ФИО: Гольчанский Максим Дмитриевич

 Группа: 3530904/10003

Тема: Рекурсия

Вариант №6

Дата отправки задания: 20.11.2021

**Общая постановка задачи**

Разработать детальные требования, тест план и написать программу для решения следующей задачи:

1.Напишите функцию, выполняющую проверку соответствия строки на соответствие выражению.

Функция должна возвращать значение TRUE, если строка соответствует правилу, и FALSE, если строка НЕ соответствует правилу.

Замечания:

* Реализация каждого определения должно быть оформлено в виде отдельной функции.
* Для повторяющихся действий использовать только рекурсию.
* Тестовые наборы данных можно определить в функции main.

Вариант №6

Понятие вещественного числа 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> ::= + | -

1. Примеры входных и выходных данных

| Входные данные | Выходные данные |
| --- | --- |
| ("+2.5E-123",0) | true |
| ("2.5E-123", 0) | true |
| ("+234.2225E-12333", 0) | true |
| ("+50.2124E-23", 0) | true |
| ("+.5E-123", 0) | true |
| ("+2.5E123", 0) | true |
| ("+2.5E-g23", 0) | false |
| ("+25-123", 0) | false |
| ("+2.5123", 0) | false |

3. Реализация

// recurse.h

#ifndef REC\_H

#define REC\_H

namespace recurse {

bool isRealNumber(const char\* str, int i);

}

#endif

// recurse.cpp

#include"recurse.h"

#include <iostream>

bool isSign(const char\* str, int& i) {

return str[i] == '+' || str[i] == '-';

}

bool UInt(const char\* str, int& i) {

return std::isdigit(str[i]) && std::isdigit(str[i + 1]) ? UInt(str, ++i) : std::isdigit(str[i]);

}

bool isMantissa(const char\* str, int& i) {

return UInt(str, i) ? str[++i] == '.' && UInt(str, ++i)

: str[i] == '.' && UInt(str, ++i);

}

bool isExhibitor(const char\* str, int& i) {

return str[++i] == 'E' && (isSign(str, ++i) ? i++ : i) && UInt(str, i);

}

bool recurse::isRealNumber(const char\* str, int i) {

(isSign(str, i)) ? i++ : i;

return isMantissa(str, i) && isExhibitor(str, i);

}

// main.cpp

#include <iostream>

#include "recurse.h"

int main()

{

std::cout << recurse::isRealNumber("+2.5E-123", 0) << std::endl;

std::cout << recurse::isRealNumber("2.5E-123", 0) << std::endl;

std::cout << recurse::isRealNumber("+234.2225E-12333", 0) << std::endl;

std::cout << recurse::isRealNumber("+50.2124E-23", 0) << std::endl;

std::cout << recurse::isRealNumber("+.5E-123", 0) << std::endl;

std::cout << recurse::isRealNumber("+2.5E123", 0) << std::endl;

std::cout << recurse::isRealNumber("+2.5E-g23", 0) << std::endl;

std::cout << recurse::isRealNumber("+25-123", 0) << std::endl;

std::cout << recurse::isRealNumber("+2.5123", 0) << std::endl;

return 0;

}