**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 |  | Кринкин К.В. |

Санкт-Петербург

2017

Цель:

Написать программу, проверяющую валидность html-странички.

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:

* Подключение необходимых библиотек

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

* Функции для работы со стеком

void push(char \*\* stack, char \* element){ //Положить элемент

strcpy(stack[++flag],element);

}

void pop(char \*\* stack){ //Убрать элемент

flag--;

}

int is\_empty(){ //Проверка на пустоту

return (flag == -1) ? 0 : 1;

}

* Тело программы

int main(){

char c; //Объявление переменной через которую считываем текст

char tag[15]; //Буфер под теги

char \*\*stack;

stack = (char\*\*)malloc(100 \* sizeof(char\*)); //Выделяем помять под стек, каждый элемент которого - массив

for (int i = 0; i<100; i++)

stack[i] = (char\*)malloc(15 \* sizeof(char)); //Каждый член - тэг

while (c != '\n' ){ //До тех пор пока не конец строки

scanf("%c",&c); //Считываем c

if (c == '<') { //Если с = «<»

scanf("%15[^>]",tag); //То начинаем считывать в стэк до > (15 обусловлено тем, что самый длинный тэг 10 символов, но 5 на прозапас)

if (strcmp(tag,"hr") && strcmp(tag,"br")){ //Если это не br или hr

if (tag[0] != '/') //Если последний тег не закрывающий

push(stack,tag); // Кладём в стэк

else{ //Иначе

if (is\_empty() == 0){ // Если стек пуст

printf("wrong"); //То ошибка

return 0; //Выходим

}

else{ //Иначе

if (strcmp(stack[flag],tag+1)){ //если закрывающий отличен от нынешнего

printf("wrong");

return 0;

}

else

pop(stack);

}

}

}

}

}

if (!is\_empty()) //Конечное условие проверки

printf("correct"); //Если пусто, то все тэги закрыты

else

printf("wrong"); //А если нет, то не все

}

Вывод:

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