**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: **Динамические структуры данных.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Некрасов Н.А. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Цель:

Написать программу, которую реализует стек и проверяет на вилодность html-страницу.

Задание:

Требуется написать программу, получающую на вход строку, (без кириллических символов и не более 3000 символов) представляющую собой код "простой" html-страницы и проверяющую ее на валидность. Программа должна вывести correct если страница валидна или wrong.

html-страница, состоит из тегов и их содержимого, заключенного в эти теги. Теги представляют собой некоторые ключевые слова, заданные в треугольных скобках. Например, <tag> (где tag - имя тега). Область действия данного тега распространяется до соответствующего закрывающего тега </tag> который отличается символом /. Теги могут иметь вложенный характер, но не могут пересекаться

<tag1><tag2></tag2></tag1> - верно

<tag1><tag2></tag1></tag2> - не верно

Существуют теги, не требующие закрывающего тега.

Валидной является html-страница, в коде которой всякому открывающему тегу соответствует закрывающий (за исключением тегов, которым закрывающий тег не требуется)

Во входной строке могут встречаться любые парные теги, но гарантируется, что в тексте, кроме обозначения тегов, символы < и > не встречаются. аттрибутов у тегов также нет.

Теги, которые не требуют закрывающего тега: <br>, <hr>

Стек (который потребуется для алгоритма проверки парности тегов) требуется реализовать самостоятельно на базе массива.

Ход работы:

#define TAG 15

* Подключение необходимых библиотек. Подключаются библиотеки ввода/вывода, стандартных функций и работы со строкой.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

* Описание стека. Реализуется на основе массива для упрощения получения последнего элемента.

typedef struct Stack{ //Описание стека

char tag[200][TAG];

size\_t size;

} Stack\_t;

* Описание функций для работы со стеками. Функции: положить элемент, удалить элемент, проверить на пустоту.

const char \* pop(Stack\_t \*stack){ //Функция удаления верхнего эл-та

stack->size--;

return stack->tag[stack->size+1];

}

void push(Stack\_t \*stack, char \*tag\_push) { //Функция для записи эл-та в стек

stack->size++;

strncpy(stack->tag[stack->size], tag\_push,15);

}

int is\_empty(Stack\_t \* stack){ //

return stack->size;

}

* Тело программы. Происходит считывание строки, если начинается с “<”, то начинается считывание тега, если он начинается с “/”, то сравнивается с верхним элементом стека, иначе, проверяется на теги, не имеющие закрывающих, иначе записывается в стек. Если закончилась строка, но стек не пуст или открывающий не соответствует закрывающему тегу, если закрывающий первый выводится “wrong”, иначе “correct”.

int main(void) {

Stack\_t head;

char tag[TAG];

char tcheck[TAG];

head.size=-1;

char c;

while ((c = getchar()) != EOF ){ //Получаем символ

if (c == '<') {

scanf("%15[^>]",tag);

if (strcmp(tag,"br") && strcmp(tag,"hr")){ //Проверка на нез. теги

if(tag[0] != '/'){ //Проверка на закр. тег

push(&head, tag);

}else{

if (is\_empty(&head)==-1){ //Проверка на пустоту

printf("wrong");

return 0;

}

else{

strcpy(tcheck, pop(&head));

if(strcmp(tcheck,tag+1)!=0){

printf("wrong");

return 0;

}

}

}

}

}

}

if (is\_empty(&head)==-1)

printf("correct");

else

printf("wrong");

}

Вывод:

В данной лабораторной работе был реализован стек на основе массива, а также написана программа проверяющая валидность html-страницы.