**Форма № Н-6.01**

\_\_\_\_\_\_\_\_\_\_\_\_\_*Національний університет харчових технологій*\_\_\_\_\_\_\_\_\_

(повне найменування вищого навчального закладу)

\_\_\_\_\_\_\_\_\_\_\_\_ *Інформаційних технологій, штучного інтелекту і \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_кібербезпеки*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва кафедри, циклової комісії)

**КУРСОВА РОБОТА**

з дисципліни  *Основи програмування та алгоритмічні мови*

(назва дисципліни)

на тему:\_\_\_ *Створення консольної інформаційної системи:* Довідник гелікоптерів

Студента (ки) \_\_1\_ курсу \_\_\_*2*\_\_групи

спеціальності\_*122 "Комп’ютерні науки "*

*Держій Д.Ю*

Керівник\_\_\_\_\_\_\_*Грибков С.В.*\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_Оцінка: ECTS \_\_\_

Члени комісії

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

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

Я, як здобувач(-ка) Національного університету харчових технологій, розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Київ-2023

*НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ*

(назва вищого навчального закладу)

Кафедра *Інформаційних систем*

Дисципліна *Основи програмування та розробки алгоритмів*

Спеціальність *122 «Комп’ютерні науки»*

Курс 1 Група 2 Семестр IІ

**ЗАВДАННЯ**

**на курсову роботу студента**

Держій Денис Юрійович

(прізвище, ім’я, по-батькові)

1. Тема роботи *Створення консольної інформаційної системи обліку довідника гелікоптерів*

2. Термін здачі студентом закінченого роботи

3. Вихідні дані до роботи *1. Структура запису: модель, виробник, тип, кількість пасажирських місць, вантажопідйомність, рік випуску. 2.Перелік довідок для пошуку запитів*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) Розділ 1 «Постанова задачі»

Розділ 2 «Методи вирішення задачі»

Розділ 3 «Інструкція користувача розробки»

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Блок-схеми алгоритмів

2. Скріншоти інтерфейсу користувача

6. Дата видачі завдання

**АНОТАЦІЯ**

**Держій Д.Ю. Створення консольної інформаційної системи обліку**

**гелікоптерів.**

Курсова робота складається з \_\_ сторінок, \_\_ таблиць, \_\_ рисунків,

\_\_ додатків та \_\_ літературних джерел.

В даній курсовій роботі створена консольна інформаційна система обліку гелікоптерів, що забезпечує отримання цілої низки різної довідкової інформації про виробників, типи гелікоптерів. Під час виконання курсової роботи проведено аналіз предметної області по обліку надходження серіалів за допомогою інтернету, розроблено блок-схеми алгоритмів підпрограм для реалізацій функцій системи, створено текстовий інтерфейс користувача.

Інформаційна система створена у середовищі Visual Studio та з використанням мови С\С++.

Ключові слова: ГЕЛІКОПТЕРИ, ІНФОРМАЦІЙНА СИСТЕМА, ПІДПРОГРАМИ, ПРОГРАМНІ БІБЛІОТЕКИ.

**SUMMARY**

**Derzhii D.Y. Creation of a consolidated information system for accounting**

**of helicopters.**

The course work consists of \_\_ pages, \_\_ tables, \_\_ figures, \_\_ appendices and \_\_ references.

In this course work, a console information system for helicopter accounting has been created, which provides a number of different reference information about manufacturers and types of helicopters. In the course of the course work, the subject area of accounting for the receipt of serials via the Internet was analysed, flowcharts of subroutine algorithms for implementing the system functions were developed, and a textual user interface was created.

The information system was created in the Visual Studio environment and using the C\C++ language.

Keywords: HELICOPTERS, INFORMATION SYSTEM, SUBROUTINES, SOFTWARE LIBRARIES.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсової роботи | Термін виконання етапів курсової  роботи | Примітка |
|  | Аналіз поставленої задачі |  |  |
|  | Вибір методів вирішення задачі, опис структур та алгоритмів |  |  |
|  | Створення консольної інформаційної системи |  |  |
|  | Оформлення пояснювальної записки |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_** Держій. Д. Ю

( підпис ) (прізвище та ініціали)

**Керівник курсової роботи \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Розділ №1: Постановка задачі.**

Для створення консольної інформаційної системи, яка задовольняє наведеним вимогам, вам знадобиться використовувати мову програмування C/C++. Ось загальний опис кроків, які ви можете виконати для реалізації системи:

1. Створення файлу: Реалізуйте функцію, яка дозволяє користувачеві створити новий файл для зберігання даних. Ви можете використовувати функції введення/виведення файлів в мові C/C++ для цього.
2. Відкриття файлу: Реалізуйте функцію, яка дозволяє користувачеві відкрити існуючий файл з даними. Використовуйте функції введення/виведення файлів для цього.
3. Виведення усієї інформації: Реалізуйте функцію, яка зчитує дані з файлу і виводить їх на екран у вигляді таблиці зі заголовком. Якщо кількість записів не поміщається на екран, забезпечте можливість перегляду додаткових записів.
4. Додавання нового запису: Реалізуйте функцію, яка дозволяє користувачеві додавати нові записи до файлу. Перед додаванням перевіряйте коректність введених даних.
5. Видалення запису: Реалізуйте функцію, яка дозволяє користувачеві видаляти записи з файлу. Перед видаленням підтвердіть дію користувача.
6. Редагування запису: Реалізуйте функцію, яка дозволяє користувачеві редагувати існуючі записи у файлі. Забезпечте можливість зміни будь-якого поля запису.
7. Пошук і фільтрація даних: Реалізуйте функцію, яка дозволяє користувачеві шукати та фільтрувати дані за заданими параметрами. Виведіть результати на екран у вигляді таблиці зі заголовком.
8. Довідка та інформація про розробника: Реалізуйте функції, які виводять користувачеві інформацію про розробника.
9. Вихід: При виборі пункту меню "вихід", забезпечте збереження всіх змін і закриття відкритих файлів.

**Розділ №3: Блок-схеми**

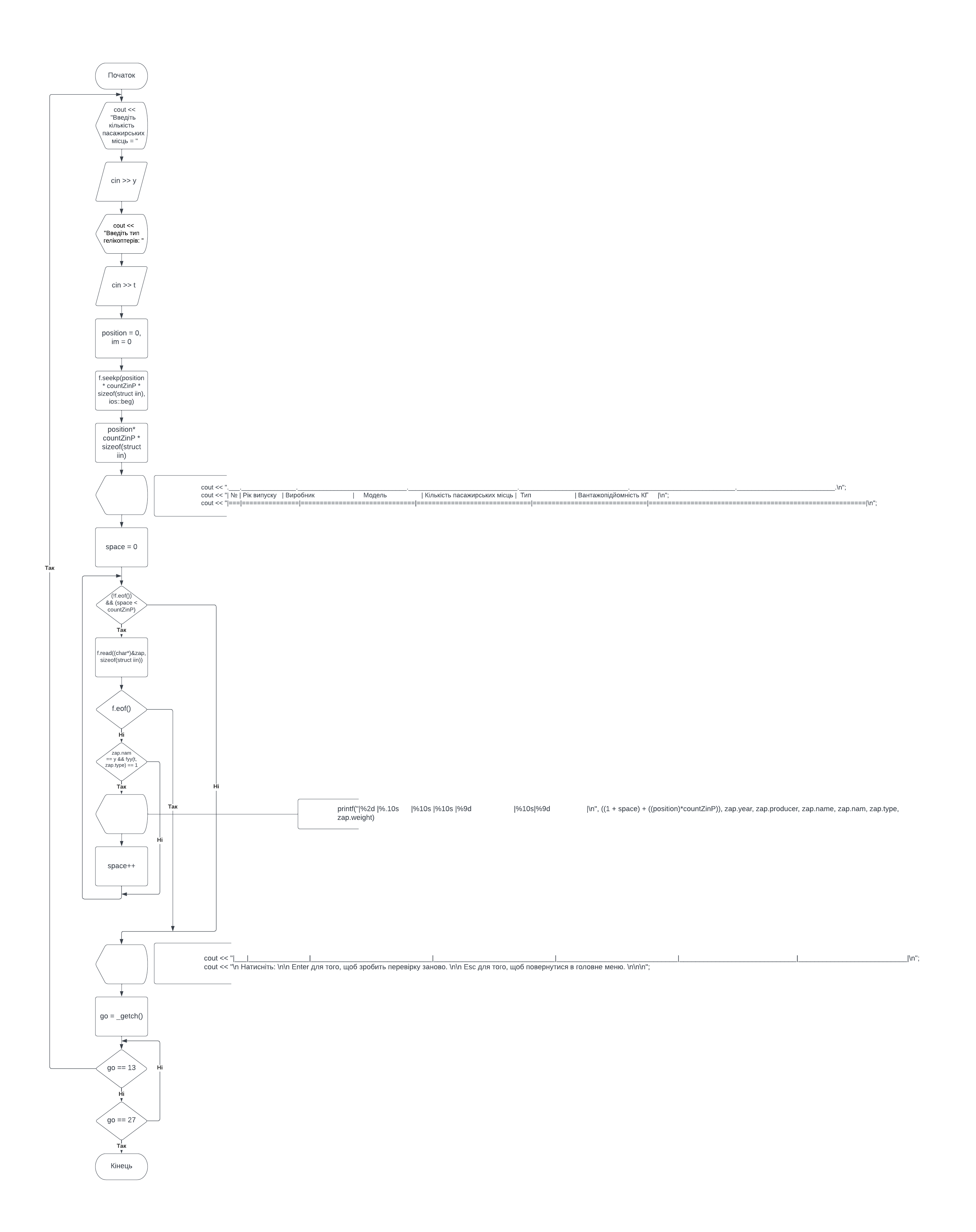


Рис. 1. Блок-схема до процедури tsk1()

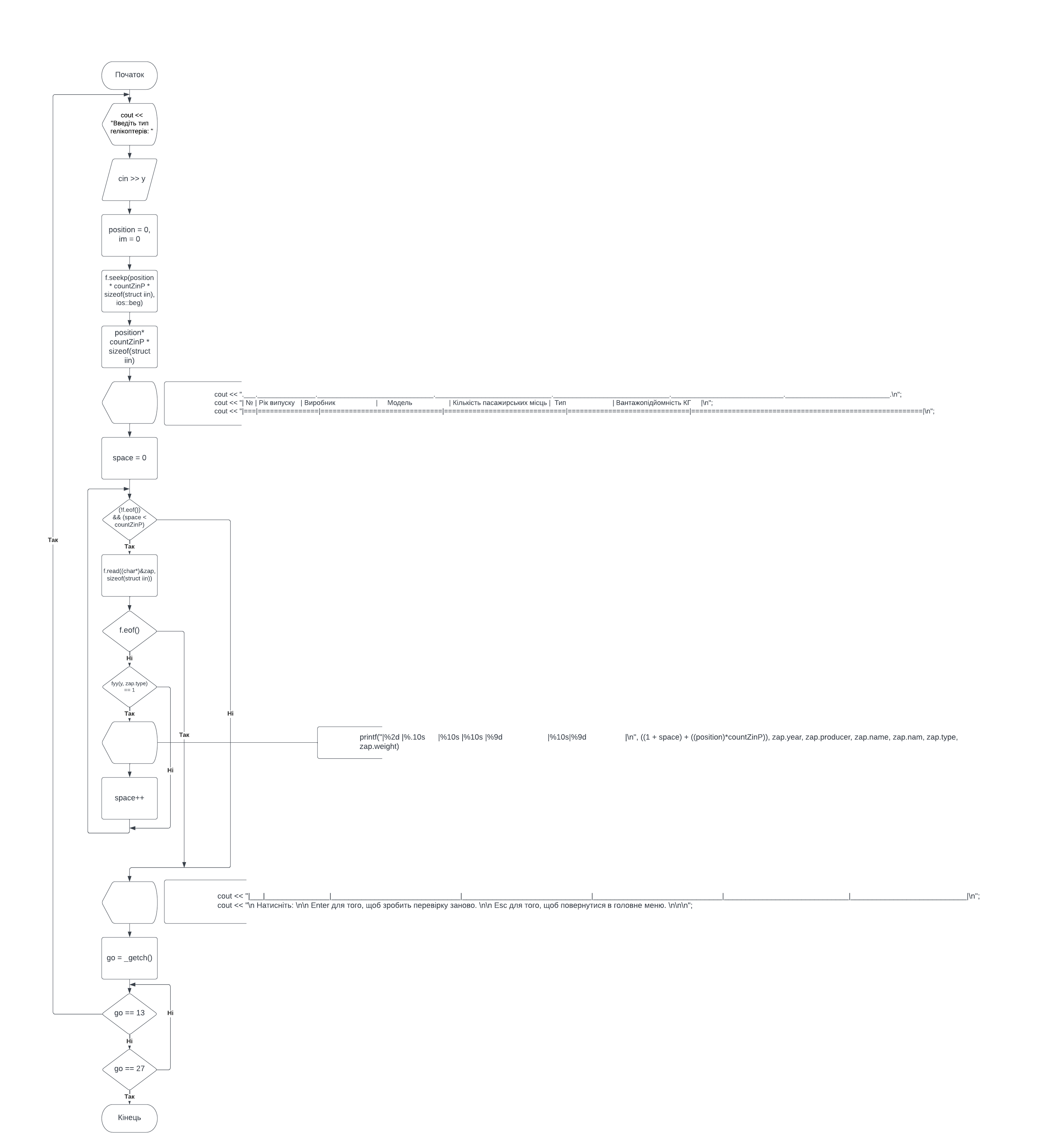


Рис. 2. Блок-схема до процедури tsk2()

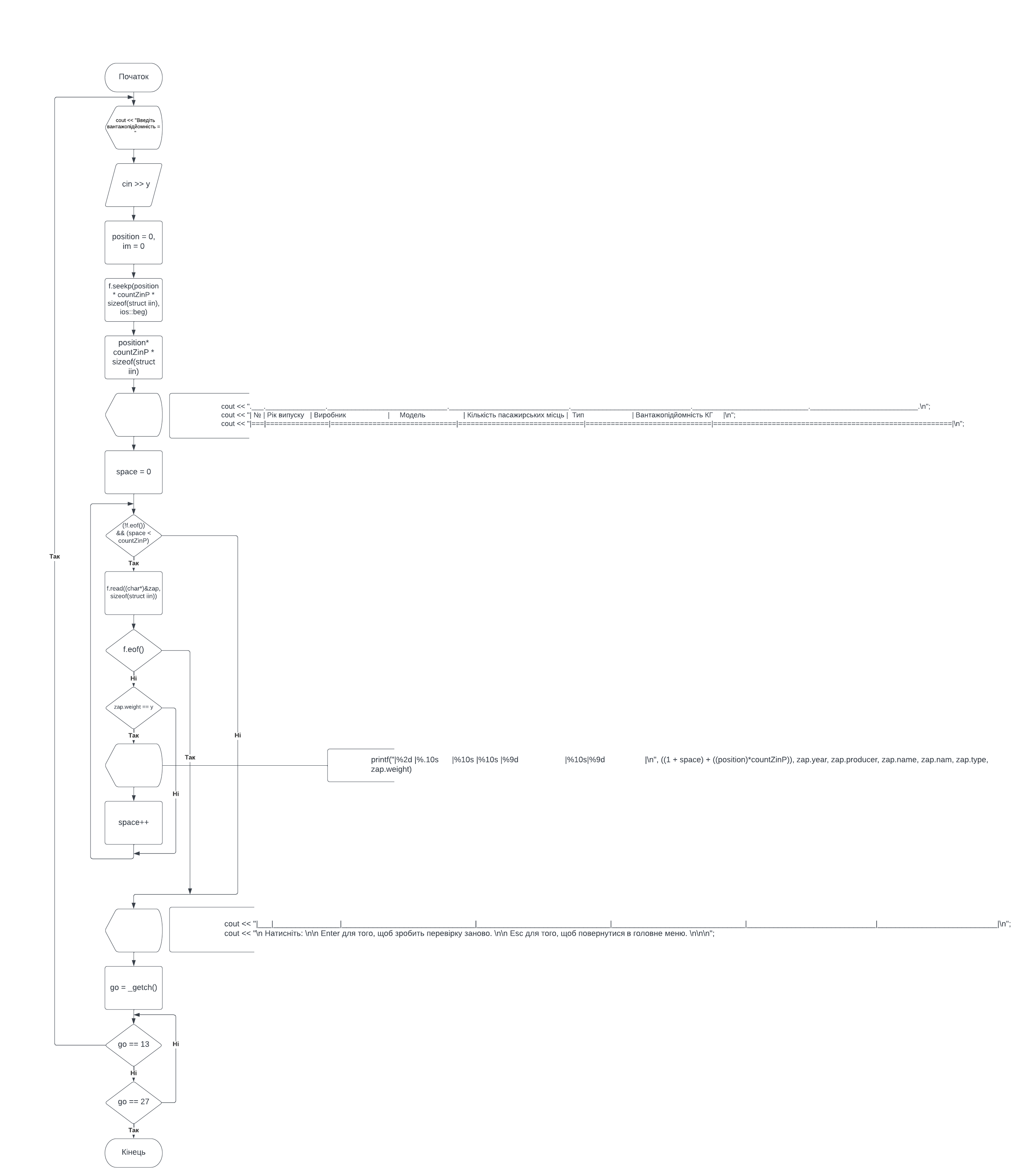


Рис. 3. Блок-схема до процедури tsk3()

**Розділ №4: Тексти програм**

#include <iostream>

#include <fstream>

#include <conio.h>

#include <cstdio>

#include <cstring>

#include <windows.h>

using namespace std;

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE); /\* Отримання

дескриптора пристрою стандартного виводу, а саме консолі\*/

//задаємо константи, що визначають довжину символьних полів: Виробник, Модель, Тип, Рік випуску.

const int l\_producer = 30, l\_model = 30, l\_type = 30, l\_year = 10;

const int countZinP = 50; // оголошуємо константу, що регламентує кількість записів на екрані

//основна структура

struct iin

{

char producer[l\_producer];

char year[l\_year];

char name[l\_model];

int nam;

int weight;

char type[l\_type];

};

// глобальні змінні

fstream f; // файлова змінна

char fdir[100] = "1.txt"; // змінна для збереження повного імені файлу

void menu(); // попереднє оголошення процедури

// \*\*\* Процедура створення/відкриття файлу \*\*\*

void open\_new()

{

system("cls");

printf(" Вкажіть повне ім'я файлу ( Приклад: \"С:\\Program Files\\file.dat\" ): ");

cin >> fdir;

opf:

f.open(fdir, ios::in | ios::out | ios::binary); // відкриваємо файл у бінарному режимі

if (!f.is\_open()) //якщо файл не відкрився

{

printf("\n\n Помилка при відкритті файлу, буде спроба його створити.\n\n\n ");

ofstream ofs(fdir); //створюємо файл

ofs.close(); //закриваємо потік створеного файлу

printf(" Файл створено. \n\n ");

goto opf;

}

else

printf(" Файл відкрито для роботи. \n\n ");

system("pause");

}

// \*\*\* Процедура додавання запису \*\*\*

void addzap()

{

system("cls");

int i, // змінна призначена для номера нового запису

pp, // змінна для збереження натиснутої клавіші

j, // змінна для збереження довжини символьних змінних

t, // змінна для збереження довжини символьних змінних

q1;

iin heli; // оголошуємо змінну для роботи із записом

f.clear(); // очищуємо прапорці помилок

f.seekg(0, ios::end); // переводимо вказівник на кінець файлу

i = f.tellp() / sizeof(struct iin); //визначаємо кількість записів

s11:

i++; //номер нового запису

// обнулюяємо поля запису

strcpy(heli.year, "");

strcpy(heli.producer, "");

strcpy(heli.name, "");

heli.nam = 0;

heli.weight = 0;

strcpy(heli.type, "");

pp = 0;

//далі заповнюємо поля нового запису

cout << " Запис № " << i << ": \n";

cout << " Рік випуску -> ";

do

{

gets\_s(heli.year); // зчитуємо значення у символьне поле

t = strlen(heli.year); //визначаємо довжину символьного поля

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

for (j = t; j < l\_year - 1; j++) // цикл забезпечує додавання до символьного поля пробілів до заданої довжини

strcat(heli.year, " ");

cout << "\n Виробник -> ";

do

{

gets\_s(heli.producer); // зчитуємо значення у символьне поле

t = strlen(heli.producer); //визначаємо довжину символьного поля

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

for (j = t; j < l\_producer - 1; j++) // цикл забезпечує додавання до символьного поля пробілів до заданої довжини

strcat(heli.producer, " ");

cout << "\n Модель -> ";

do

{

gets\_s(heli.name); // зчитуємо значення у символьне поле

t = strlen(heli.name); //визначаємо довжину символьного поля

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

for (j = t; j < l\_model - 1; j++) // цикл забезпечує додавання до символьного поля пробілів до заданої довжини

strcat(heli.name, " ");

cout << "\n Кількість пасажирських місць -> ";

do

{

cin >> heli.nam; // зчитуємо значення у числове поле

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

cout << "\n Тип -> ";

do

{

gets\_s(heli.type); // зчитуємо значення у символьне поле

t = strlen(heli.type); //визначаємо довжину символьного поля

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

for (j = t; j < l\_type - 1; j++) // цикл забезпечує додавання до символьного поля пробілів до заданої довжини

strcat(heli.type, " ");

cout << "\n Вантажопідйомність -> ";

do

{

cin >> heli.weight; // зчитуємо значення у числове поле

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

f.write((char\*)&heli, sizeof(struct iin)); // записуємо у файл сформований запис

s12:

cout << " Для введення ще одного запису натисніть -> Enter, для закінчення -> Esc \n";

if ((pp = \_getch()) != 27)

if (pp == 13)

goto s11;

else

goto s12;

}

// \*\*\* Процедура виведення усіх записів \*\*\*

void all\_out()

{

iin zap; // оголошуємо змінну для роботи із записом

int space; // оголошуємо змінну, що є лічильником кількості записів на екрані

int position = 0; // враховує к-сть виведених записів на сторінці

int im = 0; // змінна для збереження коду натиснутої клавіші

f.clear(); // очищуємо прапорці помилок

f.seekp(0, ios::end); // переміщуємо вказівник на кінець файлу

int size = f.tellp() / sizeof(struct iin); // визначаємо кількість записів у файлі

if (size < 1)

{ // якщо записів немає, то дати повідомлення на екран

cout << "Файл порожній";

goto ex; // перехід на закінчення виконання процедури

}

star:

f.clear(); // очищуємо прапорці помилок

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg); // переміщуємо вказівник на потрібний запис, що вираховується за наступною формулою

position\* countZinP \* sizeof(struct iin);

system("cls");

printf("|<<< Загальна кількість записів у файлі = %5d >>>|\n", size);

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

space = 0; //обнуляємо значення лічильника кількості записів на екрані

while ((!f.eof()) && (space < countZinP)) //умова виконання циклу: поки не досягли кінця файлу або кількість виведених записів на екран менше countZinP

{

f.read((char\*)&zap, sizeof(struct iin)); //зчитування запису з файлу у змінну zap

if (f.eof()) break; // якщо досягнуто кінець файлу, то вийти з циклу

// виводимо поля зчитаного запису на екран

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

space++; //збільшуємо лічильник кількості записів на екрані на 1

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

printf("|<<<\"Up/Down - переміщення на стор.\"<<< №%2d >>>\"Esc -головне меню \">>>|\n", position + 1);

ctrl:

im = \_getch(); // зчитуємо код натиснутої клавіші

switch (im)

{

case 80: // якщо натиснута клавіша "стрілка вниз" - Down

{

if (position < (size - 1) / countZinP)

++position;

goto star; //повернення до мітки star:

}

case 72: // якщо натиснута клавіша "стрілка у гору" - Up

{

if (position >= 1)

--position;

goto star; //перехід до мітки star:

}

case 27:

goto ex;

default:

goto ctrl;

}

ex:;

}

// \*\*\* Процедура редагування запису \*\*\*

void editzap()

{

int t, i = 0, j, q1;

int im; // змінна для збереження коду натиснутої клавіші

start:

system("cls");

iin zap; // оголошуємо змінну для роботи із записом

int nom = 0; //номер запису для редагування

cout << "Введіть номер запису, що буде відредаговано -> ";

cin >> nom;

f.clear();

f.seekp((nom - 1) \* sizeof(struct iin), ios::beg);//Переміщуємо вказівник на потрібний запис

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

f.read((char\*)&zap, sizeof(struct iin));

{

printf("| |%.10s |%10s |%10s |%9d |%10s|%9d |\n", zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

printf("|<<<<<<<\"Для підтвердження редагування натисніть Enter, а для відміни Esc\">>>>>>>|\n");

control:

im = \_getch(); // зчитуємо код натиснутої клавіші

if (im == 13)

{

iin heli; // оголошуємо змінну для роботи із записом

// обнулюяємо поля запису

strcpy(heli.year, "");

strcpy(heli.producer, "");

strcpy(heli.name, "");

heli.nam = 0;

heli.weight = 0;

strcpy(heli.type, "");

//далі заповнюємо поля нового запису

cout << "Запис № " << i << ": \n";

cout << " Рік випуску -> ";

do

{

gets\_s(heli.year);

t = strlen(heli.year);

} while (t == 0);

for (j = t; j < l\_year - 1; j++)

strcat(heli.year, " ");

cout << "\n Виробник -> ";

do

{

gets\_s(heli.producer);

t = strlen(heli.producer);

} while (t == 0);

for (j = t; j < l\_producer - 1; j++)

strcat(heli.producer, " ");

cout << "\n Модель -> ";

do

{

gets\_s(heli.name);

t = strlen(heli.name);

} while (t == 0);

for (j = t; j < l\_model - 1; j++)

strcat(heli.name, " ");

cout << "\n Кількість пасажирських місць -> ";

do

{

cin >> heli.nam;

} while (t == 0);

cout << "\n Тип -> ";

do

{

gets\_s(heli.type);

t = strlen(heli.type);

} while (t == 0);

for (j = t; j < l\_type - 1; j++)

strcat(heli.type, " ");

cout << "\n Вантажопідйомність -> ";

do

{

cin >> heli.weight; // зчитуємо значення у числове поле

} while (t == 0); // цикл буде виконуватися до тих пір, поки довжина введеного символьного поля буде нульовою

f.write((char\*)&heli, sizeof(struct iin)); // записуємо у файл сформований запис

printf("|<<<<<<<\"Для підтвердження редагування натисніть Enter, а для відміни Esc\">>>>>>>|\n");

control2:

im = \_getch(); // зчитуємо код натиснутої клавіші

if (im == 13) // якщо натиснута клавіша Enter

{

f.seekp((nom - 1) \* sizeof(struct iin), ios::beg); // переміщуємо вказівник на запис, що буде замінено

f.write((char\*)&heli, sizeof(struct iin)); // записуємо у файл новий запис

goto start;

}

else

if (im == 27) //якщо натиснута клавіша є Esc

goto start;

goto control2;

}

else

if (im != 27) //якщо натиснута клавіша не є Esc

goto control;

system("pause");

}

// \*\*\* Процедура видалення запису \*\*\*

void delzap()

{

system("cls");

iin zap; // оголошуємо змінну для роботи із записом

int im = 0; // змінна для збереження коду натиснутої клавіші

int nom = 0; // змінна для номеру запису, що буде видалено з файлу

cout << "Введіть номер запису, що буде видалено з файлу -> ";

cin >> nom;

f.clear(); // очищуємо прапорці помилок

f.seekp((nom - 1) \* sizeof(struct iin), ios::beg);//Переміщуємо вказівник на потрібний запис

// виводимо поля запису на екран

cout << " Цей запис буде видалено з файлу \n";

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

f.read((char\*)&zap, sizeof(struct iin));

printf("| |%.10s |%10s |%10s |%9d |%10s|%9d |\n", zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

printf("|<<<\"Для підтвердження редагування натисніть Enter, а для відміни Esc\">>>|\n");

control:

im = \_getch(); // зчитуємо код натиснутої клавіші

if (im == 13) // якщо натиснута клавіша Enter

{

f.clear(); // очищуємо прапорці помилок

f.seekp(0, ios::end); // переміщуємо вказівник на кінец файлу

int size = f.tellp() / sizeof(struct iin); //визначаємо кількість записів у файлі

iin\* mas;

mas = new iin[size + 1]; // виділяємо динамічну пам’ять під масив на кількість записів у файлі

f.clear(); // очищуємо прапорці помилок

f.seekp(0, ios::beg); // переміщуємо вказівник на початок у файлі

int i = 0;

while (!f.eof())

{// зчитуємо у динамічний масив записів з файлу

f.read((char\*)&mas[i], sizeof(struct iin));

i++;

}

f.close(); // закриваємо файл

f.clear(); // очищуємо прапорці помилок

f.open(fdir, ios::in | ios::out | ios::trunc); // відкриваємо файл та очищуємо його - знищуємо усі записи у файл

f.seekp(0, ios::beg); // переміщуємо вказівник на початок файлу

for (i = 0; i < size; i++) //цикл забезпечує запис усіх елементів масиву у файл

if (i != (nom - 1)) // якщо поточний елемент не дорівнює заданому

f.write((char\*)&mas[i], sizeof(struct iin)); //запис поточного елементу у файл

delete[]mas; // очищуємо виділену динамічну пам’ять під масив записів

cout << "Запис видалено \n";

}

else

if (im != 27) //якщо натиснута клавіша не є Esc

goto control;

}

// \*\*\* Процедура завершення роботи програми \*\*\*

void exit\_prog()

{

system("cls");

cout << "\n Натисніть: \n\n Enter для того, щоб вийти. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

st:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go != 13) // перевірка чи не натиснута клавіша Enter

if (go == 27) // перевірка чи натиснута клавіша Esc

goto ex;

else

goto st; // буде здійснено перехід на мітку st, якщо буде натиснута клавіша відмінна від Enter та Esc

if (f.is\_open())

f.close(); // закриваємо файл, якщо він був відкритий

exit(0); // закриваємо програму / виходимо з неї

ex:;

}

// Порівняння двох рядків

int fyy(char\* a, char\* b)

{

int i = 0, j = strlen(a);

while (j > i)

{

if (a[i] != b[i])

return 0;

i++;

}

return 1;

}

// Процедура для виводу списку гелікоптерів за кількістю місць та типом

void tsk1()

{

l1:

system("cls");

int y;

char t[18];

cout << "Введіть кількість пасажирських місць = ";

cin >> y;

cout << "Введіть тип гелікоптерів: ";

cin >> t;

iin zap; // оголошуємо змінну для роботи із записом

int space; // оголошуємо змінну, що є лічильником кількості записів на екрані

int position = 0; // враховує к-сть виведених записів на сторінці

int im = 0; // змінна для збереження коду натиснутої клавіші

f.clear(); // очищуємо прапорці помилок

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg); // переміщуємо вказівник на потрібний запис, що вираховується за наступною формулою

position\* countZinP \* sizeof(struct iin);

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

space = 0; //обнуляємо значення лічильника кількості записів на екрані

int a, b, c, a1, b1, c1;

while ((!f.eof()) && (space < countZinP)) //умова виконання циклу: поки не досягли кінця файлу або кількість виведених записів на екран менше countZinP

{

f.read((char\*)&zap, sizeof(struct iin)); //зчитування запису з файлу у змінну zap

if (f.eof()) break; // якщо досягнуто кінець файлу, то вийти з циклу

// виводимо поля зчитаного запису на екран

if (zap.nam == y && fyy(t, zap.type) == 1)

{

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << "\n Натисніть: \n\n Enter для того, щоб зробить перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

//Процедура виводу списку гелікоптерів гелікоптерів заданого типу

void tsk2()

{

l1:

system("cls");

char y[18];

cout << "Введіть тип гелікоптерів: ";

cin >> y;

iin zap;

int space;

int position = 0;

int im = 0;

f.clear();

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg);

position\* countZinP \* sizeof(struct iin);

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

space = 0;

int a, b, c, a1, b1, c1;

while ((!f.eof()) && (space < countZinP))

{

f.read((char\*)&zap, sizeof(struct iin));

if (f.eof()) break;

if (fyy(y, zap.type) == 1)

{

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << "\n Натисніть: \n\n Enter для того, щоб зробить перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

//Процедура виводу списку гелікоптерів гелікоптерів за вантажопідйомністю

void tsk3()

{

l1:

system("cls");

int y;

cout << "Введіть вантажопідйомність = ";

cin >> y;

iin zap; // оголошуємо змінну для роботи із записом

int space; // оголошуємо змінну, що є лічильником кількості записів на екрані

int position = 0; // враховує к-сть виведених записів на сторінці

int im = 0; // змінна для збереження коду натиснутої клавіші

f.clear(); // очищуємо прапорці помилок

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg); // переміщуємо вказівник на потрібний запис, що вираховується за наступною формулою

position\* countZinP \* sizeof(struct iin);

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

space = 0; //обнуляємо значення лічильника кількості записів на екрані

int a, b, c, a1, b1, c1;

while ((!f.eof()) && (space < countZinP)) //умова виконання циклу: поки не досягли кінця файлу або кількість виведених записів на екран менше countZinP

{

f.read((char\*)&zap, sizeof(struct iin)); //зчитування запису з файлу у змінну zap

if (f.eof()) break; // якщо досягнуто кінець файлу, то вийти з циклу

// виводимо поля зчитаного запису на екран

if (zap.weight == y)

{

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << "\n Натисніть: \n\n Enter для того, щоб зробить перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

//Процедура виводу списку гелікоптерів гелікоптерів заданого виробника

void tsk4()

{

l1:

system("cls");

char y[35];

cout << "Введіть назву виробника = ";

cin >> y;

iin zap; // оголошуємо змінну для роботи із записом

int space; // оголошуємо змінну, що є лічильником кількості записів на екрані

int position = 0; // враховує к-сть виведених записів на сторінці

int im = 0; // змінна для збереження коду натиснутої клавіші

f.clear(); // очищуємо прапорці помилок

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg); // переміщуємо вказівник на потрібний запис, що вираховується за наступною формулою

position\* countZinP \* sizeof(struct iin);

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

space = 0; //обнуляємо значення лічильника кількості записів на екрані

int a, b, c, a1, b1, c1;

while ((!f.eof()) && (space < countZinP)) //умова виконання циклу: поки не досягли кінця файлу або кількість виведених записів на екран менше countZinP

{

f.read((char\*)&zap, sizeof(struct iin)); //зчитування запису з файлу у змінну zap

if (f.eof()) break; // якщо досягнуто кінець файлу, то вийти з циклу

// виводимо поля зчитаного запису на екран

if (fyy(y, zap.producer) == 1)

{

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << "\n Натисніть: \n\n Enter для того, щоб зробить перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

int stringToInt(const char\* str, int startIndex, int length)

{

int result = 0;

int multiplier = 1;

for (int i = startIndex + length - 1; i >= startIndex; --i)

{

result += (str[i] - '0') \* multiplier;

multiplier \*= 10;

}

return result;

}

// Процедура для виводу списку гелікоптерів, що випустили до/після заданого року

void tsk5()

{

l1:

system("cls");

char x[5], y[5];

cout << "Введіть початковий рік: ";

cin >> x;

cout << "Введіть кінцевий рік: ";

cin >> y;

int xValue = stringToInt(x, 0, 4);

int yValue = stringToInt(y, 0, 4);

cout << "Результати за період, крім " << xValue << " - " << yValue << ":\n";

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

int position = 0;

int space = 0;

iin zap;

f.clear();

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg);

position\* countZinP \* sizeof(struct iin);

bool dataFound = false; // Додано змінну для перевірки, чи були знайдені дані в проміжку

while ((!f.eof()) && (space < countZinP))

{

f.read((char\*)&zap, sizeof(struct iin));

if (f.eof()) break;

int year = stringToInt(zap.year, 0, 4);

if (xValue <= year && year <= yValue)

{

// Запис потрапляє в проміжок, не виводимо його

continue;

}

dataFound = true; // Знайдено дані поза проміжком

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

if (!dataFound)

{

cout << "Дані поза заданим проміжком не знайдені.\n";

}

cout << "\n Натисніть: \n\n Enter для того, щоб зробити перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

//Процедура виводу cписоку гелікоптерів, які випустили у заданий період

void tsk6()

{

l1:

system("cls");

char x[5], y[5];

cout << "Введіть початковий рік: ";

cin >> x;

cout << "Введіть кінцевий рік: ";

cin >> y;

int xValue = stringToInt(x, 0, 4);

int yValue = stringToInt(y, 0, 4);

cout << "Результати за період " << xValue << " - " << yValue << ":\n";

cout << ".\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\n";

cout << "| № | Рік випуску | Виробник | Модель | Кількість пасажирських місць | Тип | Вантажопідйомність КГ |\n";

cout << "|===|===============|==============================|==============================|==============================|=========================================================|\n";

int position = 0;

int space = 0;

iin zap;

f.clear();

f.seekp(position \* countZinP \* sizeof(struct iin), ios::beg);

position\* countZinP \* sizeof(struct iin);

while ((!f.eof()) && (space < countZinP))

{

f.read((char\*)&zap, sizeof(struct iin));

if (f.eof()) break;

int year = stringToInt(zap.year, 0, 4);

if (xValue <= year && year <= yValue)

{

printf("|%2d |%.10s |%10s |%10s |%9d |%10s|%9d |\n", ((1 + space) + ((position)\*countZinP)), zap.year, zap.producer, zap.name, zap.nam, zap.type, zap.weight);

}

space++;

}

cout << "|\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";

cout << "\n Натисніть: \n\n Enter для того, щоб зробить перевірку заново. \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

//Процедура виводу інформації про розробника

void info()

{

l1:

system("cls");

cout << " |=============================================================================|\n";

cout << " | \"Інформація про розробника\" |\n";

cout << " |\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\n";

cout << " | 1. ФІО: Держій Денис Юрійович |\n";

cout << " | 2. Вік: 18 років |\n";

cout << " | 3. Місце навчання: Національний університет харчових технологій |\n";

cout << " | 4. Курс: Перший |\n";

cout << " | 5. Факультет: АКС |\n";

cout << " | 6. Група: КН-1-2 |\n";

cout << " |=============================================================================|\n";

cout << "\n Натисніть: \n\n Esc для того, щоб повернутися в головне меню. \n\n\n";

l2:

int go = \_getch(); //зчитуємо натиснуту клавішу

if (go == 13) // перевірка чи не натиснута клавіша Enter

{

goto l1;

}

if (go == 27) // перевірка чи натиснута клавіша Esc

{

goto ex;

}

else

{

goto l2;

}

ex:

system("pause");

}

// \*\*\* Реалізація текстового інтерфейсу користувача \*\*\*

void menu()

{

int im = 0, p = 0; //змінні для збереження кодів натиснутих клавіш при роботі з пунктами меню

COORD crd, end; //оголошуємо змінні типу координат, що мають поле X та Y

crd.X = 2; // задаємо початкову позицію курсору по осі Х для відображення знаку вибору

crd.Y = 5; // задаємо початкову позицію курсору по осі Y для відображення знаку вибору

end.X = 0; // задаємо позицію курсору по осі Х за межами меню

end.Y = 15; // задаємо позицію курсору по осі Y за межами меню

start: // початок виведення меню на екран

system("cls");

cout << " |=============================================================================|\n";

cout << " | \"Довідник гелікоптерів\" |\n";

cout << " |\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\n";

cout << " | \* Головне меню \* |\n";

cout << " |\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\n";

cout << " | 1. Відкрити/створити файл |\n";

cout << " | 2. Додати запис у файл |\n";

cout << " | 3. Редагувати запис |\n";

cout << " | 4. Видалити запис з файлу |\n";

cout << " | 5. Вивести усю інформацію з файлу |\n";

cout << " | 6. Список гелікоптерів із заданими кількістю пасажирських місць і типу |\n";

cout << " | 7. Список гелікоптерів гелікоптерів заданого типу |\n";

cout << " | 8. Список гелікоптерів із заданою вантажопідємністю |\n";

cout << " | 9. Список гелікоптерів заданого виробника |\n";

cout << " | 10. Список гелікоптерів, які випустили до/після заданого року |\n";

cout << " | 11. Cписок гелікоптерів, які випустили у заданий період |\n";

cout << " | 12. Інформація про розробника |\n";

cout << " | 13. Вихiд |\n";

cout << " |=============================================================================|\n";

if (!f.is\_open()) // в залежності від того чи відкрито файл даємо відповідне повідомлення

cout << "\n Файл відсутній, завантажте або створіть файл. \n";

else

cout << "\n Файл успішно завантажено.: \"" << fdir << "\"\n";

SetConsoleCursorPosition(hConsole, crd); // переводимо курсор в позицію координат crd

SetConsoleTextAttribute(hConsole, (WORD)((20 << 0) | 10)); //задаємо колір шрифта та фону для виведення

cout << ">>>";

SetConsoleTextAttribute(hConsole, (WORD)((15 << 0) | 0)); //задаємо колір шрифта та фону для виведення

SetConsoleCursorPosition(hConsole, end);//встановлюємо курсор у нижній лівий кут

im = \_getch(); //отримуємо код натиснутої клавіші

switch (im)

{

case 72: // якщо натиснута клавіша "стрілка у гору" - Up

{

if (crd.Y > 5) crd.Y--; // якщо не досягнуто перший пункт меню, то зменшуємо значення координат crd.Y на 1 позицію

goto start; // переходимо на мітку start

}

case 80: // якщо натиснута клавіша "стрілка вниз" - Down

{

if (crd.Y < 17) crd.Y++; // якщо не досягнуто останній пункт меню, то збільшуємо значення координат crd.Y на 1 позицію

goto start; // переходимо на мітку start

}

case 13:

{

p = crd.Y + 45; // обраховуємо номер пункту меню, що буде в межах

if ((!f.is\_open()) && (p != 61)) p = 50; // якщо файл не відкрито, то буде здійснено перехід до пункту відкриття файлу

switch (p)

{

case 50: // якщо обрано перший пукт меню

{

open\_new(); // викликаємо функцію відкриття файл

goto start; // переходимо на мітку start

}

case 51:

{

addzap(); // викликаємо функцію додавання записів у файл

goto start;// переходимо на мітку start

}

case 52:

{

editzap(); // викликаємо функцію редагування записів у файл

goto start;// переходимо на мітку start

}

case 53:

{

delzap(); // викликаємо функцію видалення запису з файлу

goto start;// переходимо на мітку start

}

case 54:

{

all\_out(); // викликаємо функцію виведення усіх записів з файлу

goto start;// переходимо на мітку start

}

case 55:

{

tsk1();

goto start; // переходимо на мітку start

}

case 56:

{

tsk2();

goto start; // переходимо на мітку start

}

case 57:

{

tsk3();

goto start; // переходимо на мітку start

}

case 58:

{

tsk4();

goto start; // переходимо на мітку start

}

case 59:

{

tsk5();

goto start; // переходимо на мітку start

}

case 60:

{

tsk6();

goto start; // переходимо на мітку start

}

case 61:

{

info();

goto start; // переходимо на мітку start

}

case 62:

{

exit\_prog();// викликаємо функцію завершення виконання проєкту

goto start; // переходимо на мітку start

}

default: goto start; // переходимо на мітку start

}

}

default: goto start;

}

}

//Головна функція

int main()

{

SetConsoleCP(1251); //встановлюємо 1251 кодування для шрифту консолі

SetConsoleOutputCP(1251);

system("color 07"); //встановлюємо колір фону консолі чорним, а текст - білим

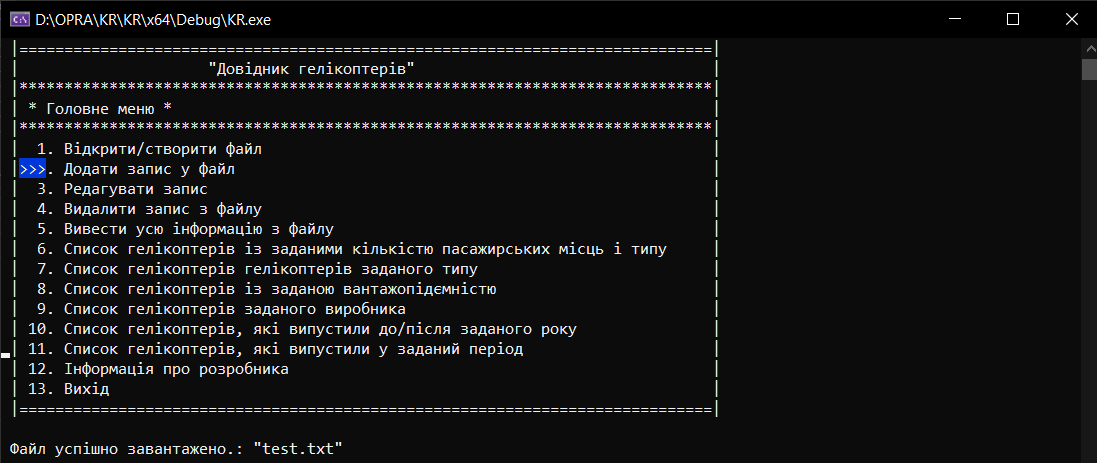
menu(); // викликаємо процедуру menu

system("pause");

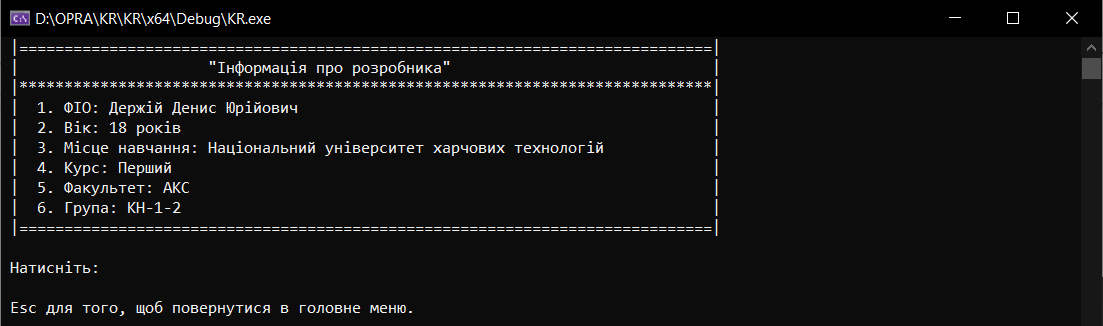
return 0;

}

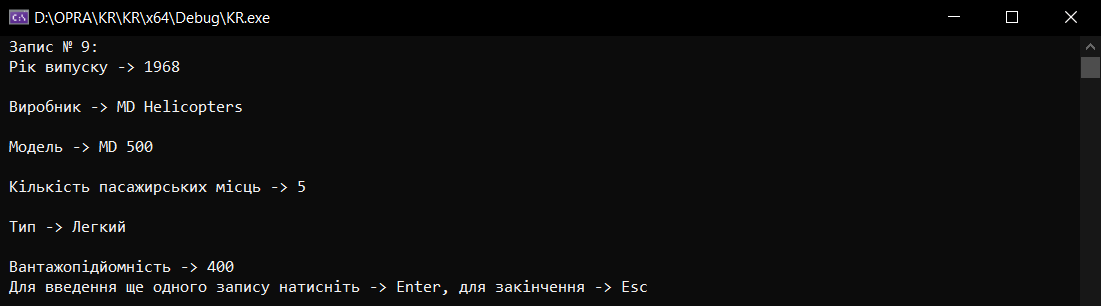
**Результати роботи програми**

****

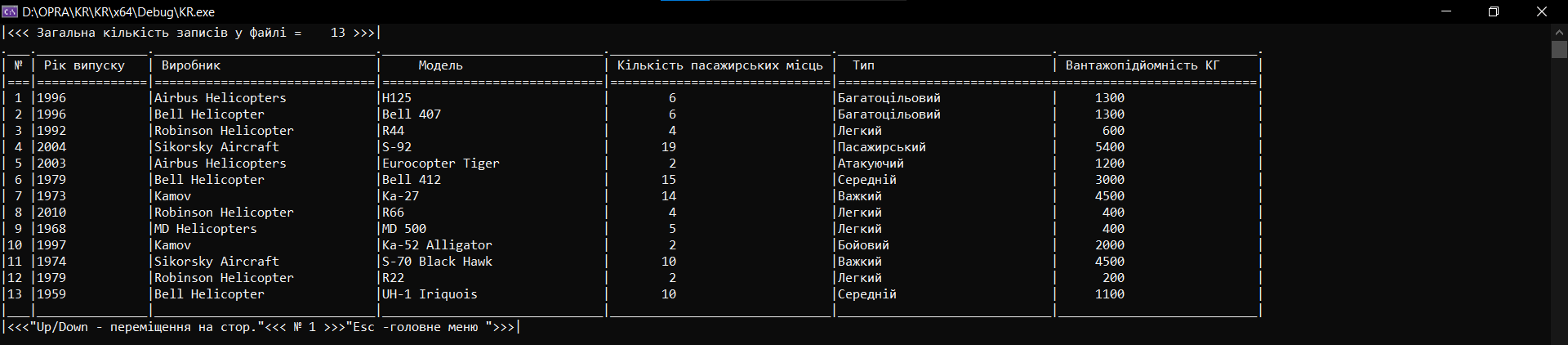
Головне меню

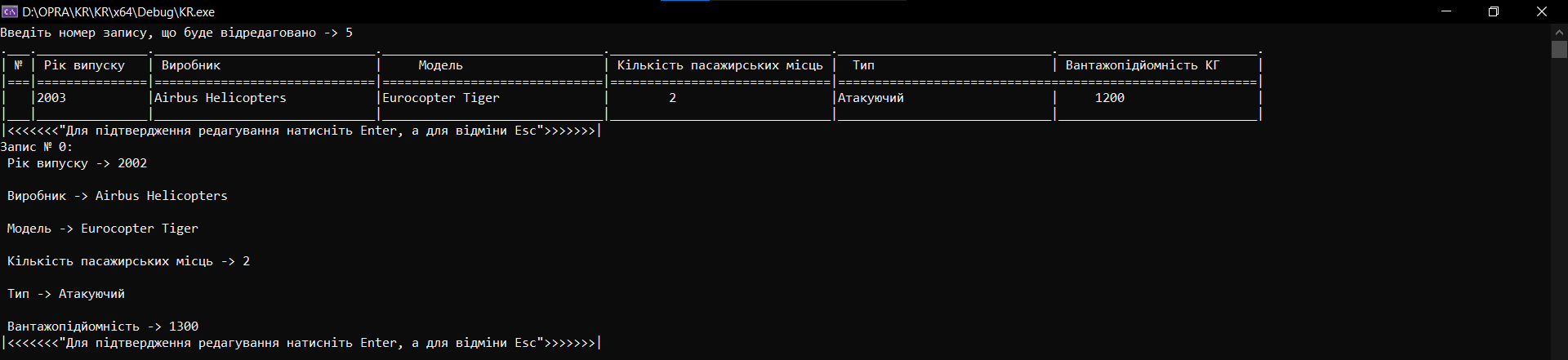
****

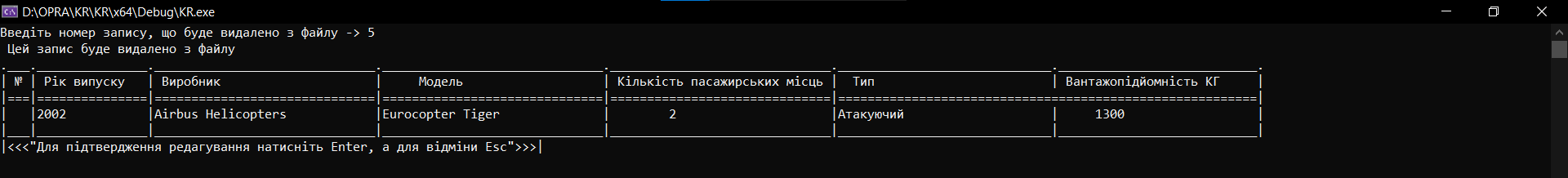
Інформація про розробника



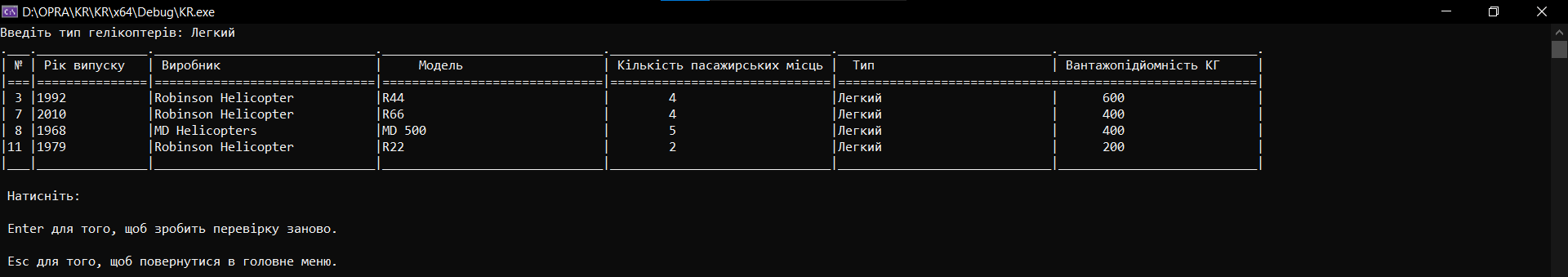
Додавання запису

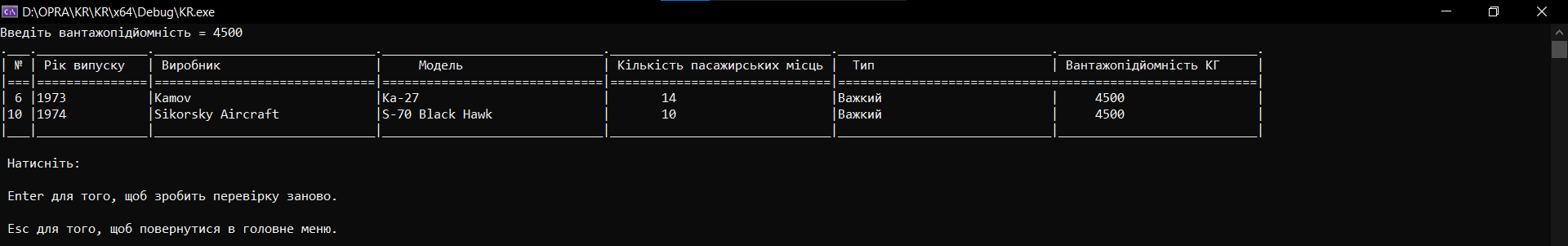
  
Вивід всієї інформації

  
Редагування запису

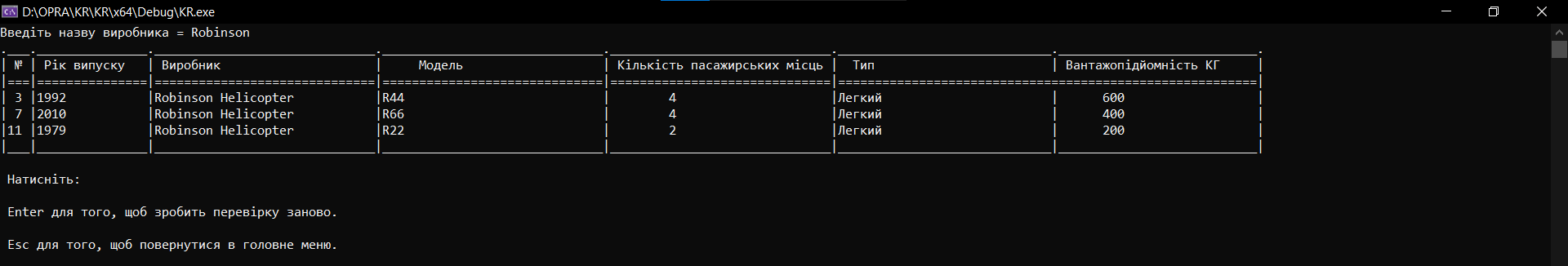


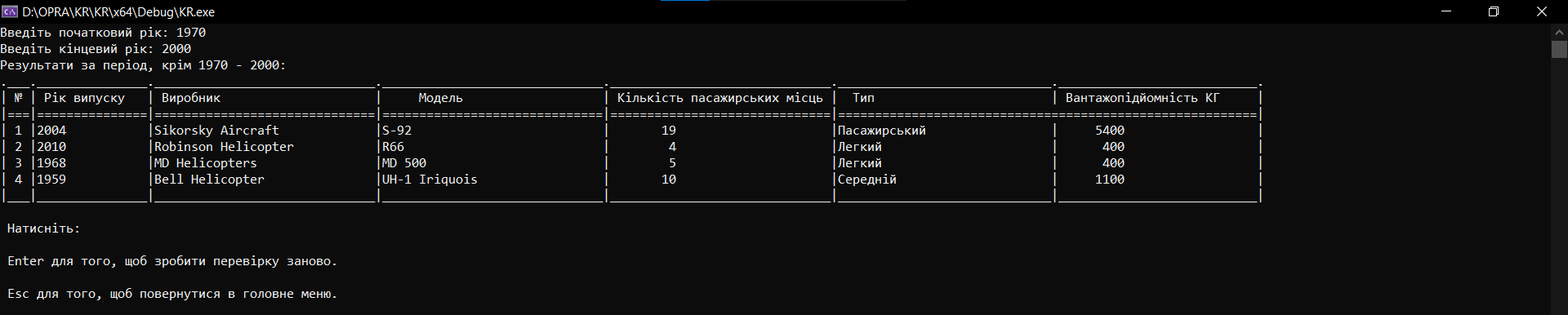
Видалення запису

  
Сортування за типом гелікоптерів

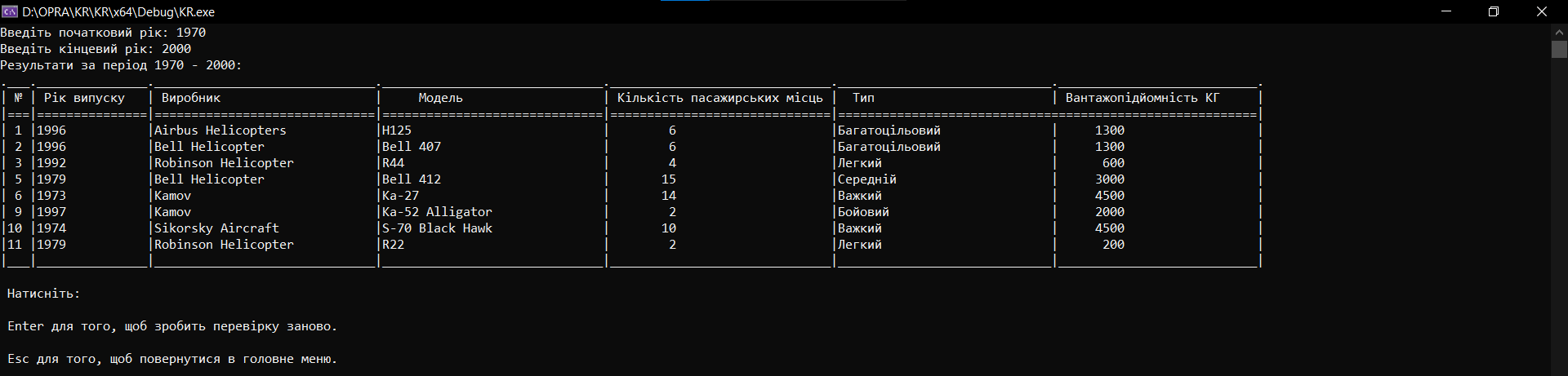


Сортування за вантажопідйомністю гелікоптерів

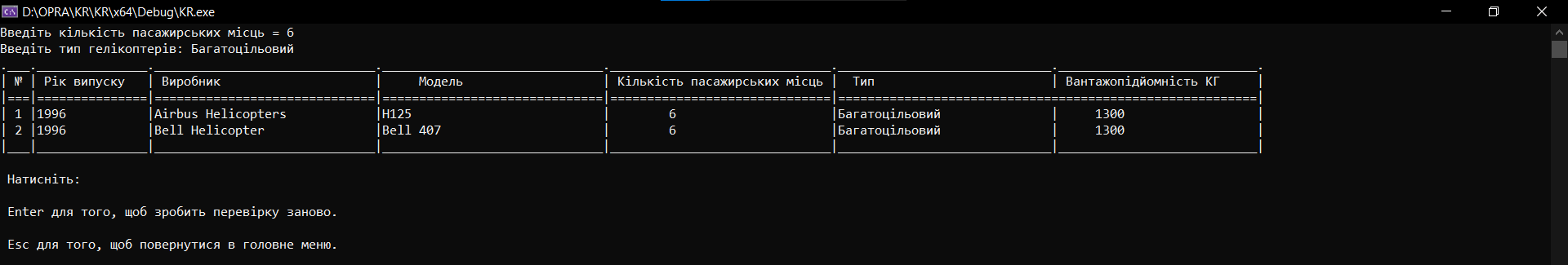
  
Сортування за виробником гелікоптерів



Сортування гелікоптерів, які випустили до/після заданого року



Сортування гелікоптерів, які випустили у заданий період



Сортування гелікоптерів із заданими кількістю пасажирських місць і типу



Вихід з програми

**Список використаних джерел**

**Книга:**

1. Герберт Ш. Полный справочник по С++ – перекладено з англійської: Видавницький дім “Вільямс”, 2006. – 800 с
2. Ярмуш, О. В. Інформатика i комп’ютерна техніка : навч. посiб. / О. В. Ярмуш, М. М. Редько. – К. : Вища освiта, 2006. – 359 с. Деннис, М. Р. Язык программирования С. / М. Р. Деннис, У. К. Брайан ; пер. с англ. – М. : «Вильямс», 2009.

**Електронні ресурси:**

1. Вільна енциклопедія – Режим доступу до ресурсу:

[https://](NULL) uk.wikipedia.org/wiki/.

1. Гелікоптери-Мілітарний – Режим доступу до ресурсу:

[https://](NULL)mil.in.ua/uk/tag/helikoptery/.