeral systems and conversion of numbers, Variant №1

FMISB 18100 Informational Technologies
Lecturer: Darius Zabulionis

Assignment. Write a program that converts the real decimal number x_{10} into the number x_r of the required radix r with the required precision ϵ_{sup}, by using any general purpose programminmg language: Java, C++, C, Python or other. The required number s of the significant numerals d_i, $s \leq i < 0$ of the converted number $\hat{x}_r$ must meet the condition for the error of the converted number:

$$\epsilon = x_r - \hat{x}_r \tag{1}$$

where x_r is the exact value of the converted number, while $\hat{x}_r$ is x_r approximation till the numeral of the fractional part whose position is $s < 0$.
Algorithms 1 and 2 can be used for the conversion. The program should not be written for the programme of the particular purpose, for example: Matlab, Octave, Scilab, R or similar. Thus, the programming languages of the general purpose should be used. The already implemented methods or functions for conversion of the numbers must not be used. The methods or/and functions for the operating the text strings and arrays can be or must be used. For example: in Java `String.reverse()`; Pitono (Python) `List.reverse()` and so on.

The work must be individual, the programm code must be unique. That is the program code cannot be the same as of other students. The work must be defendet in the classroom during the laboratory work lectures. Submition by the remote mode or by the emailing is not allowed.

List of the tasks

1. The initial decimal number must be entered from the command line or read from the original data file. That is, the number must not be "hardcoded".
2. The required number of the significant digits has to be calculated according to the formula (1).
3. The results of the calculation must be displayed on the screen or in the application console.
4. The code (text) of the programme must be written clearly and neatly.
5. the output results must be comprehensive:
   (a) The initial data must be displayed or printed:
      i. The initial decimal number x_{10},
      ii. The radix of the target system r into that the initial decimal number x_{10} has to be converted must be displayed or printed
      iii. The precision of the conversion ϵ_{sup} must be displayed or printed
   (b) The output calculation results must be comprehensible:
      i. To show the required number s of the significant numerals of the
      ii. The converted number of radix r

The possible variant of the outup results is shown below.

```
I THE INITIAL DATA
    1.1 The initial decimal number: x_10 = 101,2003
    1.2 The radix of the target numeral system: r = 3
    1.3 The required precission of the converted number: epsilon_sup = 1e-4

II THE REZULTS OF THE CONVERSION
    2.1 The required number of the significant numerals s = 9
        that corresponds the given prcission epsilon_sup = 1e-4
    2.2 The result of the conversion: x_3 = ...
```

Algorithm 1: Conversion of the integer part $floor(x_{10})$ of the decimal number x_{10} into the integer part $floor(x_r)$ of the number x_r of radix r

Data: Decimal number x_{10}, radix r of the target numerical system.
Result: Converted the integer part $int(x_{10}) = int(x_r)$ of the initial decimal number x_{10}.

initialization
$I_{[0]} \leftarrow floor(x_{10})$ // The integer part of x_{10}
$i \leftarrow 0$ // The initial value for the counter i
while $I_i > 0$ **do**
    $i \leftarrow i + 1$ // The new value for the counter i
    $I_{[i]} \leftarrow floor(I_{[i-1]}/r)$
    $d_{[i-1]} \leftarrow I_{[i-1]} - I_{[i]}r$
end
$floor(x_r) \leftarrow append(d_{[i-1]}d_{[i-2]} \dots d_{[2]}d_{[1]}d_{[0]})$ // The integer part of the converted number

Remarks: in Algorithm 1, the double slash // denotes the beginning of a comment; the operator $append(str_1, str_2)$ concatenates two strings str_1 and str_2; for example $append(\text{"}aaa\text{"}, \text{"}bbb\text{"})$ returns the string "$aaabbb$".

Algorithm 2: Conversion of the fractional part $frac(x_{10})$ of a decimal x_{10} into the fractional part $frac(x_r)$ of the r-radix number with the required precision ϵ.

Data: Dec. number x_{10}, radix r of the target numerical system, the required precision $\epsilon < 1$.
Result: Converted the fractional part $frac(x_{10}) = frac(x_r)$ of the initial dec. number x_{10} whose error $x_{10} - \hat{x}_{10} < \epsilon$.

initialization
$f_{[0]} \leftarrow frac(x_{10})$; $i = 0$; $er = \epsilon + 1$; $\hat{x}_r =$ '0,'; $\hat{x}_{10} = 0$; // The initial values of f_0, i, error er, the string of the converted number $\hat{x}_r$, and the decimal value $\hat{x}_{10}$ of the converted number $\hat{x}_r$
while $er \geq \epsilon$ **do**
    $i \leftarrow i + 1$; // The new value for the counter i
    $d_{[-i]} \leftarrow floor(f_{[i-1]}r)$;
    $f_{[i]} \leftarrow f_{[i-1]}r - d_{[-i]}$; // Or $f_{[i]} \leftarrow frac(f_{[i-1]}r)$
    $\hat{x}_{10} = \hat{x}_{10} + d_{[-i]}r^{-i}$; // The decimal value $\hat{x}_{10}$ of converted $\hat{x}_r$
    $\hat{x}_r = append(\hat{x}_r, d_{[-i]})$; // Appended string of the converted $frac(x_r)$
    $er = f_{[0]} - \hat{x}_{10}$ // Error of the approximation of $frac(x_{10})$
end