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

ОТЧЕТ

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

Студент гр. 7381 _____ Ильясов А. В.

Преподаватель _____ Фирсов М. А.

Санкт-Петербург
2018

Цель работы

Ознакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций.

Задание

Требуется построить синтаксический анализатор для понятия *вещественное число*.

вещественное_число::= целое_число.целое_без_знака |
*целое_число.целое_без_знака*E*целое число |*
*целое_число*E*целое_число*
целое_без_знака::= цифра | цифра целое_без_знака
целое_число::= целое_без_знака | +целое_без_знака | - целое_без_знака

В работе используется язык программирования C.

Пояснение задания

Задача состоит в том, чтобы определить, удовлетворяет ли входная строка заданному понятию «вещественного числа», реализовав рекурсивную функцию проверки.

Описание алгоритма

На вход подается строка, которая разделяется на подстроки в местах, где встречаются символы «.» или «E». Далее эти подстроки распределяются в функции проверки на соответствие понятиям «*целое_без_знака*» либо «*целое_число*», в которых рекурсивно проверяется, состоят ли они из цифр.

Описание функций

void tabs(int deepCount) — функция печати отступов символами табуляции для демонстрации рекурсивных вызовов функций.

int deepCount — переменная, задающая количество требуемых символов табуляции для выделения уровня вложенности рекурсии.

Функция ничего не возвращает.

int isDigit(char symbol, int deepCount) — функция проверяющая, является ли символ symbol цифрой. Основана на стандартной функции isdigit.

char symbol — символ, который будет проверятся на соответствие символу цифры.

int deepCount — переменная, передаваемая в функцию tabs, которая вызывается внутри функции isDigit.

int isUnsignedInteger(char *string, int deepCount) — рекурсивная функция, проверяющая, удовлетворяет ли входная строка понятию «*целое_без_знака*».

Char *string — строка, которая проверяется на соответствие понятию «*целое_без_знака*».

int deepCount — переменная, передаваемая в функцию tabs, которая вызывается внутри функции isDigit.

Функция возвращает 1, если заданное условие выполняется(строка удовлетворяет понятию «*целое_без_знака*»), и 0, если условие не выполняется.

int isInteger(char *string, int deepCount) — функция, проверяющая, удовлетворяет ли строка string понятию «*целое_число*».

char *string - строка, которая проверяется на соответствие понятию «*целое_число*».

int deepCount — переменная, передаваемая в функцию tabs, которая вызывается внутри функции isDigit.

Функция возвращает 1, если заданное условие выполняется(строка удовлетворяет понятию *«целое_число»*), и 0, если условие не выполняется.

void isRealNumber(char *string) — первая функция, в которую попадает входная строка string. Внутри функции происходит разделение строки на

подстроки путем поиска с помощью стандартной функции strchr символов «.» или «E» и дальнейшей заменой их символа конца строки.

char *string — строка, введенная пользователем, которая будет проверена на удовлетворение заданного понятия *«вещественное_число»*.

Функция ничего не возвращает.

Тестирование

В качестве корректных входных данных были выбраны следующие примеры записи вещественных чисел:

123.456;

123E456;

123.456E789;

1E+456.

В качестве некорректных входных данных были выбраны следующие строки:

123456;

hello world.

Результаты тестирования сохранаются в файл testresult.txt.

Ниже представлена демонстрация тестирования программы с полным выводом для одного из тестов:

Входные данные	*Результат*
1E+456	Вызов функции isRealNumber для 1E+456: . Вызов функции isInteger для 1: . . Вызов функции isUnsignedInteger для 1: . . . Вызов функции isDigit для 1: . . . . Это цифра! . . . Завершение функции isDigit для 1. . . Завершение функции isUnsignedInteger для 1: Это целое без знака! . Завершение функции isInteger для 1: Это целое число! . Вызов функции isInteger для +456: . . Вызов функции isUnsignedInteger для 456: . . . Вызов функции isDigit для 4: . . . . Это цифра! . . . Завершение функции isDigit для 4. . . . Вызов функции isUnsignedInteger для 56: . . . . Вызов функции isDigit для 5: . . . . . Это цифра! . . . . Завершение функции isDigit для 5. . . . . Вызов функции isUnsignedInteger для 6: . . . . . Вызов функции isDigit для 6: . . . . . Это цифра! . . . . . Завершение функции isDigit для 6. . . . . Завершение функции isUnsignedInteger для 6: Это целое без знака! . . . Завершение функции isUnsignedInteger для 56: Это целое без знака! . . Завершение функции isUnsignedInteger для 456: Это целое без знака! . Завершение функции isInteger для +456: Это целое число! Завершение функции isRealNumber, результат: Это вещественное число!
123E456	Это вещественное число!
123.456E789	Это вещественное число!
123.456	Это вещественное число!
123456	Это не вещественное число!
hello world	Это не вещественное число!

Выводы.

В ходе выполнения лабораторной работы получены знания по теме «рекурсия», получены навыки в написании bash-скриптов и закреплены знания синтаксиса языка C.

ПРИЛОЖЕНИЕ А

КОД LAB1.C

```
#include <stdio.h>
#include <locale.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

void tabs(int deepCount) {                                                    //
Функция печати отступов
	for (int i = 0; i < deepCount; i++)
		printf(".\t");
}

int isDigit(char symbol, int deepCount) {

	tabs(deepCount);
	printf("Вызов функции isDigit для %c:\n", symbol);

	int interimResult = isdigit(symbol);

	tabs(deepCount+1);
	printf(interimResult ? "Это цифра!\n" : "Это не цифра!\n");
	tabs(deepCount);
	printf("Завершение функции isDigit для %c.\n", symbol);

	return interimResult;
}

int isUnsignedInteger(char *string, int deepCount) {

	tabs(deepCount);
	printf("Вызов функции isUnsignedInteger для %s:\n", string);

	int interimResult = 1;
	char *interimString = string;
	if (strlen(string) == 1) {                                                //
Если строка представляет собой один символ
		char symbol = string[0];
		interimResult = isDigit(symbol, deepCount+1);
	}
```

```
	else {                                                                  //
Если строка состоит из нескольких символов
		interimString = string;
		char symbol = string[0];
		string = string + 1;
		interimResult = isDigit(symbol, deepCount+1) &&
isUnsignedInteger(string, deepCount+1);
	}
	tabs(deepCount);
	printf("Завершение функции isUnsignedInteger для %s: ",
interimString);
	printf(interimResult ? "Это целое без знака!\n" : "Это не целое без
знака!\n");

	return interimResult;
}

int isInteger(char *string, int deepCount) {

	tabs(deepCount);
	printf("Вызов функции isInteger для %s:\n", string);

	int interimResult = 1;
	char *interimString = string;
	if (string[0] == '+' && string[0] == '-')
		string = string + 1;

	if (!strlen(string)) {
		interimResult = 0;
	}
	else {
		interimResult = isUnsignedInteger(string, deepCount+1);
	}

	tabs(deepCount);
	printf("Завершение функции isInteger для %s: ", interimString);
	printf(interimResult ? "Это целое число!\n" : "Это не целое число!\n");

	return interimResult;
}

void isRealNumber(char *string) {

	printf("Вызов функции isRealNumber для %s:\n", string);
```

```
        int result = 1;
        char *firstPart;                                                    //
Указатели
        char *secondPart;                                                   //
под части строки,
        char *thirdPart;                                                    //
потенциально представляющие целое или целое без знака

        if (strchr(string, '.')) {
                char *delimiter = strchr(string, '.');                      //
                firstPart = string;                                         //
                secondPart = delimiter+1;                                   //
                *delimiter = '\0';                                          //

                if (!isInteger(firstPart, 1)) {
                        result = 0;
                }
                else if (strchr(secondPart, 'E')) {
                        char *delimiter = strchr(secondPart, 'E');          //
                        thirdPart = delimiter+1;                            //
При обнаружении символов '.' или 'E'
                        *delimiter = '\0';                                  //
разделаем строку, заменяя эти символы на '\0'
                        result = isInteger(secondPart, 1) && isInteger(thirdPart, 1);
                //
                }
                else {
                        result = isInteger(secondPart, 1);
                }

        }
        else if (strchr(string, 'E')) {
                char *delimiter = strchr(string, 'E');                      //
                firstPart = string;                                         //
                secondPart = delimiter+1;                                   //
                *delimiter = '\0';                                          //
                result = isInteger(firstPart, 1) && isInteger(secondPart, 1);
        }
        else {                                                              // В
противном случае строка
                result = 0;                                                 // не
удовлетворяет заданному условию
        }
```

```
	printf("Завершение функции isRealNumber, результат: ");
	printf(result ? "Это вещественное число!\n\n" : "Это не вещественное
число!\n\n");
}

int main() {
	setlocale(LC_ALL, "Russian");
	char *input_data = malloc(sizeof(char) * 200);

	fgets(input_data, 200, stdin);
	if (input_data[strlen(input_data) - 1] == '\n')
		input_data[strlen(input_data) - 1] = '\0';

	isRealNumber(input_data);

	free(input_data);
	return 0;
}
```

ПРИЛОЖЕНИЕ Б
ФАЙЛ COMPILE.SH

```
#!/bin/bash
gcc Source/lab1.c -o lab1

echo "Для запуска с выводом на экран, используйте ./lab1"
echo "Для запуска с выводом в файл, используйте run.sh"
echo "После запуска сразу вводите входные данные."
```

ПРИЛОЖЕНИЕ В
ФАЙЛ RUN.SH

```
#!/bin/bash
echo -n > result.txt

./lab1 >> result.txt

echo "Результат записан в файл result.txt"
```

ПРИЛОЖЕНИЕ Г
ФАЙЛ RUNTESTS.SH

```
#!/bin/bash
echo -n > testsresult.txt

echo "Примеры корректных входных данных: "

echo "Тест1: 123.456"
echo "Тест 1: " >> testsresult.txt
./lab1 < Tests/test1.txt >> testsresult.txt

echo "Тест2: 123E456"
echo "Тест 2: " >> testsresult.txt
./lab1 < Tests/test2.txt >> testsresult.txt

echo "Тест3: 123.456E789"
echo "Тест 3: " >> testsresult.txt
./lab1 < Tests/test3.txt >> testsresult.txt

echo "Тест4: 1E+456"
echo "Тест 4: " >> testsresult.txt
./lab1 < Tests/test4.txt >> testsresult.txt

echo "Примеры некорректных входных данных: "

echo "Тест5: 123456"
echo "Тест 5: " >> testsresult.txt
./lab1 < Tests/test5.txt >> testsresult.txt

echo "Тест6: hello world"
echo "Тест 6: " >> testsresult.txt
./lab1 < Tests/test6.txt >> testsresult.txt

echo "Результаты тестов сохранены в файл testsresult.txt"
```