­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Кафедра САПР

Контрольна робота №2

Варіант №21

з курсу: “Основи Програмування”

для студентів базових напрямків 6.0804 “Комп`ютерні науки”

Виконав студент гр. КНз-11

Чалий Михайло

­­

Львів 2013

#include <stdio.h>

#include <math.h>

#include <string.h>­­

#include <stdlib.h>

typedef struct node Node;

struct node {

char \*word;

char \*type;

Node \*left;

Node \*right;

};

/\* Stack impl\*/

typedef struct

{

Node \*v[20];

int top;

} NodeStack;

void tryPush(NodeStack \*s, Node \*val){

if (val != NULL){

s->v[s->top] = val;

(s->top)++;

}

}

Node \*pop(NodeStack \*s){

(s->top)--;

return (s->v[s->top]);

}

void initStack(NodeStack \*s){

s->top = 0;

}

int isNotEmpty(NodeStack \*s){

return s->top;

}

Node \*createNode(char \*word, char \*type, Node \*left, Node \*right){

Node \*node = (Node \*)malloc(sizeof(Node));

node->word = word;

node->type = type;

node->left = left;

node->right = right;

return node;

}

void print(Node \*head){

// Traverse to print

NodeStack \*s = (NodeStack \*)malloc(sizeof(NodeStack));

Node \*current;

initStack(s);

tryPush(s, head);

while(isNotEmpty(s)){

current = pop(s);

tryPush(s, current->right);

tryPush(s, current->left);

if (current->word != ""){

printf(">%s<", current->word);

}

}

printf("\n");

free(s);

}

Node \*findParentOfNode(Node \*head, char \*word){

// Traverse to print

NodeStack \*s = (NodeStack \*)malloc(sizeof(NodeStack));

Node \*current = NULL;

initStack(s);

tryPush(s, head);

while(isNotEmpty(s)){

current = pop(s);

tryPush(s, current->right);

tryPush(s, current->left);

if ((current->right != NULL && current->right->word == word)

|| (current->left != NULL && current->left->word == word)){

break;

}

}

free(s);

return current;

}

int main(void){

Node \*n;

Node \*head = createNode("", "S",

createNode("Пливуть ", "V", NULL, NULL),

createNode("", "NP",

createNode("", "AP",

createNode("осінні ", "A", NULL, NULL),

createNode("тихі ", "A", NULL, NULL)),

createNode("небеса", "N", NULL, NULL)));

print(head);

// Modify tree

n = findParentOfNode(head, "тихі ");

n->right = createNode("білі ", "A", NULL, NULL);

print(head);

// TODO cleanup

}

https://github.com/chaliy/studies-octo-adventure/blob/master/lp/ap/lab9/l14\_21.c