**Отчет по лабораторной работе № 23** по курсу

«Языки и методы программирования»

Студент группы М8О-112Б-21 Орешкин Максим Алексеевич, № по списку \_\_\_13\_\_

Контакты www, e-mail: maks-oreh03@mail.ru

Работа выполнена: « 18 » марта 2021 г.

Преподаватель: доцент каф. 806 \_Никулин С.П\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Отчет сдан « » \_\_\_\_\_\_\_\_\_202\_ г., итоговая оценка \_\_\_\_\_

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

1. **Тема:** Динамические структуры данных. Обработка деревьев.
2. **Цель работы:** Научиться создавать и работать с деревом.
3. **Задание (22):** Определить число вершин дерева.
4. **Оборудование** :

ЭВМ , процессор , имя узла сети с ОП ГБ,

НМД ГБ. Терминал адрес . Принтер

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

 Процессор intel core i7, с   ОП 8 ГБ, НМД  120832 МБ. Монитор  15,6 /1920\*1820

Другие устройства

1. **Программное обеспечение:**

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 Ubuntu версия 20.04

интерпретатор команд bash версия 4.4.18

Система программирования версия

Редактор текстов vim версия 8.1

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 постусловиями)

Каждая задача (добавление нового узла, текстовая визуализация, удаление узла, вычисление количества листьев) реализуется с помощью отдельных функций, каждая из которых допускает рекурсию и работает со структурным типом, который мы создали в начале для предстваления дерева в памяти устройства с помощью языка Си.

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

Файл “tree.h”

#ifndef \_TREE\_H\_

#define \_TREE\_H\_

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

typedef int TreeItem;

typedef struct \_tree \*Tree;

Tree tree\_create(TreeItem value);

void tree\_add\_node(Tree tree, TreeItem parent, TreeItem value);

void tree\_print(Tree tree);

void tree\_print\_node(Tree tree, int indent);

void tree\_destroy(Tree tree);

void tree\_del\_node(Tree tree, TreeItem value);

Tree tree\_find(Tree tree, TreeItem c);

void l(Tree tree, int \*count);

#endif // \_TREE\_H\_

Файл “tree1.c”

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include "tree1.h"

#define DEEP 0

#define max(x,y) ((x) > (y) ? (x) : (y))

struct \_tree {

TreeItem value;

Tree child;

Tree sibling;

};

Tree tree\_create(TreeItem value)

{

Tree tree = (Tree) malloc(sizeof(\*tree));

if(!tree) {

fprintf(stderr, "%s\n", "Error: no memory");

exit(1);

}

tree->value = value;

tree->child = NULL;

tree->sibling = NULL;

return tree;

}

void tree\_add\_node(Tree tree, TreeItem parent, TreeItem value)

{

Tree parent\_node = tree\_find(tree, parent);

if(parent\_node) {

if(!parent\_node->child) {

parent\_node->child = tree\_create(value);

} else {

parent\_node = parent\_node->child;

parent\_node->sibling = tree\_create(value);

}

} else {

printf("Указанная родительская вершина не найдена.\n");

}

}

Tree tree\_find(Tree tree, TreeItem c)

{

if(!tree) {

printf("Нет корня");

return NULL;

}

if(tree->value == c) {

return tree;

}

Tree result = NULL;

if(tree->child) {

result = tree\_find(tree->child, c);

if(result) return result;

}

if(tree->sibling) {

result = tree\_find(tree->sibling, c);

if(result) return result;

}

}

void tree\_del\_node(Tree tree, TreeItem value)

{

if(tree->child) {

if(tree->child->value == value) {

Tree tmp = tree->child;

tree->child = tree->child->sibling;

if (tmp->child) {

tree\_destroy(tmp->child);

}

free(tmp);

tmp = NULL;

return;

} else {

tree\_del\_node(tree->child, value);

}

}

if(tree->sibling) {

if(tree->sibling->value == value) {

Tree tmp = tree->sibling;

tree->sibling = tree->sibling->sibling;

if(tmp->child) {

tree\_destroy(tmp->child);

}

free(tmp);

tmp = NULL;

return;

} else {

tree\_del\_node(tree->sibling, value);

}

}

}

void tree\_destroy(Tree tree)

{

if(tree->child) {

tree\_destroy(tree->child);

}

if(tree->sibling) {

tree\_destroy(tree->sibling);

}

free(tree);

tree = NULL;

}

void tree\_print(Tree tree)

{

tree\_print\_node(tree, 0);

}

void tree\_print\_node(Tree tree, int indent)

{

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

printf("|");

if(i==indent-1) printf(">");

}

printf("%d\n", tree->value);

if(tree->child) {

tree\_print\_node(tree->child, indent + 1);

}

if(tree->sibling) {

tree\_print\_node(tree->sibling, indent);

}

}

void l(Tree tree, int \*count)

{

if(tree->child!=0) {\*count+=1; l(tree->child, count);}

if(tree->sibling!=0) {\*count+=1; l(tree->sibling, count);}

}

Файл 23.с

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "tree1.h"

int main()

{

Tree tree = NULL;

int root = 0, ver = 0, parent = 0, count= 0;

while(1)

{

printf("Введите число.\n1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход\n");

int n=0;

scanf("%d",&n);

switch(n){

case(1):

if(tree != NULL)

printf("Tree already exists\n");

else

{

printf("Введите значение корня дерева:\n");

scanf("%d", &root);

tree = tree\_create(root);

printf("Tree created\n");

}

break;

case(2):

printf("Введите родителя и ребенка\n");

scanf("%d%d", &parent, &ver);

tree\_add\_node(tree, parent, ver);

break;

case(3):

if(!tree) printf("Дерева не существует\n");

else

{

printf("Введите вершину, которую хотите удалить\n");

scanf("%d", &ver);

tree\_del\_node(tree, ver);

}

break;

case(4):

if (!tree) printf("Дерева не существует\n");

else

{

printf("\n\n");

tree\_print(tree);

printf("\n\n");

}

break;

case(5):

l(tree,&count);

printf("%d\n",count+1);

count=0;

break;

case(6):

tree\_del\_node(tree,root);

printf("End program\n");

exit(0);

break;

default:

printf("Некорректный ввод данных\n");

}

}

}

*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, подписанный преподавателем).

maxim@vb:~$ gcc 23\_1.c 23\_tree.c

maxim@vb:~$ ./a.out

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

1

Введите значение корня дерева:

1

Tree created

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

1

2

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

1

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

2

4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

3

5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|>3

||>5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|>3

||>5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

1

5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|>5

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

5

51

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

51

511

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

511

521

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|>5

||>51

|||>511

||||>521

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

4

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

4

41

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>3

|||>41

|>5

||>51

|||>511

||||>521

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

13

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

521

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>3

|||>41

|>5

||>51

|||>511

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

2

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>5

||>51

|||>511

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

16

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

^Z

[3]+ Stopped ./a.out

maxim@vb:~$ gcc 23\_1.c 23\_tree.c

maxim@vb:~$ ./a.out

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

1

Введите значение корня дерева:

1

Tree created

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

1

2

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

1

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

2

4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|>3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

4

41

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

4

42

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

42

421

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

3

31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

8

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

||>31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

31

321

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

322

4

Указанная родительская вершина не найдена.

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

||>31

|||>321

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

2

Введите родителя и ребенка

321

322

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

||>31

|||>321

||||>322

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

10

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

10

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

||>31

|||>321

||||>322

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

321

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>41

|||>42

||||>421

|>3

||>31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

8

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

41

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

||>4

|||>42

||||>421

|>3

||>31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

7

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

4

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>2

|>3

||>31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

2

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

|>3

||>31

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

3

Введите вершину, которую хотите удалить

3

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

4

1

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

5

1

Введите число.

1 Создание корня дерева 2 добавление элементов 3 удаление элементов 4 печать дерева 5 Подсчет вершин 6 Выход

6

End program

**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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 11. **Выводы**Научился работать с адресами и понял, как создавать деревья.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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