KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Programavimo kalbų teorija (P175B124)

Laboratorinių darbų ataskaita

Atliko:

IFF-1/8 gr. studentas

Matas Palujanskas

2023 m. kovo 1 d.

Priėmė:

doc. Sajavičius Svajūnas

KAUNAS 2023

TURINYS

1. C++(L1)

1.1. Darbo užduotis

p575

When a number is expressed in decimal, the k-th digit represents a multiple of 10^k. (Digits are numbered from right to left, where the least significant digit is number 0.) For example,

$$81307_{10} = 8 \times 10^4 + 1 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 7 \times 100 = 80000 + 1000 + 300 + 0 + 7 = 81307.$$

When a number is expressed in binary, the k-th digit represents a multiple of 2^k. For example,

$$10011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 0 + 0 + 2 + 1 = 19.$$

In **skew binary**, the k-th digit represents a multiple of $2^{k+1} - 1$. The only possible digits are 0 and 1, except that the least-significant nonzero digit can be a 2. For example,

$$10120_{skew} = 1 \times (2^5 - 1) + 0 \times (2^4 - 1) + 1 \times (2^3 - 1) + 2 \times (2^2 - 1) + 0 \times (2^1 - 1) = 31 + 0 + 7 + 6 + 0 = 44.$$

The first 10 numbers in skew binary are 0, 1, 2, 10, 11, 12, 20, 100, 101, and 102. (Skew binary is useful in some applications because it is possible to add 1 with at most one carry. However, this has nothing to do with the current problem.)

Input

The input file contains one or more lines, each of which contains an integer n. If $n = 0$ it signals the end of the input, and otherwise n is a nonnegative integer in skew binary.

Output

For each number, output the decimal equivalent. The decimal value of n will be at most $2^{31} - 1 = 2147483647$.

Sample Input

```
10120
2000000000000000000000000000000
10
1000000000000000000000000000000
11
100
11111000001110000101101102000
0
```

Sample Output

```
44
2147483646
3
2147483647
4
7
1041110737
```

1.2. Programos tekstas

```
#include <iostream>
#include <fstream>
#include <string>
#include <cmath>
#include <chrono>
using namespace std;
using namespace std::chrono;

/// <summary>
/// Skew binary to decimal number class
/// </summary>
struct SkewBinaryToDecimal
{
    string line; //binary number

    /// <summary>
    /// Converting from skew binary to decimal number
    /// </summary>
    /// <param name="line"> Skew binary number </param>
    /// <param name="n"> Amount of numbers </param>
    /// <returns> Converted decimal number </returns>
    int to_decimal(string line, int& n)
    {
        int res = 0;
        for (int j = 0; j < n; j++)
        {
            int l = line.size();
            for (int i = 0; i < l; i++) {
                res += (((int)line[i] - 48) * (pow(2, l - i) - 1));
            }
            return res;
        }
    }
};

/// <summary>
/// Decimal to skew binary number class
/// </summary>
struct DecimalToSkewBinary {

    int decimal; //decimal number

    string decimalToSkewBinary(int number) {

        if (number == 0) {
            return "0";
        }
        string result = "";
        while (number > 0) {
            int remainder = number % 3;
            result = to_string(remainder) + result;
            number /= 3;
            if (remainder == 2 && number > 0) {
                number++;
            }
        }
        return result;
    }
};


/// <summary>
/// Printing converted numbers to file
/// </summary>
/// <param name="outputFile"> Output file </param>
/// <param name="line"> Binary numbers </param>
```

```
/// <param name="convertedNumber"> Decimal numbers </param>
/// <param name="append"></param>
void PrintResult1(string outputFile, string line,
    int convertedNumber, bool append)
{
    ofstream result;

    if (append)
        result.open(outputFile, ios_base::app);
    else
        result.open(outputFile);
    result << "Skew binary: " << line << "  " << "decimal: " << convertedNumber <<
endl;
    return;
}

/// <summary>
/// Reading data and performing all tasks
/// </summary>
/// <param name="inputFile"> Data file </param>
/// <param name="outputFile"> Result file </param>
void ReadAndPerformToDecimal(string inputFile, string outputFile)
{
    int n;// Amount of numbers
    SkewBinaryToDecimal toDecimal[20];

    ifstream data(inputFile);
    data >> n;
    for (int i = 0; i < n; i++)
    {
        string line;
        data.ignore();
        data >> toDecimal[i].line;
        line = toDecimal[i].line;

        int calculated = toDecimal->to_decimal(line, n);

        //string skew_Binary = toBinary->to_skew_binary(calculated, n)

        bool append = false;
        if (i > 0)
            append = true;
        PrintResult1(outputFile, line, calculated, append);
    }

    data.close();
    return;
}
// <summary>
/// Printing converted numbers to file
/// </summary>
/// <param name="outputFile"> Output file </param>
/// <param name="line"> Binary numbers </param>
/// <param name="convertedNumber"> Decimal numbers </param>
/// <param name="append"></param>
void PrintResult2(string outputFile, int line,
    string convertedNumber, bool append)
{
    ofstream result;

    if (append)
        result.open(outputFile, ios_base::app);
    else
        result.open(outputFile);
    result << "Decimal: " << line << "  " << "skew binary: " << convertedNumber <<
endl;
    return;
}

/// <summary>
```

```
/// Reading data and performing all tasks
/// </summary>
/// <param name="inputFile"> Data file </param>
/// <param name="outputFile"> Result file </param>
void ReadAndPerformToBinary(string inputFile, string outputFile)
{
    int n;// Amount of numbers
    DecimalToSkewBinary toBinary[100];

    ifstream data(inputFile);
    data >> n;
    for (int i = 0; i < n; i++)
    {
        int line;
        data.ignore();
        data >> line;
        //line = toBinary[i].decimal;

        string calculated = toBinary->decimalToSkewBinary(line);

        bool append = false;
        if (i > 0)
            append = true;
        PrintResult2(outputFile, line, calculated, append);
    }

    data.close();
    return;
}

int main()
{
    string inputFile1 = "Data1.txt";
    string outputFile1 = "Results1.txt";
    string inputFile2 = "Data2.txt";
    string outputFile2 = "Results2.txt";

    // Duration of operations start point
    auto start = high_resolution_clock::now();

    // Main calculations method
    ReadAndPerformToDecimal(inputFile1, outputFile1);
    ReadAndPerformToBinary(inputFile2, outputFile2);

    // Duration of operations end point
    auto stop = high_resolution_clock::now();

    // Duration
    auto duration = duration_cast<microseconds>(stop - start);
    cout << "Time taken by function: "
        << duration.count() << " microseconds" << endl;

    return 0;
}
```

1.3. Pradiniai duomenys ir rezultatai

Data1(skew binary numbers)

Data1.txt:

```
8
10120
20000000000000000000000000000000
10
10000000000000000000000000000000
11
100
11111000001110000101101102000
0
```

Result1.txt:

```
Skew binary: 10120  decimal: 44
Skew binary: 20000000000000000000000000000000  decimal: 2147483646
Skew binary: 10  decimal: 3
Skew binary: 10000000000000000000000000000000  decimal: 2147483647
Skew binary: 11  decimal: 4
Skew binary: 100  decimal: 7
Skew binary: 11111000001110000101101102000  decimal: 1041110737
Skew binary: 0  decimal: 0
```

Data2 (binary numbers)

Data2.txt:

```
8
44
2147483646
3
2147483647
4
7
1041110737
0
```

Result2.txt:

```
Decimal: 44  skew binary: 2202
Decimal: 2147483646  skew binary: 20220200022111212100
Decimal: 3  skew binary: 10
Decimal: 2147483647  skew binary: 2022020022111212101
Decimal: 4  skew binary: 11
Decimal: 7  skew binary: 21
Decimal: 1041110737  skew binary: 10201220001020110021
Decimal: 0  skew binary: 0
```