Отчет по лабораторной работе № 24 по курсу
«Практикум на ЭВМ»

Студент группы М8О-105Б-21 Минеева Светлана Алексеевна, № по списку 14

Контакты e-mail: svetlana.mineewa2003@yandex.ru

Работа выполнена: «11» мая 2022 г.

Преподаватель: Вячеслав Константинович Титов каф. 805

Отчет сдан «11» мая 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Алгоритмы и структуры данных.

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

3. **Задание** (*вариант №* 36)**:** Подсчитать количество переменных, используемых в данном выражении.

4. **Оборудование** (лабораторное):
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____
_____

Оборудование ПЭВМ студента, если использовалось:
Процессор 2,9 GHz 2-ядерный процессор Intel Core i5 с ОП 8 Гб, НМД 500 Гб. Монитор 13,3-дюймовый (2560 x 1600).
Другие устройства _____
_____

5. **Программное обеспечение (лабораторное):**
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
_____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных_____
_____

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства UNIX, наименование Terminal версия 2.10
интерпретатор команд bash версия 3.2.
Система программирования _____версия _____
Редактор текстов emacs версия 27.2-2
Утилиты операционной системы cat, ls, cp, mv и другие
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)

Алгоритм работы:

Подсчет количества переменных в выражении осуществляется проходом по всему дереву. Переменная – буквенное значение, которому при решении выражения могут быть заданы различные числовые значение. Из этого следует, что мы ищем буквенное значение. Однако одна переменная может входить в выражение несколько раз, значит нужно ввести пустой массив из 26 элементов (строчные латинские буквы), в который будем добавлять встреченные буквы. Когда мы встречаем переменную, проверяем что её нет в массиве проходом по нему. Если её нет, мы прибавляем на единицу счетчик и добавляем эту букву в массив, иначе продолжаем движение дальше.

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

```
#include<stdio.h>
#include<stdlib.h>

typedef char tdata;

int i; char ch;

struct node;

typedef node * link;

struct node{
   tdata data;
   link left,right;}
*tree;

void printtree(link t){
   static int l=0;
   l++;
   if(t){
      printtree(t->right);
      for(i=0;i<l;i++)printf("    ");
      printf("\\__%c\n",t->data);
      printtree(t->left);}
   l--;}

int isAN(){
   return (ch>='a')&&(ch<='z')||(ch>='0')&&(ch<='9');}

int isN(char c){
   return ((c>='a')&&(c<='z'));}

link mknode(char c, link l, link r){
   link t=new node;
   t->data=c; t->left=l; t->right=r;
   return t;}

link expr();

link fact(){
   link t;
   scanf("%c",&ch);
   if(ch=='('){ t=expr();
      if(ch!=')') printf("ERROR: not )\n");}
   else if(isAN()) t=mknode(ch,0,0);
               else printf("ERROR: not AN\n");
  return t;}

link term(){
   link tm; int done; char ch1;
   tm=fact(); done=0;
   while((ch!='\n')&&(!done)){
     scanf("%c",&ch);
```

```
    if((ch=='*')||(ch=='/')){ch1=ch; tm=mknode(ch1,tm,fact());}
    else done=1;}
  return tm;}

link expr(){
  link ex; int done; char ch1;
  ex=term(); done=0;
  while((ch!='\n')&&(!done)){
    if((ch=='+')||(ch=='-')) {ch1=ch; ex=mknode(ch1,ex,term()); }
    else done=1;}
  return ex;}

void tree2expr(link tree){
  if(tree){
    if((tree->data=='+')||(tree->data=='-')) printf("(");
    tree2expr(tree->left);
    printf("%c",tree->data);
    tree2expr(tree->right);
    if((tree->data=='+')||(tree->data=='-')) printf(")");}}

int numvar(link tree, int *k, char *a, int n){
  if(tree){
    if(isN(tree->data)==1){ int c=0;
      for(int j=0;j<n;j++) if(a[j]==tree->data){c=1;break;}
      if(c==0){a[*k]=tree->data;*k+=1;}}
    numvar(tree->left,k,a,n);
    numvar(tree->right,k,a,n);}}

int main(){
  printf("Input expression:\n");
  tree=expr();
  printf("\n\n----------------------\n\n");
  int k=0; int n=26;
  char arr[26];
  numvar(tree,&k,arr,n); printf("Number of variables = %d",k);
  printf("\n\n----------------------\n\n");
  printtree(tree);
  printf("\n\n----------------------\n\n");
  tree2expr(tree);
  printf("\n\n----------------------\n\n");
  return 0;}
```

Тесты:

1) f+k/h-t/k
   Переменные: f, k, h, t
   Количество переменных = 4

2) a+b+c+d+e+f+g+h+i+j+k
   Переменные: a, b, c, d, e, f, g, h, I, j, k
   Количество переменных = 11

3) 2+4*k-5/t+t*k
   Переменные: k, t
   Количество переменных = 2

4) c*e+4-5*k+j/k-9*t+g*i*k
   Переменные: c, e, k, j, t, g, i
   Количество переменных = 7

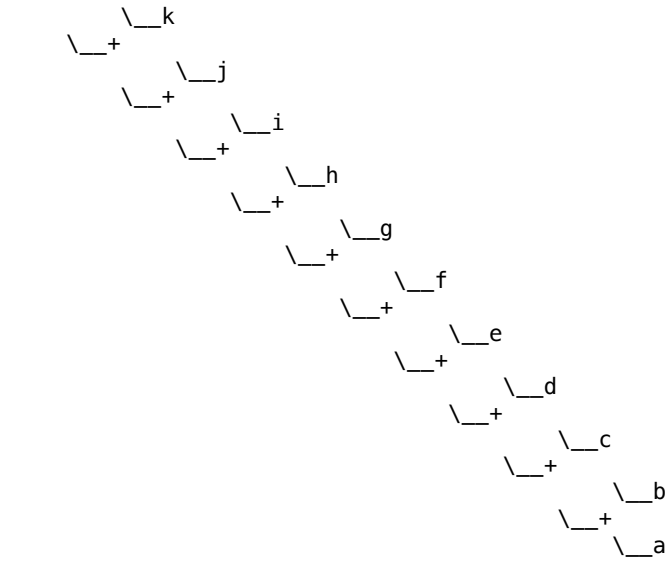
5) 4*e*a*b-c*4*g*(b+a)
   Переменные: e, a, b, c, g
   Количество переменных = 5

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

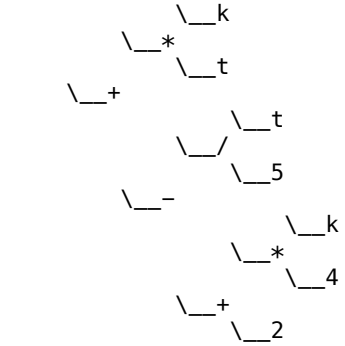
8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
Last login: Wed May 11 20:26:08 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro-MacBook:~ macbookpro$ cat zag.txt
**********************************
*   Минеева Светлана Алексеевна   *
*          М8О-105Б-21            *
*    Лабораторная работа №24      *
**********************************
MacBook-Pro-MacBook:~ macbookpro$ ls |tail -4
PycharmProjects
lab24.cpp
pslog_20200707_123036.log
zag.txt
MacBook-Pro-MacBook:~ macbookpro$ cat lab24.cpp
#include<stdio.h>
#include<stdlib.h>

typedef char tdata;

int i; char ch;

struct node;

typedef node * link;

struct node{
    tdata data;
    link left,right;}
*tree;

void printtree(link t){
    static int l=0;
    l++;
    if(t){
        printtree(t->right);
        for(i=0;i<l;i++)printf("    ");
        printf("\\__%c\n",t->data);
        printtree(t->left);}
    l--;}

int isAN(){
    return (ch>='a')&&(ch<='z')||(ch>='0')&&(ch<='9');}

int isN(char c){
    return ((c>='a')&&(c<='z'));}

link mknode(char c, link l, link r){
    link t=new node;
    t->data=c; t->left=l; t->right=r;
    return t;}

link expr();

link fact(){
    link t;
    scanf("%c",&ch);
    if(ch=='('){ t=expr();
        if(ch!=')') printf("ERROR: not )\n");}
    else if(isAN()) t=mknode(ch,0,0);
         else printf("ERROR: not AN\n");
   return t;}

link term(){
    link tm; int done; char ch1;
    tm=fact(); done=0;
    while((ch!='\n')&&(!done)){
       scanf("%c",&ch);
       if((ch=='*')||(ch=='/')){ch1=ch; tm=mknode(ch1,tm,fact());}
       else done=1;}
    return tm;}

link expr(){
    link ex; int done; char ch1;
```

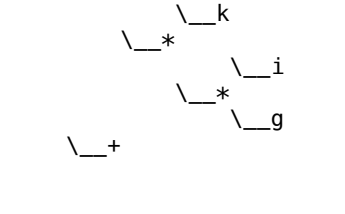
```
    ex=term(); done=0;
    while((ch!='\n')&&(!done)){
        if((ch=='+')||(ch=='-')) {ch1=ch; ex=mknode(ch1,ex,term()); }
        else done=1;}
    return ex;}

void tree2expr(link tree){
    if(tree){
        if((tree->data=='+')||(tree->data=='-')) printf("(");
        tree2expr(tree->left);
        printf("%c",tree->data);
        tree2expr(tree->right);
        if((tree->data=='+')||(tree->data=='-')) printf(")");}}

int numvar(link tree, int *k, char *a, int n){
    if(tree){
        if(isN(tree->data)==1){ int c=0;
            for(int j=0;j<n;j++) if(a[j]==tree->data){c=1;break;}
            if(c==0){a[*k]=tree->data;*k+=1;}}
        numvar(tree->left,k,a,n);
        numvar(tree->right,k,a,n);}}

int main(){
    printf("Input expression:\n");
    tree=expr();
    printf("\n\n-----------------------\n\n");
    int k=0; int n=26;
    char arr[26];
    numvar(tree,&k,arr,n); printf("Number of variables = %d",k);
    printf("\n\n-----------------------\n\n");
    printtree(tree);
    printf("\n\n-----------------------\n\n");
    tree2expr(tree);
    printf("\n\n-----------------------\n\n");
    return 0;}
MacBook-Pro-MacBook:~ macbookpro$ g++ -Wall -o lab24 lab24.cpp
MacBook-Pro-MacBook:~ macbookpro$ ls |tail -4
lab24
lab24.cpp
pslog_20200707_123036.log
zag.txt
MacBook-Pro-MacBook:~ macbookpro$ ./lab24
Input expression:
f+k/h-t/k


-----------------------

Number of variables = 4

-----------------------

            \__k
        \__/
            \__t
    \__-
               \__h
            \__/
               \__k
        \__+
            \__f


-----------------------

((f+k/h)-t/k)

-----------------------

MacBook-Pro-MacBook:~ macbookpro$ ./lab24
Input expression:
a+b+c+d+e+f+g+h+i+j+k


-----------------------

Number of variables = 11

-----------------------
```

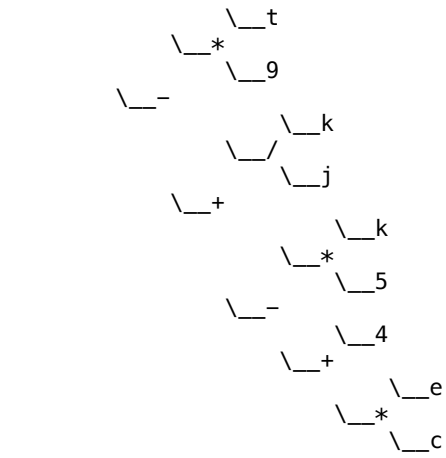



```
------------------------

((((((((((a+b)+c)+d)+e)+f)+g)+h)+i)+j)+k)

------------------------

MacBook-Pro-MacBook:~ macbookpro$ ./lab24
Input expression:
2+4*k-5/t+t*k


------------------------

Number of variables = 2

------------------------
```

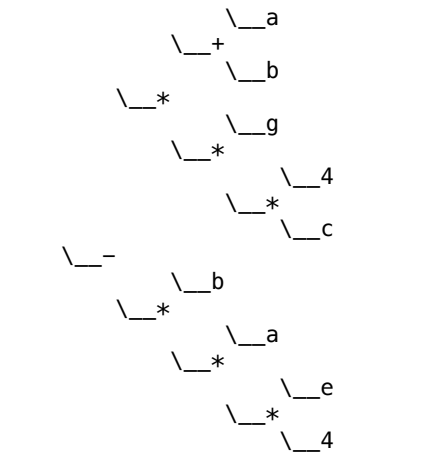



```
------------------------

(((2+4*k)-5/t)+t*k)

------------------------

MacBook-Pro-MacBook:~ macbookpro$ ./lab24
Input expression:
c*e+4-5*k+j/k-9*t+g*i*k


------------------------

Number of variables = 7

------------------------
```






```
-----------------------

(((((c*e+4)-5*k)+j/k)-9*t)+g*i*k)

-----------------------

MacBook-Pro-MacBook:~ macbookpro$ ./lab24
Input expression:
4*e*a*b-c*4*g*(b+a)


-----------------------

Number of variables = 5

-----------------------
```




```
-----------------------

(4*e*a*b-c*4*g*(b+a))

-----------------------

MacBook-Pro-MacBook:~ macbookpro$
```

9. 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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	Дом.	11.05.22	20:20	Неверный вывод количества переменных	Ошибка исправлена, введен массив, который контролирует повторение переменных	Одна переменная может несколько раз содержаться в выражении

10. Замечания автора по существу работы

Эта лабораторная работа очень полезна, она отлично развивает мышление и учит программированию на языке Си.

11. Выводы

Я составила программу выполнения заданных преобразований арифметических выражений с применением деревьев.

Недочёты при выполнении задания могут быть устранены следующим образом: больше практиковаться в написании программ на языке программирования Си.

Подпись студента Минеева С.А