

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 3

з дисципліни “ Основи програмування ”

тема “Структури даних, функції, вказівники і файлові потоки.”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП- 61  Корунська Анна Михайлівна  (*прізвище, ім’я, по батькові*)  варіант № 17 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі** | **Оформлення звіту** | |  |  | | Нараховані бали:   |  |  |  | | --- | --- | --- | | **Корект. програм (2 бала)** | **Відп. на теор. питання (1 бал)** | **Відп. на прогр. питання (2 бала)** | |  |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2016

**Мета роботи**

Навчитися працювати зі структурами, файлами, динамічно виділяти пам’ять, а також вдосконалити та закріпити вміння роботи з вказівниками.

**Постановка завдання**

#### **Загальні вимоги до завдання:**

Після входу у програму користувач має можливість зробити вибір:

* Створити новий масив даних
* Зчитати масив даних із файлу

Після цього масив даних можна модифікувати:

* Видалити дані із вказаної позиції у масиві
* Перезаписати дані у вказаній позиції на нововведені
* Перезаписати тільки обране поле даних елемента на вказаній позиції

Також дозволити користувачу виконувати спеціальні команди за варіантом.

Дати можливість користувачу зберегти зміни даних на файлову систему, запропонувавши йому ввести назву дискового файлу, у який збережуться дані.

#### **Завдання №1. Розробка структур і функцій**

1. Описати нову струкутуру даних типу **"Курс лекцій"**. Визначити у структурі не менше 3-х полів різного типу, серед яких обов'язковими є: строка, ціле число, дробове число, **інша структура**.
2. Виконати декомпозицію завдання на окремі функції і описати їх прототипи у окремому header-файлі:
   1. Описати функції, одна з яких виконує запис даних із одної структури (через вказівник) у строку, та функцію, що виконує зворотнє перетворення.
   2. Описати функцію, що із тексту створює масив структур та зворотню функцію.
   3. Описати функцію, що видаляє структуру із масиву структур по індексу, функцію, що виконує перезапис структури по індексу даними із іншої структури.
   4. Описати функцію, що отримує вказівник на структуру, строку, що позначає поле у структурі та строку, що позначає нове значення вказаного поля у структурі і виконує перезапис поля заданої структури новим значенням.
   5. Описати функцію, що виконує дію над масивом вказівників на структури: **"Знайти всі курси лекцій, у яких загальна тривалість більша Х годин"**.
3. Реалізувати відповідні функції у окремому файлі з кодом.

#### **Завдання №2. Робота із файловими потоками**

1. Реалізувати функцію для зчитування даних із текстового файлу і перетворення їх у масив структур.
2. Реалізувати функцію для запису масиву структур у текстовий файл.

#### **Завдання №3. Тестування функцій**

1. Протестувати функції, що піддаються тестуванню, за допомогою assert() у окремій функції(-ях). Кожна функція має тестуватись мінімум на 5 тестових випадках.
2. Винести прототипи тестових функцій у окремий header-файл.
3. Винести тестові функції у окремий файл із кодом.

#### **Завдання №4. Розробка CUI-інтерфейсу**

1. Створити CUI - консольний інтерфейс користувача (просто взаємодія з користувачем через консоль) для використання описаного функціоналу.
2. Всі прототипи відповідних функцій CUI описати у окремому header-файлі, а їх реалізацію винести у окремий .c файл із кодом.
3. Тестові функції запускати тільки при заданні певного аргумента командного рядка.
4. У файлі із кодом функції main() не повинно бути жодних інших функцій.
5. Весь ввід користувача потрбно перевіряти на валідність щоби не допускати збоїв у роботі програми.

**Тексти коду програм**

**Завдання 1.**

|  |
| --- |
| main.c |
| **#include "lab\_3.h"**  **#include "CUI.h"**  **int main(void){**  **mainMenu();**  **return 0;**  **}** |

**Завдання 2.**

|  |
| --- |
| test.c |
| **#include "lab\_3.h"**  **#include "func.h"**  **#include "test.h"**  **#include <assert.h>**  **void tests(void){**  **testCopyStruct();**  **testRefillParam();**  **testStructToStr();**  **}**  **void testCopyStruct(void){**  **struct lectureCourse oldStr = {"Bla", "", "mama", "no", 45, 9, 5.0};**  **struct lectureCourse newStr;**  **copyStruct(&newStr, &oldStr);**  **assert(strcmp(oldStr.courseName, oldStr.courseName ) == 0);**  **assert(strcmp(oldStr.lecturer.surname, oldStr.lecturer.surname ) == 0);**  **assert(newStr.lecturer.age == 45);**  **assert(newStr.rating == 5.0);**  **assert(newStr.length == oldStr.length);**  **}**  **void testRefillParam(void){**  **struct lectureCourse test;**  **char cname [300] = "some course name";**  **refillParam(&test, 1, cname);**  **assert(strcmp(test.courseName, cname) == 0);**  **char age [300] = "50";**  **refillParam(&test, 5, age);**  **assert(test.lecturer.age == 50);**  **char empty [300] = "";**  **refillParam(&test, 4, empty);**  **assert(strcmp(test.lecturer.middlen, empty) == 0);**  **char numbers [300] = "123";**  **refillParam(&test, 2, numbers);**  **assert(strcmp(test.lecturer.surname, numbers) == 0);**  **char letters [300] = "abc";**  **refillParam(&test, 6, letters);**  **assert(test.length == 0);**  **}**  **void testStructToStr(void) {**  **struct lectureCourse test1 = { "Електротехніка", "Бужко", "Олена", "Миколаївна", 51, 48, 4.8};**  **char str1 [1000];**  **structToStr(&test1, str1);**  **assert(strcmp(str1, "Електротехніка Бужко Олена Миколаївна 51 48 4.800000") == 0);**  **struct lectureCourse\* test2 = NULL;**  **char str2 [1000];**  **structToStr(test2, str2);**  **assert(strcmp(str2, "") == 0);**  **}** |

**Завдання 3.**

|  |
| --- |
| GUI.c |
| **#include "lab\_3.h"**  **#include "CUI.h"**  **#include "func.h"**  **#include "test.h"**  **void drawBg (void) {**  **int x;**  **int y;**  **conReset();**  **for (x = 0; x <= 40; x++) {**  **for (y = 0; y <= 144; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **}**  **void drawBox(struct menuBox box){**  **int x;**  **int y;**  **drawBg();**  **conSetAttr(BG\_CYAN);**  **for (x = box.top; x <= box.head; x++) {**  **for (y = box.left; y <= box.right; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conSetAttr(BG\_WHITE);**  **for (x = box.head; x < box.down; x++) {**  **for (y = box.left; y <= box.right; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conSetAttr(BG\_CYAN);**  **for (x = box.down; x <= box.bottom; x++) {**  **for (y = box.left; y <= box.right; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **for (x = box.head; x < box.down; x++) {**  **conMove(x, box.right);**  **putchar(' ');**  **conMove(x, box.left);**  **putchar(' ');**  **}**  **conReset();**  **puts("");**  **}**  **void drawDownBox(struct menuBox box){**  **int x;**  **int y;**  **conSetAttr(BG\_CYAN);**  **for (x = box.down; x <= box.bottom; x++) {**  **for (y = box.left; y <= box.right; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **}**  **void pEnterOperationCode(struct menuBox box){**  **conSetAttr(BG\_CYAN);**  **conMove(box.bottom -1, box.left);**  **printf("\t Введіть номер операції: ");**  **}**  **void printMainMenu(struct menuBox box) {**  **drawBox(box);**  **conSetAttr(BG\_CYAN);**  **conMove((box.top + box.head)/2, box.left + 17);**  **printf("Лабораторна робота № 3");**  **conSetAttr(BG\_WHITE);**  **conSetAttr(FG\_BLACK);**  **conMove(box.head + 1, box.left + 3);**  **printf("\t 1. Видалити дані із вказаної позиції у масиві \n\t \**  **2. Перезаписати дані у вказаній позиції на нововведені \n\t 3. Перезаписати\**  **тільки обране поле даних елемента на \n\t вказаній позиції \n\t 4. Знайти всі\**  **курси лекцій, у яких загальна тривалість \n\t більша Х годин \n\t 5.\**  **Представити структуру у вигляді строки");**  **conMove(box.down - 3, 1);**  **printf("\tНатисніть t для виклику тестового режиму \n\tНатисніть s для збереження масиву \**  **у файл\n\tНатисніть q для виходу");**  **conReset();**  **pEnterOperationCode(box);**  **}**  **int printGB(struct menuBox box){**  **int x;**  **int y;**  **conReset();**  **for (x = 0; x <= 40; x++) {**  **for (y = 0; y <= 140; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conSetAttr(BG\_CYAN);**  **for (x = box.top; x <= box.bottom; x++) {**  **for (y = box.left; y <= box.right; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conMove((box.top + box.bottom)/2, box.left + 20);**  **printf("That's all)");**  **conReset();**  **conMove(box.bottom + 2, 0);**  **conShowCursor();**  **return 0;**  **}**  **void mainMenu(void) {**  **struct menuBox box = {4, 32, 4, 70, 7, 20};**  **int exit = 1;**  **struct lectureCourse\* pArr = NULL;**  **int quan = 0;**  **drawBox(box);**  **pArr = startMenu(box, &quan);**  **printMainMenu(box);**  **while(exit) {**  **char ch;**  **ch = conGetChar();**  **printf("%c", ch);**  **int check = checkSrt(box, pArr);**  **if(check){**  **conEmptyStr(box);**  **exit = printGB(box);**  **}**  **if(isalpha(ch)){**  **if('q' == ch){**  **free(pArr);**  **exit = printGB(box);**  **break;**  **}**  **if('t' == ch){**  **drawDownBox(box);**  **tests();**  **conMove(box.down + 1, box.left);**  **printf("\t Нічого не відбулося, отже все працює");**  **pEnterOperationCode(box);**  **}**  **if('s' == ch){**  **int status = conSaveToFile(box, pArr, quan);**  **if(status){**  **printMainMenu(box);**  **conMove(box.bottom, box.left);**  **printf("\t Зміни успішно збережені ");**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t На жаль, при збереженні файлу виникла помилка ");**  **}**  **pEnterOperationCode(box);**  **}**  **} else if(isdigit(ch)){**  **if('1' == ch) {**  **pArr = conDeleteStruct(box, pArr, &quan);**  **printMainMenu(box);**  **int check = checkSrt(box, pArr);**  **if(check){**  **conEmptyStr(box);**  **exit = printGB(box);**  **} else {**  **pEnterOperationCode(box);**  **}**  **}**  **if('2' == ch) {**  **drawParametrsList(box);**  **conRefillStruct(box, pArr, quan);**  **drawDownBox(box);**  **pEnterOperationCode(box);**  **}**  **if('3' == ch){**  **drawParametrsList(box);**  **conRefillParam(box, pArr, quan);**  **drawDownBox(box);**  **pEnterOperationCode(box);**  **}**  **if('4' == ch){**  **conSearchStr(box, pArr, quan);**  **pEnterOperationCode(box);**  **}**  **if('5' == ch){**  **conStructToStr(box, pArr, quan);**  **pEnterOperationCode(box);**  **}**  **} else {**  **drawBox(box);**  **printMainMenu(box);**  **conMove(box.bottom, box.left);**  **printf("\t Неправильно введений номер операції");**  **pEnterOperationCode(box);**  **}**  **}**  **}**  **struct lectureCourse\* startMenu(struct menuBox box, int\* quan){**  **int exit = 1;**  **struct lectureCourse\* pArr = NULL;**  **drawStartMenu(box);**  **while(exit) {**  **char ch;**  **ch = conGetChar();**  **printf("%c", ch);**  **if(isdigit(ch)){**  **if('1' == ch) {**  **int flag = 1;**  **while(flag){**  **conMove(box.bottom -2, box.left);**  **printf(" Введіть кількість масивів, які ви бажаєте створити: ");**  **\*quan = getInt();**  **if(flag > 0){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Введена кількість масивів є невірною. Спробуйте ще раз.");**  **}**  **}**  **pArr = makePArr(\*quan);**  **drawDownBox(box);**  **conFillArr(pArr, \*quan, box);**  **exit = 0;**  **}**  **if('2' == ch) {**  **char FileName [50];**  **int flag = 1;**  **drawDownBox(box);**  **drawFileInstructions(box);**  **while(flag){**  **conMove(box.bottom -2, box.left);**  **printf(" Введіть кількість масивів, яка має бути зчитана з файлу: ");**  **\*quan = getInt();**  **if(flag > 0){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Введена кількість масивів є невірною. Спробуйте ще раз.");**  **}**  **}**  **conMove(box.bottom - 1, box.left);**  **printf(" Введіть назву файлу, з якого будуть считані дані: ");**  **scanf("%s", FileName);**  **pArr = makePArr(\*quan);**  **int status = readFromFile(pArr, \*quan, FileName);**  **if(status) {**  **conMove(box.bottom, box.left);**  **printf("\t При зчитуванні даних виникла помилка");**  **}**  **exit = 0;**  **}**  **if('0' == ch || '3' == ch || '4' == ch || '5' == ch || '6' == ch \**  **|| '7' == ch || '8' == ch || '9' == ch) {**  **pEnterOperationCode(box);**  **}**  **} else {**  **drawBox(box);**  **drawStartMenu(box);**  **conMove(box.bottom, box.left);**  **printf("\t Неправильно введений номер операції");**  **pEnterOperationCode(box);**  **}**  **}**  **return pArr;**  **}**  **void drawStartMenu(struct menuBox box){**  **int shift = 1;**  **drawDownBox(box);**  **conMove(box.down + shift, box.left);**  **printf("\t Перш ніж почати працювати з масивом, його потрібно створити");**  **shift += 2;**  **conMove(box.down + shift, box.left);**  **printf("\t 1. Ввести масив даних з консолі \n\t 2. Зчитати масив з файлу");**  **pEnterOperationCode(box);**  **}**  **void drawFileInstructions (struct menuBox box) {**  **int x;**  **int y;**  **int wildth = 80;**  **int length = 17;**  **conSetAttr(BG\_BLACK);**  **for (x = box.head; x < box.head + length; x++) {**  **for (y = box.right + 5; y < box.right + 5 + wildth; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conMove(box.head + 1, box.right + 7);**  **printf("Для правильності зчитування дані в файлі мають бути записані ");**  **conMove(box.head + 2, box.right + 7);**  **printf("наступним чином: ");**  **conMove(box.head + 3, box.right + 7);**  **printf("Строка з назвою курсу, прізвище, ім'я, по-батькові, вік викладача,");**  **conMove(box.head + 4, box.right + 7);**  **printf("тривалість курсу у годинах, оцінка курсу.");**  **conMove(box.head + 5, box.right + 7);**  **printf("Приклад оформлення:");**  **conMove(box.head + 7, box.right + 7);**  **printf("Випічка та кондитерське мистецтво");**  **conMove(box.head + 8, box.right + 7);**  **printf("Марішина Вікторія Петрівна 27 80 4.5");**  **conMove(box.head + 10, box.right + 7);**  **printf("Наступні стуктури описувати без пропуску рядка");**  **conMove(box.head + 12, box.right + 7);**  **printf("Також можна скористатися уже готовими тестовими стуктурами,");**  **conMove(box.head + 13, box.right + 7);**  **printf("описаними в sample.txt");**  **conMove(box.head + 14, box.right + 7);**  **printf("Даний файл містить 5 тестових наборів даних для структур, ");**  **conMove(box.head + 15, box.right + 7);**  **printf("написаних українською мовою");**  **conSetAttr(BG\_CYAN);**  **}**  **void conFillArr(struct lectureCourse\* pArr, int quan, struct menuBox box){**  **struct lectureCourse\* lC = NULL;**  **char str [300];**  **for(int i = 0; i < quan; i++){**  **// сделать проверку корректности ввода пользователя !**  **drawDownBox(box);**  **int shift = 1;**  **int number = i + 1;**  **conMove(box.down + shift, box.left);**  **printf("\tLecture course №%i", number);**  **shift += 2;**  **lC = &pArr[i];**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть назву курсу: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 1, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть прізвище викладача: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 2, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть ім'я викладача: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 3, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть по-батькові викладача: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 4, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть вік викладача: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 5, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть довжину курсу: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 6, str);**  **shift++;**  **conMove(box.down + shift, box.left);**  **printf("\t Введіть оцінку курсу: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam(lC, 7, str);**  **shift++;**  **}**  **}**  **int conSaveToFile(struct menuBox box, struct lectureCourse\* pArr, int quan){**  **char FileName [50];**  **int status = 0;**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Введіть назву файлу, у якому збережуться дані: ");**  **scanf("%s", FileName);**  **status = saveToFile(pArr, quan, FileName);**  **return status;**  **}**  **struct lectureCourse\* conDeleteStruct(struct menuBox box, struct lectureCourse\* pArr, int\* quan){**  **int delnum = 0;**  **int flag = 1;**  **while(flag){**  **conMove(box.bottom - 1, box.left);**  **printf("\t Введіть номер структури, що буде видалена: ");**  **scanf("%i", &delnum);**  **if(delnum > 0 && delnum <= \*quan){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Структури з таким порядковим номером не існує ");**  **}**  **}**  **pArr = deleteStruct(pArr, quan, delnum);**  **return pArr;**  **}**  **int checkSrt(struct menuBox box, struct lectureCourse\* pArr){**  **if(pArr == NULL){**  **conMove(box.bottom - 1, box.left);**  **printf(" \tНа жаль, масив структур тепер не містить жодного елемента");**  **conHideCursor();**  **return 1;**  **} else {**  **return 0;**  **}**  **}**  **void conEmptyStr(struct menuBox box) {**  **int flag = 1;**  **while(flag){**  **char ch = conGetChar();**  **if(isalpha(ch)){**  **if('q' == ch){**  **flag = 0;**  **}**  **} else {**  **conMove(box.bottom - 1, box.left);**  **printf("\t На жаль, масив структур тепер не містить жодного елемента");**  **conMove(box.bottom, box.left);**  **printf("\t Операції над масивом є неможливими");**  **conHideCursor();**  **}**  **}**  **}**  **void drawParametrsList(struct menuBox box){**  **int x;**  **int y;**  **int wildth = 60;**  **conSetAttr(BG\_BLACK);**  **for (x = box.head; x < box.down - 1; x++) {**  **for (y = box.right + 5; y < box.right + 5 + wildth; y++) {**  **conMove(x, y);**  **putchar(' ');**  **}**  **}**  **conMove(box.head + 1, box.right + 7);**  **printf("1. Назва курсу (строка)");**  **conMove(box.head + 2, box.right + 7);**  **printf("2. Прізвище викладача (строка)");**  **conMove(box.head + 3, box.right + 7);**  **printf("3. Ім'я викладача (строка)");**  **conMove(box.head + 4, box.right + 7);**  **printf("4. По-батькові/друге ім'я викладача (строка)");**  **conMove(box.head + 5, box.right + 7);**  **printf("5. Вік викладача(ціле число)");**  **conMove(box.head + 6, box.right + 7);**  **printf("6. Тривалість курсу у годинах (ціле число)");**  **conMove(box.head + 7, box.right + 7);**  **printf("7. Оцінка курсу(дробове число)");**  **}**  **void conRefillParam(struct menuBox box, struct lectureCourse\* pArr, int quan){**  **int line = 0;**  **int strNum = 0;**  **char str [300];**  **int flag = 1;**  **while(flag){**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Оберіть зі списку параметрів той, що бажаєте змінити: ");**  **line = getInt();**  **if(line > 0 && line <= 7){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Невірний номер параметра ");**  **}**  **}**  **flag = 1;**  **while(flag){**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Оберіть номер структури, параметр якої бажаєте змінити: ");**  **strNum = getInt();**  **if(strNum > 0 && strNum <= quan){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Структури з таким порядковим номером не існує ");**  **}**  **}**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf(" Введіть нове значення: ");**  **fgets(str, 300, stdin);**  **str[strlen(str) - 1] = '\0';**  **refillParam( &(pArr[strNum - 1]), line, str );**  **}**  **void conRefillStruct(struct menuBox box, struct lectureCourse\* pArr, int quan){**  **int strNum = 0;**  **int flag = 1;**  **while(flag){**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Оберіть номер структури, параметри якої бажаєте змінити: ");**  **strNum = getInt();**  **if(strNum > 0 && strNum <= quan){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Структури з таким порядковим номером не існує ");**  **}**  **}**  **conMove(box.down + 1, box.left);**  **printf("\t Введіть нові параметри для структури ");**  **conFillArr(&(pArr[strNum - 1]), 1, box);**  **}**  **void conSearchStr(struct menuBox box, struct lectureCourse\* pArr, int quan){**  **int x = 0;**  **int resultsc = 0;**  **struct lectureCourse\* res [quan];**  **int shift = 2;**  **char str [20];**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Введіть значення Х: ");**  **fgets(str, 20, stdin);**  **x = atoi(str);**  **drawDownBox(box);**  **resultsc = searchStr(pArr, quan, x,res);**  **conMove(box.bottom, box.left);**  **if(resultsc == 0){**  **conMove(box.down + 1, box.left);**  **printf("\t Жодної структури з такими значеннями не було знайдено ");**  **} else {**  **conMove(box.down + 1, box.left);**  **printf("\t Курси, що мають загальну тривалість більше за %i:", x);**  **for(int i = 0; i < resultsc; i++){**  **conMove(box.down + shift, box.left);**  **printf("\t %s", (res[i])->courseName);**  **fflush(stdout);**  **shift ++;**  **}**  **}**  **}**  **void conStructToStr(struct menuBox box, struct lectureCourse\* pArr, int quan){**  **int strNum = 0;**  **int flag = 1;**  **char str [1000] = "";**  **while(flag){**  **drawDownBox(box);**  **conMove(box.bottom - 1, box.left);**  **printf("\t Оберіть номер структури, яку бажаєте вивести як строку: ");**  **strNum = getInt();**  **if(strNum > 0 && strNum <= quan){**  **flag = 0;**  **} else {**  **conMove(box.bottom, box.left);**  **printf("\t Структури з таким порядковим номером не існує ");**  **}**  **}**  **(void) structToStr(&pArr[strNum - 1], str);**  **drawDownBox(box);**  **conMove(box.down + 1, box.left + 1);**  **conSetAttr(BG\_BLACK);**  **printf("%s", str);**  **conSetAttr(BG\_CYAN);**  **}** |

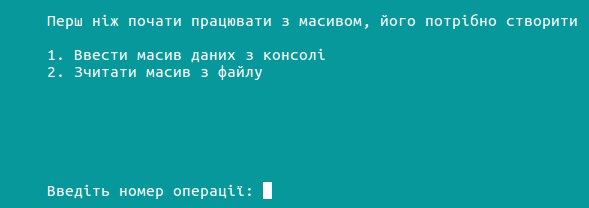
**Завдання 4.**

|  |
| --- |
| func.c |
| **#include "lab\_3.h"**  **#include "func.h"**  **struct lectureCourse\* makePArr(int quan){**  **struct lectureCourse\* pArr = NULL;**  **pArr = (struct lectureCourse\*) malloc(quan \* sizeof(struct lectureCourse));**  **if (NULL == pArr){**  **printf("Problems allocating memory");**  **return NULL;**  **} else {**  **return pArr;**  **}**  **// обязательно в конце кода сделать free!**  **}**  **int saveToFile(struct lectureCourse\* pArr, int quan, const char \* FileName){**  **int number = 0;**  **FILE \* fout = fopen(FileName, "w");**  **if (fout == NULL){**  **printf("Error opening file\n");**  **return 0;**  **}**  **for(int i = 0; i < quan; i++){**  **number = i + 1;**  **fprintf(fout, "Lecture course № %i", number);**  **fprintf(fout, "\ncourse name: %s", pArr[i].courseName);**  **fprintf(fout, "\nlecturer`s surname: %s", pArr[i].lecturer.surname);**  **fprintf(fout, "\nlecturer`s name: %s", pArr[i].lecturer.name);**  **fprintf(fout, "\nlecturer`s middle name: %s", pArr[i].lecturer.middlen);**  **fprintf(fout, "\nlecturer`s age: %i", pArr[i].lecturer.age);**  **fprintf(fout, "\ncourse length: %i", pArr[i].length);**  **fprintf(fout, "\ncourse rating: %f", pArr[i].rating);**  **fprintf(fout, "\n\n");**  **}**  **fclose(fout);**  **return 1;**  **}**  **int readFromFile (struct lectureCourse\* pArr, int quan, const char \* FileName) {**  **FILE \* fr = fopen(FileName, "r");**  **if (fr == NULL){**  **printf("Error opening file \n");**  **return 1;**  **}**  **char temp [5];**  **for(int i = 0; i < quan && (!feof(fr)); i++){**  **fgets(pArr[i].courseName, 300, fr);**  **pArr[i].courseName[strlen(pArr[i].courseName) - 1] = '\0';**  **fscanf(fr, "%s", pArr[i].lecturer.surname);**  **fscanf(fr, "%s", pArr[i].lecturer.name);**  **fscanf(fr, "%s", pArr[i].lecturer.middlen);**  **fscanf(fr, "%i", &pArr[i].lecturer.age);**  **fscanf(fr, "%d", &pArr[i].length);**  **fscanf(fr, "%f", &pArr[i].rating);**  **fgets(temp, 5, fr);**  **}**  **fclose(fr);**  **return 0;**  **}**  **void copyStruct (struct lectureCourse\* newStr, struct lectureCourse\* oldStr){**  **strcpy(newStr->courseName, oldStr->courseName);**  **strcpy(newStr->lecturer.surname, oldStr->lecturer.surname);**  **strcpy(newStr->lecturer.name, oldStr->lecturer.name);**  **strcpy(newStr->lecturer.middlen, oldStr->lecturer.middlen);**  **newStr->lecturer.age = oldStr->lecturer.age;**  **newStr->length = oldStr->length;**  **newStr->rating = oldStr->rating;**  **}**  **struct lectureCourse\* deleteStruct(struct lectureCourse\* pArr, int\* quan, int delnum){**  **struct lectureCourse temp [\*quan - 1];**  **for(int i = 0; i < delnum - 1; i ++){**  **copyStruct(&temp[i], &pArr[i]);**  **}**  **for(int i = delnum; i < \*quan; i ++){**  **copyStruct(&temp[i -1], &pArr[i]);**  **}**  **\*quan = \*quan - 1;**  **pArr = (struct lectureCourse\*) realloc(pArr, \*(quan) \* sizeof(struct lectureCourse));**  **for(int i = 0; i < \*quan && i != delnum + 1; i ++){**  **copyStruct(&pArr[i], &temp[i]);**  **}**  **return pArr;**  **}**  **void refillParam(struct lectureCourse\* course, int line, char str [300]){**  **int var = 0;**  **switch(line){**  **case 1:**  **strcpy(course ->courseName, str);**  **break;**  **case 2:**  **strcpy(course ->lecturer.surname, str);**  **break;**  **case 3:**  **strcpy(course ->lecturer.name, str);**  **break;**  **case 4:**  **strcpy(course ->lecturer.middlen, str);**  **break;**  **case 5:**  **var = atoi(str);**  **course ->lecturer.age = var;**  **break;**  **case 6:**  **var = atoi(str);**  **course ->length = var;**  **break;**  **case 7:**  **var = atof(str);**  **course ->rating = var;**  **break;**  **}**  **}**  **int searchStr(struct lectureCourse\* pArr, int quan, int x, struct lectureCourse\* res[]){**  **int count = 0;**  **for(int i = 0; i < quan; i++){**  **if(pArr[i].length > x) {**  **res[count] = &pArr[i];**  **count++;**  **}**  **}**  **return count;**  **}**  **char\* structToStr(struct lectureCourse\* lC, char str [1000]){**  **char buffer [20] = "";**  **str[0] = '\0';**  **if(lC == NULL){**  **return str;**  **}**  **strcat(str, lC->courseName);**  **strcat(str, " ");**  **strcat(str, lC->lecturer.surname);**  **strcat(str, " ");**  **strcat(str, lC->lecturer.name);**  **strcat(str, " ");**  **strcat(str, lC->lecturer.middlen);**  **strcat(str, " ");**  **sprintf(buffer, "%i ", lC ->lecturer.age);**  **strcat(str, buffer);**  **sprintf(buffer, "%i ", lC ->length);**  **strcat(str, buffer);**  **sprintf(buffer, "%5f", lC ->rating);**  **strcat(str, buffer);**  **return str;**  **}** |

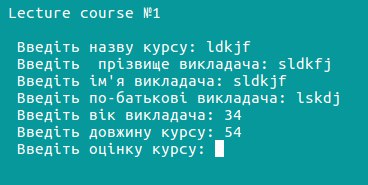
**Приклади результатів**

**Загальні риси роботи.**

При виклику програми відкривається початкове меню, в якому користувач може зробити вибір щодо способу сворення робочого масиву.



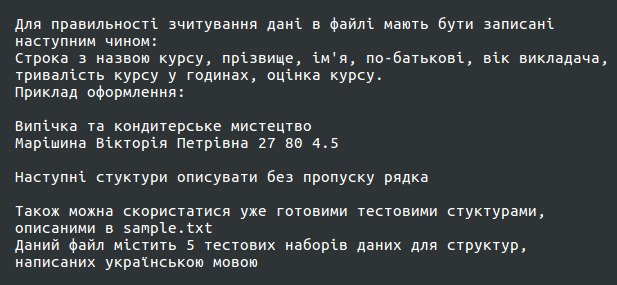
У випадку вибору першого варіанту відкривається меню, в якому користувач вводить з консолі бажані значення структур



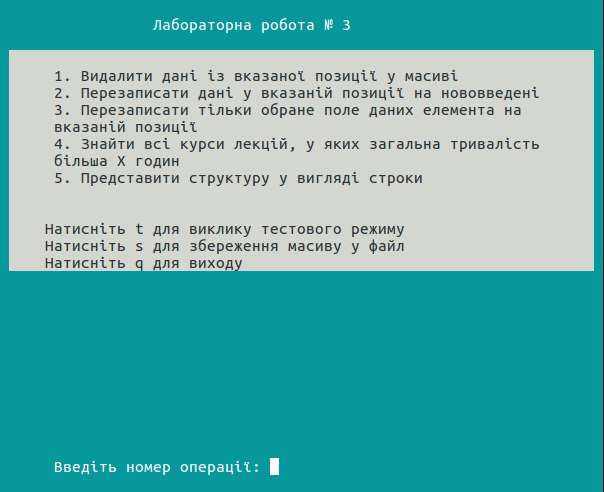
Якщо користувач обирає варіант два, відкривається наступне вікно

https://pp.vk.me/c638725/v638725471/15f4f/-vpc2Wr8-NM.jpg

Воно супроводжується інстукціями щодо того, як мають бути розташовані дані у файлі задля коректного считування



Після створення масиву відкривається головне меню:



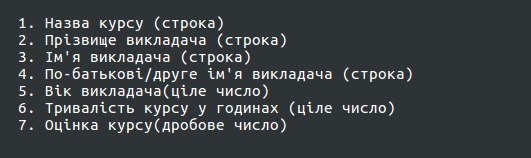
Для видалення елементу масиву користувачу достатньо ввести номер відповідного елемента:

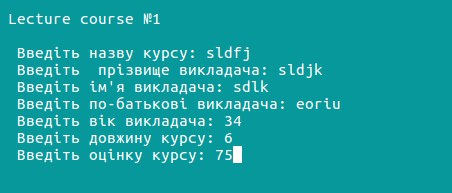
https://pp.vk.me/c638725/v638725471/15f65/YNDL6Zojr_8.jpg

Якщо користувач бажає змінити певні параметри окремого елемента або ж всі параметри окремого елемента, він має ввести номер відповідного елемента

https://pp.vk.me/c638725/v638725471/15f6c/1rXcDG7a3H4.jpg

А також номер із списку, що з’явився, який позначатиме відповідне поле у структурі.





Якщо користувач бажає знайти курси лекцій, що довші за Х, він має ввести Х. Тоді отримується список таких курсів лекцій:

https://pp.vk.me/c638725/v638725471/15f81/KrVpCTJuo8I.jpg

Для виводу окремого елемента масиву у вигляді строки користувачу також достатньо ввести номер структури

https://pp.vk.me/c638725/v638725471/15f88/_kRbZ0eQrZU.jpg

При вході в тестовий режим користувач отримує повідомлення про успішне проходження усіх тестів:

https://pp.vk.me/c638725/v638725471/15f5e/h5JNzBpaf5Q.jpg

**Висновки**

При виконанні лаботаторної роботи було закріплено навички роботи зі структурами, вказівниками. Відпрацьовано навички створення header-файлів та розбиття коду на окремі файли в залежності від того, яку функцію він виконує.

Крім того було винонано розбиття програми на функції, кожна з котрих виконувала певну дію.

Декотрі функції було протестовано на певних стандартних та межових випадках за допомогою бібліотеки <assert.h>. Тести функцій знаходилися у файлі test.c

Також було розроблено консольний інтерфейс користувача, оформлений задля зручності його роботи.

Компіляція всього коду відбувалася за допомогою компілятора gcc.