

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

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

Студентка гр. 8381

Ивлева О. А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

## Цель работы.

Познакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++. Разработать программу, использующую рекурсию.

## Задание.

Построить синтаксический анализатор для понятия вещественное число.

```
вещественное_число ::= целое_число . целое_без_знака |  
                    целое_число . целое_без_знака Е целое_число |  
                    целое_число Е целое_число  
целое_без_знака ::= цифра | цифра целое_без_знака  
целое_число ::= целое_без_знака | + целое_без_знака | - целое_без_знака
```

## Основные теоретические положения.

Любая функция (метод) в своем теле может вызывать сама себя. *Рекурсия* – это такой способ определения функции, при котором результат возврата из функции для данного значения аргумента определяется на основе результата возврата из той же функции для предыдущего (меньшего или большего) значения аргумента.

Если функция (метод) вызывает сам себя, то такой вызов называется *рекурсивным вызовом* функции. При каждом рекурсивном вызове запоминаются предыдущие значения внутренних локальных переменных и полученных параметров функции. Чтобы следующий шаг рекурсивного вызова отличался от предыдущего, значение как-минимум одного из параметров функции должно быть изменено. Остановка процесса рекурсивных вызовов функции происходит, когда изменяемый параметр достиг некоторого конечного значения, например, обработан последний элемент в массиве.

Рекурсивное обращение к функции может быть осуществлено, если алгоритм определен рекурсивно.

Чтобы циклический процесс преобразовать в рекурсивный, нужно уметь определить (выделить) три важных момента:

- условие прекращения последовательности рекурсивных вызовов функции
- формулу следующего элемента или итератора, который используется в рекурсивном процессе
- список параметров, которые передаются в рекурсивную функцию. Один из параметров обязательно есть итератор (счетчик) изменяющий свое значение.

### **Выполнение работы.**

Написание работы производилось на базе операционной системы Windows 10 в среде разработки CLion. Сборка, отладка и тестирование также производились в CLion.

Программе передается строка. Далее происходит объявление переменной *k*, в которую будет записано значение после выполнения рекурсивного алгоритма. Если вернется 0, то было передано вещественное число, иначе число не подходит под описание.

Строка передается в функцию `integ()`, которая проверяет является ли число целым(`целое_число::=целое_без_знака | + целое_без_знака | -целое_без_знака`), если первый символ удовлетворяет условию, то вызывается функция `uns_int()`, которая проверяет на число без знака(`целое_без_знака::=цифра | цифра целое_без_знака`). Если символ является числом, точкой или *E*, то идет проверка следующего символа, возвращаемое значение в таком случае будет 0. Если не удовлетворяет условию, то возвращается 1. Если встречается символ переноса строки, т.е. условие выхода и ошибки не обнаружены, то возвращается всегда 0. Также идет проверка не встретилось где-нибудь две точки или *E*, для этого созданы счетчики, которые при встрече данных символов увеличиваются, и если обнаруживается ещё один такой символ, то возвращается код ошибки 1.

### **Пример работы.**

Пример строки	Результат
3.3E+32	Hi, it is integ Hi, it is + - isdig integ Now:3 Hi, it is uns_int Hi, it is . uns_int Now:. Hi, it is uns_int Hi, it is isDig uns_int Now:3 Hi, it is uns_int Hi, it is E uns_int Now:E Hi, it is integ Hi, it is + - isdig integ Now:+ Hi, it is uns_int Hi, it is isDig uns_int Now:3 Hi, it is uns_int Hi, it is isDig uns_int Now:2 Yes
123EE+123	Hi, it is integ Hi, it is + - isdig integ Now:1 Hi, it is uns_int Hi, it is isDig uns_int Now:2 Hi, it is uns_int Hi, it is isDig uns_int Now:3 Hi, it is uns_int Hi, it is E uns_int Now:E Hi, it is integ NO

### Оценка эффективности алгоритма.

Алгоритм, реализованный в программе, имеет линейную зависимость от длины строки, то есть сложность оценивается как  $O(n)$ .

## **Выводы.**

В ходе выполнения лабораторной работы был изучен такой вид программ, как синтаксические анализаторы. Была реализована программа, которая анализирует строку рекурсивным методом, определяя соответствие определению.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <cstdio>
#include <cctype>
#include <string>
#include <fstream>
#include <iostream>
using namespace std;

int uns_int(string, int, int);
int integ(string, int, int);
int z = 0;
int point = 0;
int e = 0;

int uns_int(string s, int kol, int start){
    if (start < kol) {
        cout << "Hi, it is uns_int" << endl;
        if (isdigit(s[start])) {
            cout << "Hi, it is isDig uns_int" << endl;
            return uns_int(s, kol, start + 1) + z;
        } else if (s[start] == '.') {
            cout << "Hi, it is . uns_int" << endl;
            if ((point == 0) && (start != (kol-1)) && (e != 1)){
                point = 1;
                return uns_int(s, kol, start + 1) + z;
            } else return 1;
        } else if (s[start] == 'E') {
            cout << "Hi, it is E uns_int" << endl;
            if ((e == 0) && (s[start-1] != '.')){
                e = 1;
                return integ(s, kol, start + 1) + z;
            } else return 1;
        } else if (isspace(s[start])) {
            cout << "Hi, it is isprint uns_int" << endl;
            if ((point != 0) || (e != 0))
                return 0;
            else {
                z = 1;
                return 1;
            }
        } else {
            cout << "Hi";
            z = 1;
            return 1;
        }
    }
    else return 0;
}

int integ(string s, int kol, int start){
    cout << "Hi, it is integ" << endl;
    int k = 0;
    if (s[start] == '+' || s[start] == '-' || isdigit(s[start])) {
```

```

        cout << "Hi, it is + - isdig integ" << endl;
        if ((s[start] == '+' || s[start] == '-') && (start == (kol-
1)) )
            return 1;
        else
            return uns_int(s, kol, start+1);
    }
    else return 1;
}

int main() {
    int f;
    int k;
    string mas;
    cout << "Do you want to read from file?" << endl;
    cin >> f;
    if (f == 1) {
        ifstream fin; // создаем объект класса ifstream (читать)
        fin.open("D:\\prog\\cpp\\lab1\\text.txt"); // открываем файл
        для считывания
        fin >> mas;
        if(!fin.is_open()) cout<<"ERROR";
        fin.close(); // закрываем файл
        ofstream fin2; //дозаписать
        fin2.open("D:\\prog\\cpp\\lab1\\text.txt", ios::app); //
        открываем файл и дозываетываем в него
        //cout << mas.length() << endl;
        k = integ(mas, mas.length(), 0);
        fin2 << endl;
        if ((k == 0) && mas.length() != 1)
            fin2 << "Yes";
        else
            fin2 << "NO";
        fin.close(); // закрываем файл
    }
    else{
        mas = "656.7896E+7678";
        //cout << mas.length();
        k = integ(mas, mas.length(), 0);
        if ((k == 0) && mas.length() != 1)
            cout << "Yes";
        else
            cout << "NO";
    }
    return 0;
}

```