**Отчет по лабораторной работе №24** по курсу практикум на ЭВМ

Студентка группы М8О-110Б-21 Агеева Алиса, № 2 по списку

Контакты www, e-mail, icq, skype

Работа выполнена: « » \_\_\_\_\_\_\_\_\_\_\_\_2022г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_2021 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.Тема:** Преобразование выражения в дерево

1. **2. Цель работы:** Составить программу выполнения заданных преобразований арифметических выражений с применением деревьев.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Задание** (*вариант №* *17* )**:**  Преобразовать a^n\*a^k в a^(n+k)

**4. Оборудование** (лабораторное):

ЭВМ Intel Pentium G2140, процессор 3.30 GHz , имя узла сети Cameron с ОП 8096 Мб, НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_AMD Ryzen 5 3500 с ОП 8096 Мб, НМД 131072 Мб. Монитор Huawei

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Программное обеспечение (лабораторное):**

Операционная система семейства Unix , наименование Ubuntu версия 18.15.0

интерпретатор команд bash версия 4.4.20

Система программирования CLion версия 2020.3

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы cat, gcc

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix , наименование Fedora версия 33

интерпретатор команд bash версия 5.0.17

Система программирования CLion версия 2020.3

Редактор текстов emacs версия 25.2.2 Утилиты операционной системы cat, gcc

Прикладные системы и программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Идея, метод, алгоритм** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями).

**7.Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

Тесты:

a^k\*a^n → a^(k+n)

b^(m+n)\*b^q → b^(m+n+q)

a^k\*b^n → a^k\*b^n

(a+q)^k\*(a+q)^n → (a+p)^(k+n)

(a^k(e+v) → ошибка

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

*Допущен к выполнению работы.* **Подпись преподавателя****\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

alice@aliceVirtualBox:~$ cat trees.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct tnode

{

char data[40];

struct tnode \*left, \*right;

};

typedef struct tnode \*node;

int op(char data[40]) {

if (data[0] == '^' || data[0] == '/' || data[0] == '\*' || data[0] == '+' || data[0] == '-') {

return 1;

}

return 0;

}

int priority(char c) {

switch (c) {

case '-' :

case '+' :

return 1;

case '\*' :

case '/' :

return 2;

case '^' :

return 3;

default:

return 100;

}

}

node MakeTree(char Expr[], int first, int last) {

int MinPrt, k, prt, nest = 0;

node Tree = (node) malloc(sizeof(node));

MinPrt = 100;

for (int i = first; i <= last; i++) {

if (Expr[i] == '(') {

nest++;

continue;

}

if (Expr[i] == ')') {

nest--;

continue;

}

if (nest > 0) {

continue;

}

prt = priority(Expr[i]);

if (prt <= MinPrt) {

MinPrt = prt;

k = i;

}

}

if (nest != 0) {

printf("Wrong expression\n");

exit(1);

}

if (MinPrt == 100) {

if (Expr[first] == '(' && Expr[last] == ')') {

free(Tree);

return MakeTree(Expr, first + 1, last - 1);

} else {

k = last - first + 1;

for (int i = 0; i < k; i++) {

Tree->data[i] = Expr[first + i];

}

Tree->data[k] = '\n';

Tree->left = NULL;

Tree->right = NULL;

return Tree;

}

}

Tree->data[0] = Expr[k];

Tree->data[1] = '\n';

Tree->left = MakeTree(Expr, first, k - 1);

Tree->right = MakeTree(Expr, k + 1, last);

return Tree;

}

node parent(node t, node son) {

if (t == NULL) {

return NULL;

}

if (t->left == son || t->right == son) {

return t;

}

node left = parent(t->left, son);

node right = parent(t->right, son);

return (left == NULL) ? right : left;

}

void print\_tree(node t, int x) {

if (t == NULL) {

return;

}

print\_tree(t->right, x + 1);

for (int i = 0; i < x; i++) {

printf("\t");

}

for (int i = 0; i <= 40; i++) {

if (t->data[i] != '\n') {

printf("%c", t->data[i]);

continue;

}

printf("\n");

break;

}

print\_tree(t->left, x + 1);

}

void print\_expression(node t) {

if (t == NULL) {

return;

}

if (op(t->data) && op(t->left->data) &&

(priority(t->left->data[0]) < priority(t->data[0]) || t->left->data[0] == '^' && t->data[0] == '^')) {

printf("(");

print\_expression(t->left);

printf(")");

} else {

print\_expression(t->left);

}

for (int i = 0; i <= 40; i++) {

if (t->data[i] != '\n') {

printf("%c", t->data[i]);

continue;

}

break;

}

if (op(t->data) && op(t->right->data) &&

(priority(t->right->data[0]) < priority(t->data[0]) || t->right->data[0] == '^' && t->data[0] == '^')) {

printf("(");

print\_expression(t->right);

printf(")");

} else {

print\_expression(t->right);

}

}

void transform(node T, node t) {

if (t == NULL) {

return;

}

if (t->data[0] == '\*' && t->right->data[0] == '^' && t->right->data[1] == '\n' && t->left->data[0] == '^' && t->left->left == t->right->left ) {

node a = t->left->left;

node k = t->left->right;

node n = t->right->right;

t->left = a;

t->right = (node) malloc(sizeof(node));

t->data[0] = '^';

t->right->data[0] = '+';

t->right->data[1] = '\n';

t->right->left = k;

t->right->right = n;

transform(T, parent(T, t));

}

transform(T, t->left);

transform(T, t->right);

}

int main() {

node t = NULL;

int choose, g = 1;

while (g) {

printf("1. Create tree\t 2. Transform expression\t 3. Print tree\t 4. Print expression\t 5.Exit\n");

scanf("%d", &choose);

switch (choose) {

case 1: {

printf("Input your expression\n");

char Expr[1000];

scanf("%s", Expr);

int n = 0;

while (Expr[n] != '\0') {

n++;

}

t = MakeTree(Expr, 0, n - 1);

parent(t, NULL);

break;

}

case 2: {

transform(t, t);

break;

}

case 3: {

print\_tree(t, 0);

break;

}

case 4: {

print\_expression(t);

printf("\n");

break;

}

case 5: {

g = 0;

break;

}

default: {

printf("Wrong answer\n");

}

}

}

return 0;

}

alice@alice~VirtualBox:~$ gcc trees.c

alice@alice~VirtualBox:~$ ./a.out

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

1

Input your expression

a^k\*a^n

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

^

a

\*

k

^

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

2

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

+

k

^

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

4

a^(k+n)

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

1

Input your expression

b^(m+n)\*b^q

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

q

^

b

\*

n

+

m

^

b

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

2

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

4

b^(m+n+q)

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

1

Input your expression

a^k\*b^n

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

^

b

\*

k

^

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

2

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

^

b

\*

k

^

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

4

a^k\*b^n

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

1

Input your expression

(a+q)^k\*(a+q)^n

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

^

q

+

a

\*

k

^

q

+

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

2

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

3

n

+

k

^

q

+

a

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

4

(a+q)^(k+n)

1. Create tree 2. Transform expression 3. Print tree 4. Print expression 5.Exit

1

(a^k(e+v)

Wrong expression

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные события(ошибки в сценарии и программе,нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Выводы:** В лабораторной работе я познакомлась с разными видами записи алгебраических выражений и деревом выражений.
2. Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_