

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра ФКТИ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Рекурсия**

Студент гр. 9303

\_\_\_\_\_

Ефимов М. Ю

Преподаватель

\_\_\_\_\_

Филатов А. Ю.

Санкт-Петербург

2020

### **Цель работы.**

При освоении этого раздела студент должен познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

### **Основные теоретические положения.**

Рекурсивным называется объект, содержащий сам себя или определенный с помощью самого себя. Мощность рекурсии связана с тем, что она позволяет определить бесконечное множество объектов с помощью конечного высказывания. Точно так же бесконечные вычисления можно описать с помощью конечной рекурсивной программы. Рекурсивные алгоритмы лучше всего использовать, когда решаемая задача, вычисляемая функция или обрабатываемая структура данных определены с помощью рекурсии. Если процедура (функция)  $P$  содержит явное обращение к самой себе, она называется прямо рекурсивной. Если  $P$  содержит обращение к процедуре (функции)  $Q$ , которая содержит (прямо или косвенно) обращение к  $P$ , то  $P$  называется косвенно рекурсивной. Многие известные функции могут быть определены рекурсивно. Например факториал, который присутствует практически во всех учебниках по программированию, а также наибольший общий делитель, числа Фибоначчи, степенная функция и др

### **Задание.**

#### **Вариант 7.**

Построить синтаксический анализатор для понятия вещественное число.

вещественное\_число ::= целое\_число . целое\_без\_знака |  
целое\_число.целое\_без\_знака | целое\_число | целое\_число  
целое\_без\_знака ::= цифра | цифра целое\_без\_знака  
целое\_число ::= целое\_без\_знака | + целое\_без\_знака | - целое\_без\_знака

## **Выполнение работы.**

`bool integer(std::string& str, int* count, int* depth)` – прямо рекурсивная функция. Принимает на вход указатель на строку, указатель на целое число `count` – текущий индекс, и `depth` – глубина рекурсии. Возвращает `bool err` – была ошибка или нет.

`bool cleverInteger(std::string& str, int* count, int* depth)` – функция для пропуска знаков `+` и `-`. Вызывает функцию `integer`.

`bool realNumber(std::string& str, int* count)` – проверка на вещественное число. Вызывает функции `integer` и `cleverInteger`.

Служит для пропуска знаков `'.'` и `'E'`.

`void printDepth(int depth)` – служит для правильного вывода глубины рекурсии.

Входные данные читаются из файла `test.txt`. Переменные и итоговый результаты выводятся на консоль.

Разработанный код смотреть в приложении А.

Результаты тестирования смотреть в Приложении В.

## **Выводы.**

Была написана программа определяющая строку как вещественное число. Программа использует рекурсивные функции. Входные данные считываются с файла файла. Промежуточные данные, в том числе глубина рекурсии выводятся на консоль.

## ПРИЛОЖЕНИЕ А.

Код файла main.cpp: #include <iostream>

#include <string>

#include <fstream>

/\*вещественное\_число::= целое\_число . целое\_без\_знака |

целое\_число.целое\_без\_знакаЕцелое число |/

целое\_числоЕцелое\_число

целое\_без\_знака::=цифра | цифра целое\_без\_знака

целое\_число::=целое\_без\_знака | + целое\_без\_знака | -целое\_без\_знака

\*/

void printDepth(int depth)

{

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

std::cout<<" ";

std::cout << depth <<std::endl;

}

bool integer(std::string& str, int\* count, int\* depth)

{

if (\*depth == 0) std::cout<< "Начало рекурсии"<<std::endl;

\*depth += 1;

printDepth(\*depth);

if (\*count >= str.length() - 1)

{

bool err = (char)str[\*count] > (char)'0' && str[\*count] < (char)'9';

std::cout << "Конец рекурсии"<< std::endl;

return err;

}

if ((char)str[\*count] == '.' || (char)str[\*count] == 'E')

```

    {
        if (*count != 0 && ((char)str[*count - 1] != '.' && (char)str[*count - 1] !=
'E'))
        {
            std::cout << "Конец рекурсии"<< std::endl;
            return true;
        }
        else
        {
            std::cout << "Конец рекурсии"<< std::endl;
            return false;
        }
    }
    bool err = (char)str[*count] > (char)'0' && str[*count] < (char)'9';
    *count += 1;
    if (err == false) std::cout << "Конец рекурсии"<< std::endl;
    return err && integer(str, count, depth);
}

bool cleverInteger(std::string& str, int* count, int* depth)
{
    if ((char)str[*count] == '-' || (char)str[*count] == '+')
        *count += 1;
    return integer(str, count, depth);
}

bool realNumber(std::string& str, int* count)
{
    bool x;
    int depth = 0;
    x = cleverInteger(str, count, &depth);

```

```

if (str[*count] == '.')
{
    depth = 0;
    *count += 1;
    x = x && integer(str, count, &depth);
}
if (str[*count] == 'E' && x)
{
    depth = 0;
    *count += 1;

    x = x && cleverInteger(str, count, &depth);
}
return x;
}

```

```

int main()
{
    using namespace std;
    setlocale(LC_ALL, "Russian");
    ifstream fin;
    string str;
    fin.open("test.txt");
    if (!fin.is_open())
    {
        std::cout << "Ошибка открытия файла";
        return 0;
    }
    getline(fin, str);

```

```
    fin.close();  
    if (str.length() == 0) { cout << "Пустая строка"; return 0; }  
    int lin = 0;  
    if (realNumber(str, &lin) && lin == (str.length() - 1))  
        cout << ("Это вещественное число");  
    else cout << "Это не вещественное число";  
    return 0;  
}
```

## ПРИЛОЖЕНИЕ В.

Результаты тестирования:

№	Ввод	Вывод
1	Пустая строка	error:Пустая строка
2	Maksimyss222	Начало рекурсии 1 Конец рекурсии Это не вещественное число
3		Начало рекурсии 1 2 3 4 5 6 7 8 9 10 11 12 13 Конец рекурсии Это вещественное число



4		<p>Начало рекурсии</p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p> <p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>Конец рекурсии</p> <p>Начало рекурсии</p> <p>1</p> <p>2</p> <p>3</p> <p>Конец рекурсии</p> <p>Начало рекурсии</p> <p>1</p> <p>Конец рекурсии</p> <p>Это вещественное число</p>
---	--	---

5	1231241241344.32E-2.3	1 2 3 4 5 6 7 8 9 10 11 12 13 14 Конец рекурсии Начало рекурсии 1 2 3 Конец рекурсии Начало рекурсии 1 2 Конец рекурсии Это не вещественное число
---	-----------------------	---