Тампио Илья Сергеевич 3530904/10002 Индивидуальное задание – Рекурсия - Вариант 5

Постановка задачи:

Напишите функцию, выполняющую проверку соответствия строки, заданной параметром,   
правилу, указанному в Вашем варианте.   
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 определить в отдельном файле.

Понятие выражение определено следующим образом:   
<выражение> ::= <терм> | <терм> +<выражение> | <терм> – <выражение>   
<терм> ::= <множитель> | <множитель> \* <терм>   
<множитель> ::= <целое без знака> | <идентификатор> | ( <выражение>)   
<целое без знака> ::= <цифра> | <цифра> <целое без знака>   
<идентификатор> ::= <буква>   
<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9   
<буква> ::= A | B | C | D | E

Детальные требования и тест-план:

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Ввод | | | |
| 1.2 Файл ввода | Файл доступен программе, иначе исключение | 1. Нет файла ввода: | 1. Сообщение: «"Файл ввода недоступен. Завершение программы..."» |
| 1. Тестирование | | | |
| Ожидаемый результат для соответствующих вводов:  FALSE 1A  FALSE B2  FALSE 1+  FALSE 1\_  FALSE 1\*  FALSE 1\*1\*  FALSE 1\*1А1\*1  FALSE \*1\*1  FALSE 1\*1-  FALSE 1\*1+  FALSE 1\*1\_  FALSE (  FALSE )  FALSE ()  FALSE (\_)  FALSE 1234)  FALSE 12)34)  FALSE (12)34)  FALSE (12+34)\*(34\*12)\_  FALSE (12+3\_4)\*(34\*12)  FALSE (12+34)\_\*(34\*12)  FALSE (12+34)\*(34\_\*12)  FALSE (12+34)\*(34\*\_12)  TRUE (12)\*(34)  TRUE (A)\*(B)  TRUE 0\*1  TRUE (1)  TRUE (A)  TRUE (1\*2)  TRUE ((1-2))\*(1+2)  TRUE 1  TRUE 2  TRUE 3  TRUE 1\*1  TRUE 1+1  TRUE A+1  TRUE B+1  TRUE B+B  TRUE 2-B  TRUE 9+0  TRUE 9-9-9  TRUE 333  TRUE 33+33  TRUE 99+99  TRUE A+B\*C+D\*E  TRUE 1111111111 | | | |

Код:

#include <iostream>

#include <fstream>

bool isDigitC(const char\*& arr) {

return \*arr >= '0' && \*arr <= '9';

}

bool isLetterC(const char\*& arr) {

return \*arr >= 'A' && \*arr <= 'E';

}

bool isIdentifierC(const char\*& arr) {

return isLetterC(arr);

}

bool isUIntM(const char\*& arr, bool flag=false)

{

if (isDigitC(arr))

{

arr++;

flag = true;

return isUIntM(arr, flag);

}

return flag;

}

bool isExpressionM(const char\*&, bool);

bool isMultiplierM(const char\*& arr) {

if (\*arr == '(') {

arr++;

if (isExpressionM(arr, true)) {

return \*(arr++) == ')';

}

}

if (isIdentifierC(arr)) {

arr++;

return true;

}

if (isUIntM(arr)) {

return true;

}

return false;

}

bool isTermM(const char\*& arr) {

if (isMultiplierM(arr)) {

if (\*arr == '\*') {

arr++;

return isTermM(arr);

}

else {

return true;

}

}

return false;

}

bool isExpressionM(const char\*& arr, bool braced=false) {

if (isTermM(arr)) {

if (\*arr == '+' || \*arr == '-') {

arr++;

return isExpressionM(arr, braced);

}

if (\*arr == ')' && braced) {

return true;

}

return \*arr=='\0';

}

return false;

}

int main() {

setlocale(LC\_ALL, "Russian");

const int MAX\_STR\_LENGTH = 100;

char\* strf = new char[MAX\_STR\_LENGTH];

std::ifstream ifs("input.txt");

try {

if (!ifs.good()) {

throw "Файл ввода недоступен. Завершение программы...";

}

}

catch (const char \* e){

std::cerr << e;

return 1;

}

std::cout << "Reading input from file...\n";

while (ifs) {

ifs.getline(strf, MAX\_STR\_LENGTH);

const char\* str = strf;

const char\* strBuf = str;

std::cout << (isExpressionM(str) ? "TRUE" : "FALSE") << '\t' << strBuf << "\n";

}

}