

Спецификация лабораторной работы № 6

Данная программа выполняет ввод строк из файла input.txt и для каждой введенной строки выполняет проверку по правилу, указанному в варианте.

На выходе программа выводит слово "ДА", если все соответствует правилу и "НЕТ", если не соответствует.

Также структура программы составлялась по определенным замечаниям:

- Ввод и вывод информации разместить в функции main().
- Каждое определение должно быть оформлено в виде отдельной функции.
- Данные для проверки вводить из файлов, читая анализируемую строку целиком.

Задание:

Вариант 5

Понятие вещественное определено следующим образом:

<вещественное> ::= <мантисса> <порядок> | <знак> <мантисса> <порядок>

<мантисса> ::= . <целое без знака> | <целое без знака> . <целое без знака>

<порядок> ::= E <знак> <целое без знака> | E <целое без знака>

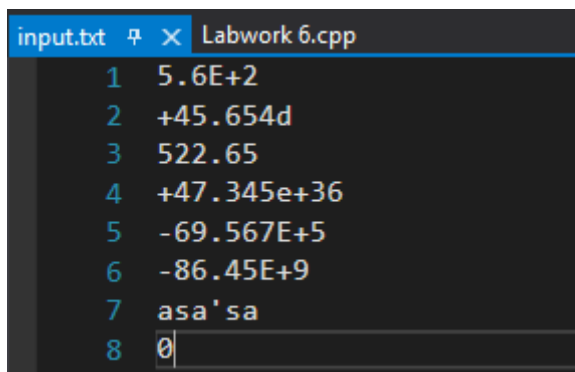
<целое без знака> ::= <цифра> | <цифра> <целое без знака>

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<знак> ::= + | -

Тестовый набор с выходными данными

В input.txt и input2.txt лежат входные данные для работы с матрицей.



```
input.txt 7 X Labwork 6.cpp
1 5.6E+2
2 +45.654d
3 522.65
4 +47.345e+36
5 -69.567E+5
6 -86.45E+9
7 asa'sa
8 0
```

Вывод:

```
Консоль отладки Microsoft Visual Studio
5.6E+2 | ДА
+45.654d | НЕТ
522.65 | НЕТ
+47.345e+36 | НЕТ
-69.567E+5 | ДА
-86.45E+9 | ДА
asa'sa | НЕТ
0 | НЕТ
```

Программа:

```
#include <iostream>
#include <string>
#include <fstream>

bool floatingPointNumber(std::string str, std::string mant, std::string ord);
bool isMantissa(std::string mant);
bool isNumber(std::string& str);
bool isSign(std::string& str);
bool noSignInteger(std::string str);
bool isOrder(std::string ord);

int main()
{
    setlocale(LC_ALL, "Russian");

    std::ifstream fin;
    std::string str;
    std::string mant = "";
    std::string ord = "";
    fin.open("input.txt");
    if (!fin)
    {
        std::cout << "Файл не открылся для чтения.";
    }
    while (!fin.eof())
    {
        fin >> str;

        bool flag = true;
        for (int i = 1; i < str.length(); i++)
        {
            if (str[i] == 'E')
            {
                flag = false;
            }

            if (flag)
            {
                mant += str[i];
            }
            else
            {
                ord += str[i];
            }
        }

        std::cout << str << " | ";
        std::cout << (floatingPointNumber(str, mant, ord) ? "ДА" : "НЕТ") <<
std::endl;
        mant = "";
    }
}
```

```

        ord = "";
    }
    fin.close();
    return 0;
}

bool floatingPointNumber(std::string str, std::string mant, std::string ord)
{
    if (isOrder(ord) && isMantissa(mant) && isSign(str) || isMantissa(mant) &&
isOrder(ord))
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isMantissa(std::string mant)
{
    std::string beforePoint = "";
    std::string afterPoint = "";
    bool flag = true;
    for (int i = 0; i < mant.length(); i++)
    {
        if (mant[i] == '.')
        {
            flag = false;
        }

        if (flag)
        {
            beforePoint += mant[i];
        }
        else
        {
            if (mant[i] != '.')
            {
                afterPoint += mant[i];
            }
        }
    }
    if ((noSignInteger(afterPoint)) || noSignInteger(beforePoint) &&
noSignInteger(afterPoint))
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool isOrder(std::string ord)
{
    if (ord[0] == 'E')
    {
        ord.erase(0, 1);
        if ((isSign(ord) && noSignInteger(ord)) || noSignInteger(ord))
        {
            return true;
        }
        else
        {

```

```

        return false;
    }
    return false;
}

bool noSignInteger(std::string str)
{
    if (isNumber(str))
    {
        noSignInteger(str);
        return true;
    }
    else
    {
        return false;
    }
}

bool isNumber(std::string& str)
{
    if (str[0] >= '0' && str[0] <= '9')
    {
        str.erase(0, 1);

        return true;
    }
    return false;
}

bool isSign(std::string& str)
{
    if (str[0] == '+' || str[0] == '-')
    {
        str.erase(0, 1);

        return true;
    }
    else
    {
        return false;
    }
}

```