**Отчет по лабораторной работе №23** по курсу\_\_\_\_\_\_1\_\_\_\_\_\_\_

Студент группы: **М8О-101Б-22**, **Соколова Виктория Дмитриевна**, № по списку: **20**, Контакты**,** [viktoriyasokolova04@gmail.com](mailto:viktoriyasokolova04@gmail.com) telegram **@Angrytea\_01** Работа выполнена: « 25 » апрель 2023г.

Преподаватель: **каф. 806 Крылов Сергей Сергеевич**, Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » 2023 г., итоговая оценка \_\_\_\_\_

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

1. **Тема:** Динамические структуры данных. Обработка деревьев.
2. **Цель работы:** Составить программу на языке Си для построения и обработки дерева, содержащего узлы типа float.
3. **Задание** (*вариант № 20* )**:** Определить глубину максимальной вершины дерева.
4. **Оборудование** (лабораторное):

ЭВМ , процессор , имя узла сети с ОП Мб, НМД Мб. Терминал адрес . Принтер Другие устройства

**Оборудование ПЭВМ студента, если использовалось:**

Процессор **Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz** с ОП **16384** Мб, ТТН **524 288** Мб. Встроенный монитор 15.6 дюймов диагональ, разрешение **1920 \* 1080**.

Другие устройства

1. **Программное обеспечение ЭВМ студента, если использовалось:**

Операционная система семейства **Linux** , наименование **Linux Ubuntu** версия 22.04.01 интерпретатор команд **bash** версия **5.1.16** Система программирования версия Редактор текстов **nano** версия **6.2** Утилиты операционной системы

Прикладные системы и программы

Местонахождение и имена файлов программ и данных на домашнем компьютере

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

1. typedef struct tree – задание узла, который будет иметь указатель на правое и левое поддерево и значение

2. node\_t \*create – создание корня узла

3. node\_t \*add – добавление узла в дерево

4. void maxuzel – нахождение глубины максимального узла

5. void PrintTree – вывод дерева на консоли

6. void minuzel – нахождение минимального по значению узла в правом поддереве для функции удаления

7. node\_t \*delete - удаление узла

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <string.h>
5. //представляем узел
6. typedef struct tree{
7. //значение в узле
8. float data;
9. //левое поддерево
10. struct tree \*left;
11. //правое поддерево
12. struct tree \*right;
13. }node\_t;
14. //создаем дерево
15. node\_t \*create(float data){
16. //память для корня дерева
17. node\_t \*tie = malloc(sizeof(node\_t));
18. //присваиваем значение
19. tie -> data = data;
20. //задаем узлы, чтобы позже наполнять
21. tie -> left = NULL;
22. tie -> right = NULL;
23. return tie;
24. }
25. //добавление узла
26. node\_t \*add(node\_t \*tie, float data){
27. if (tie == NULL){
28. return create(data);
29. }
30. if (data < tie -> data){
31. tie -> left = add(tie -> left, data);
32. }
33. else if (data > tie -> data){
34. tie -> right = add(tie -> right, data);
35. }
36. return tie;
37. }
38. void maxuzel(node\_t \*tie){
39. int l = 0;
40. if (tie == NULL){
41. printf("Derevo pystoe.......\n");
42. return;
43. }
44. while (tie -> right != NULL){
45. tie = tie -> right;
46. l++;
47. }
48. printf("%d\n", l);
49. }
50. void PrintTree(node\_t \*tie, int level) {
51. if (tie != NULL) {
52. PrintTree(tie->right, level + 1);
53. for (int i = 0; i < level; i++) {
54. printf("        ");
55. }
56. printf("%f\n", tie->data);
57. PrintTree(tie->left, level + 1);
58. }
59. }
60. float minuzel(node\_t \*tie){
61. tie = tie -> right;
62. while (tie -> left != NULL){
63. tie = tie -> left;
64. }
65. return tie -> data;
66. }
67. node\_t \*delete(node\_t \*tie, float data){
68. if (tie == NULL){
69. return tie;
70. } else if(data > tie -> data){
71. tie -> right = delete(tie -> right, data);
72. } else if (data < tie -> data){
73. tie -> left = delete(tie -> left, data);
74. } else if (tie -> left != NULL && tie -> right != NULL){
75. tie -> data = minuzel(tie);
76. tie -> right = delete(tie -> right, tie -> data);
77. } else{
78. if (tie -> left != NULL){
79. tie = tie -> left;
80. } else if (tie -> right != NULL){
81. tie = tie -> right;
82. } else {
83. tie = NULL;
85. }
86. }
87. return tie;
88. }
89. int main(){
90. char c[100];
91. int m = 1;
92. char proverka;
93. float data;
94. node\_t \*root = NULL;
95. while (m){
96. printf(" \n");
97. printf("1 - dobavit uzel\n");
98. printf("2 - delete uzel\n");
99. printf("3 - print derevo\n");
100. printf("4 - max uzel\n");
101. printf("5 - exit\n");
102. printf("Sho nada? ");
103. scanf("%99s", &c);
104. printf(" \n");
105. if (strcmp(c, "1") == 0){
106. printf("Vvedite znachenie uzla: ");
107. scanf("%f", &data);
108. root = add(root, data);
109. } else if (strcmp(c, "3") == 0) {
110. PrintTree(root, 0);
111. } else if(strcmp(c, "4") == 0){
112. maxuzel(root);
113. } else if(strcmp(c, "5") == 0){
114. break;
115. } else if (strcmp(c, "2") == 0){
116. printf("Shto udalyaem? ");
117. scanf("%f", &data);
118. root = delete(root, data);
119. } else {
120. printf("dyrachok\n");
122. }
124. }
125. }

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

1. **Распечатка протокола:**

PS C:\code\test> ./a

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 5

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 7

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 8

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 2

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? апрпв

dyrachok

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 9

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 1

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 3

9.000000

8.000000

7.000000

5.000000

2.000000

1.000000

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 5,5

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada?

dyrachok

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 1,2

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada?

dyrachok

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 3

9.000000

8.000000

7.000000

5.000000

2.000000

1.000000

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 1.2

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 1.5

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 1

Vvedite znachenie uzla: 5.1

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 3

9.000000

8.000000

7.000000

5.100000

5.000000

2.000000

1.500000

1.200000

1.000000

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 4

3

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 2

Shto udalyaem? 2

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 3

9.000000

8.000000

7.000000

5.100000

5.000000

1.500000

1.200000

1.000000

1 - dobavit uzel

2 - delete uzel

3 - print derevo

4 - max uzel

5 - exit

Sho nada? 5

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. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы
2. **Выводы**: Я составила программу двоичного дерева с функциями удаления узла, добавления узла, вывода дерева, поиска глубины максимального узла.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_