

Лабораторна робота № 18, 19. Динамічні масиви та динамічні списки

1 ВИМОГИ

1.1 Розробник

- Радєвич Владислав Романович;
- студент групи КІТ – 320;
- 15.04.2021 р.

1.2 Загальне завдання

Виконати надані завдання з лабораторних робіт 18 та 19 та розробити для них модульні тести.

1.3 Індивідуальне завдання

Розробити програму, умови якої надано у лабораторному практикуму. Мною було взято умови для розробки з розділу на оцінку «добре».

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення

Програма призначена для сортування структури даних заданих птахів з файлу та виведення результату у файл або на екран, використовуючи динамічні списки.

2.2 Опис логічної структури

2.2.1 Основна функція

`int main`

Призначення: головна функція.

Схема алгоритму функції подана на рис. 1.

Опис роботи: демонструє роботу заданого динамічного списку елементів структури та методів оперування ним, викликаючи функція для роботи з динамічним списком.

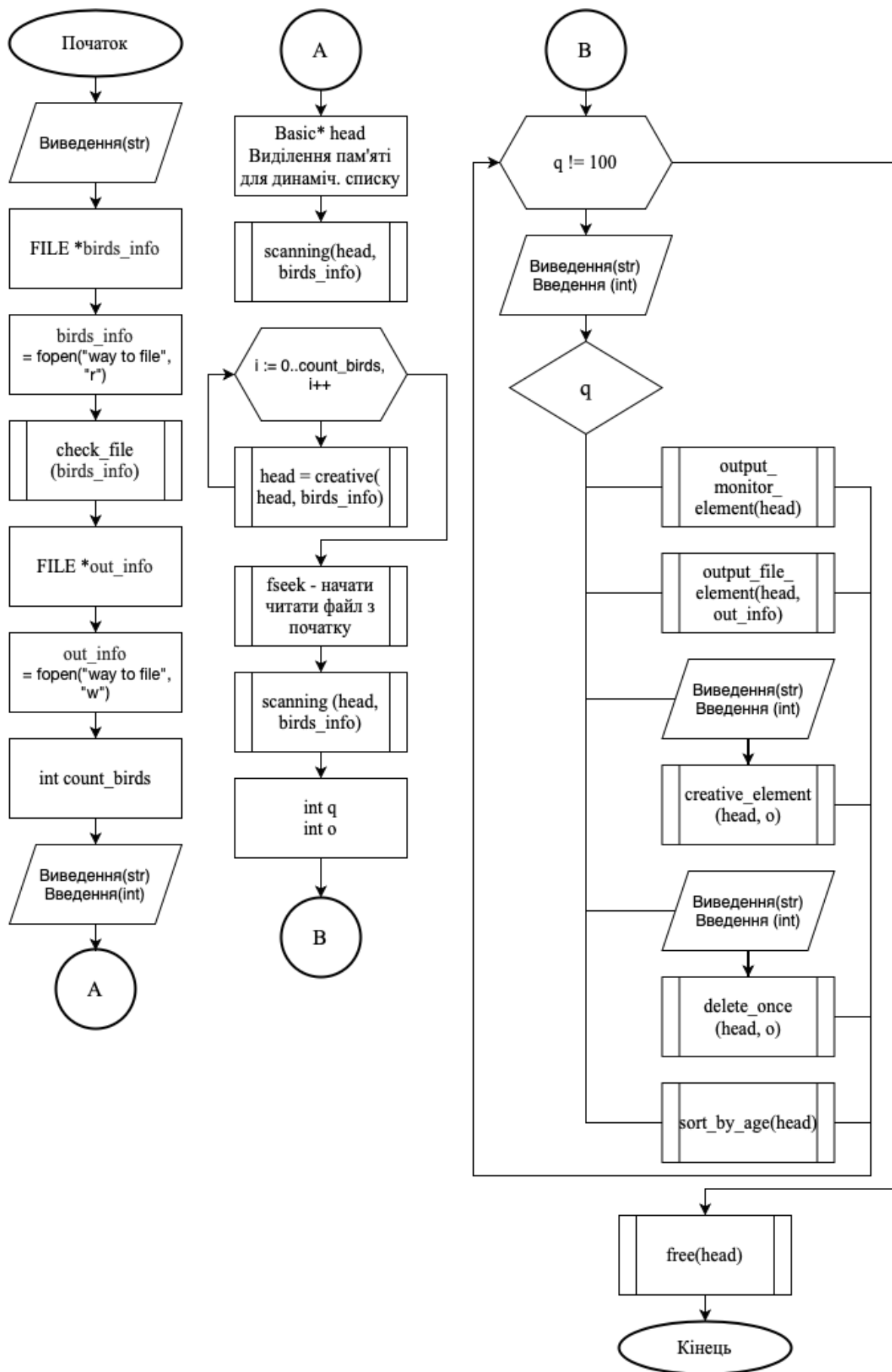


Рисунок 1 – Схема алгоритму функції main

2.2.2 Функція перевірки файлу

```
void check_file (FILE *fmatrix);
```

Призначення: перевірка місцезнаходження файлу по вказаному шляху.

Схема алгоритму функції подана на рис. 2

Опис роботи: функція перевіряє знаходження файлу з даними для структурами по заданій директорії.

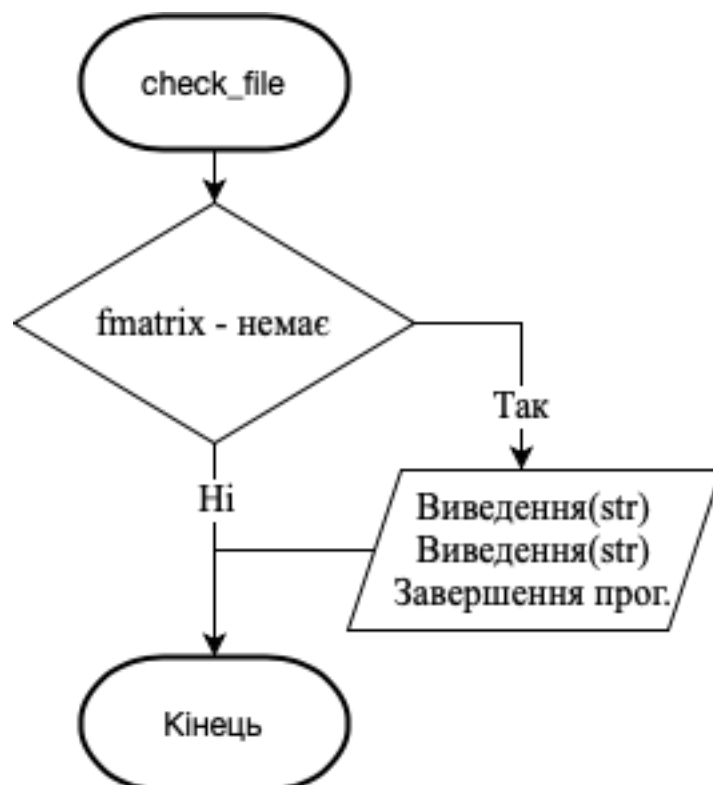


Рисунок 2 – Схема алгоритму функції `check_file`

2.2.3 Функція сканування даних

```
void scanning (Basic* p_bird, int count_birds, FILE *birds_info);
```

Призначення: сканування данні з файлу, в якому заздалегідь написані данні для структури.

Схема алгоритму функції подана на рис. 3.

Опис роботи: функція сканує та переносить данні з заданого файлу в задану структуру за допомогою циклу.

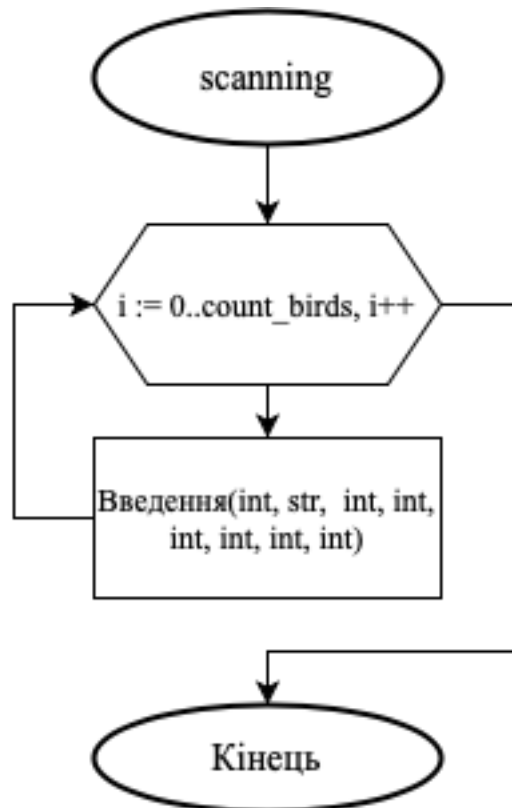


Рисунок 3 – Схема scanning

2.2.4 Функція виводу у файл

```
void output_file (Basic* p_bird, int count_birds, FILE* out_info);
```

Призначення: вивід даних у файл за певної директорією.

Схема алгоритму функції подана на рис. 4

Опис роботи: оброблення усіх полів структури та вивід усіх даних, що містяться в усіх структурах в певний файл.

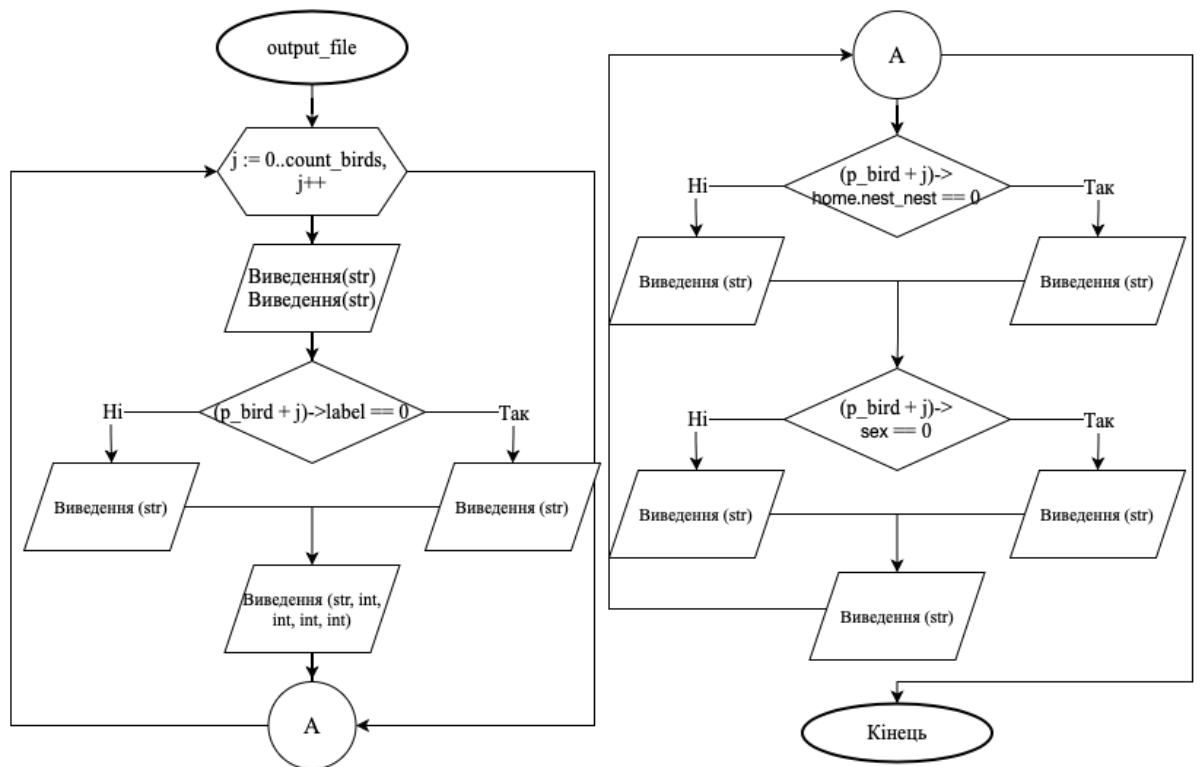


Рисунок 4 – схема алгоритму функції output_file

2.2.5 Функція виводу даних на екран

```
void output_monitor (Basic* p_bird, int count_birds);
```

Призначення: вивід окремих даних кожної структури з масиву структур на екран.

Схема алгоритму функції подана на рис. 5

Опис роботи: оброблення та вивід усіх даних кожної структури з зазначеного масиву структур.

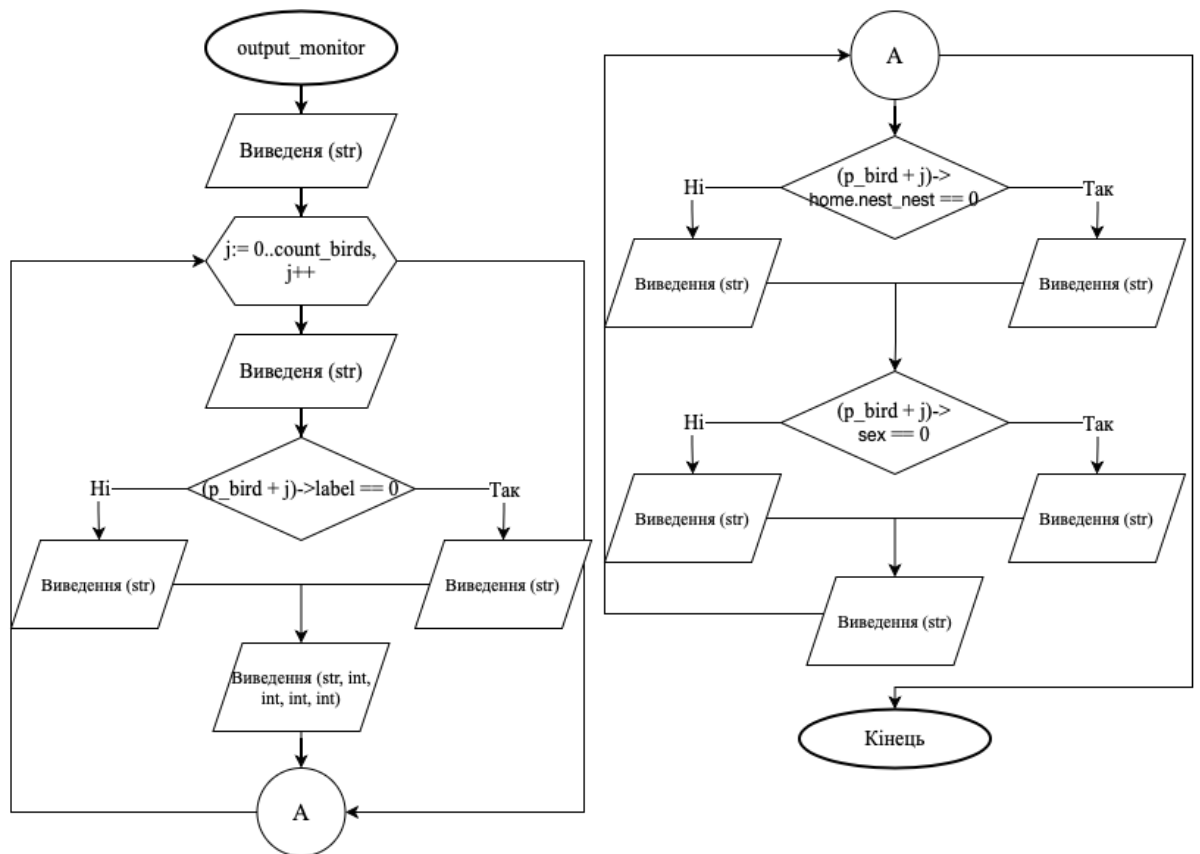


Рисунок 5 – схема алгоритму функції output_monitor

2.2.6 Функція вводу даних з файлу для списку

```
Basic* creative(Basic* head, FILE *birds_info);
```

Призначення: перепис даних з заданого файлу до динамічного списку елементів структури.

Схема алгоритму функції подана на рис. 6

Опис роботи: за допомогою функції scanning записуються дані з файлу до динамічного списку.

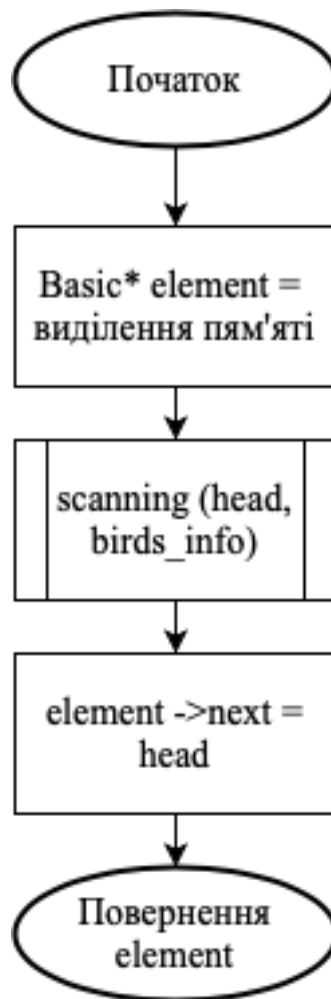


Рисунок 6 – схема алгоритму функції `creative`

2.2.7 Функція створення одного елементу

```
void creative_element(Basic* head, int position);
```

Призначення: додавання елементу структури до динамічного списку.

Схема алгоритму функції подана на рис. 7

Опис роботи: функція додає, у певну позицію, до заданого динамічного списку ще один елемент структури.

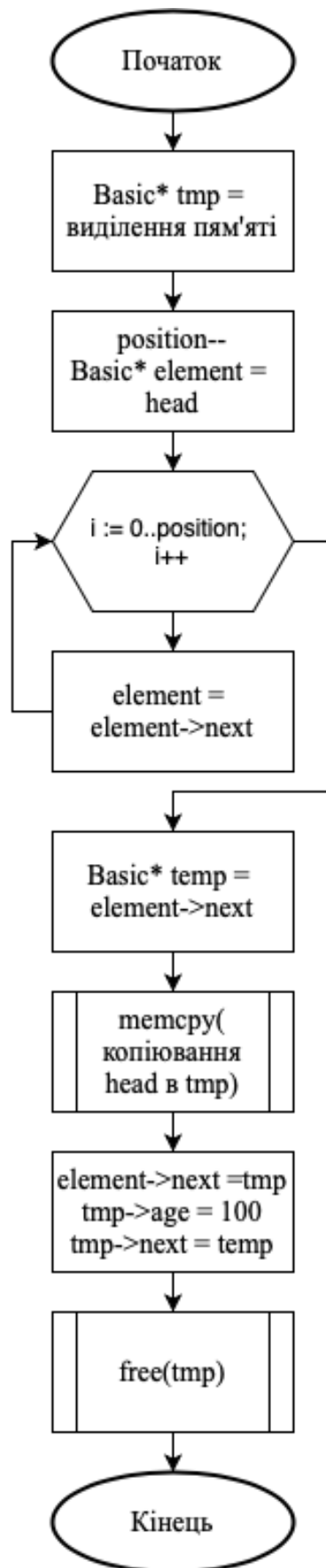


Рисунок 7 – схема алгоритму функції `creative_element`

2.2.8 Функція видалення елементу

```
void delete_once (Basic* head, int n);
```

Призначення: видалення елементу структури з динамічного списку.

Схема алгоритму функції подана на рис. 8

Опис роботи: функція видаляє певний елемент структури з динамічного списку елементів структури.

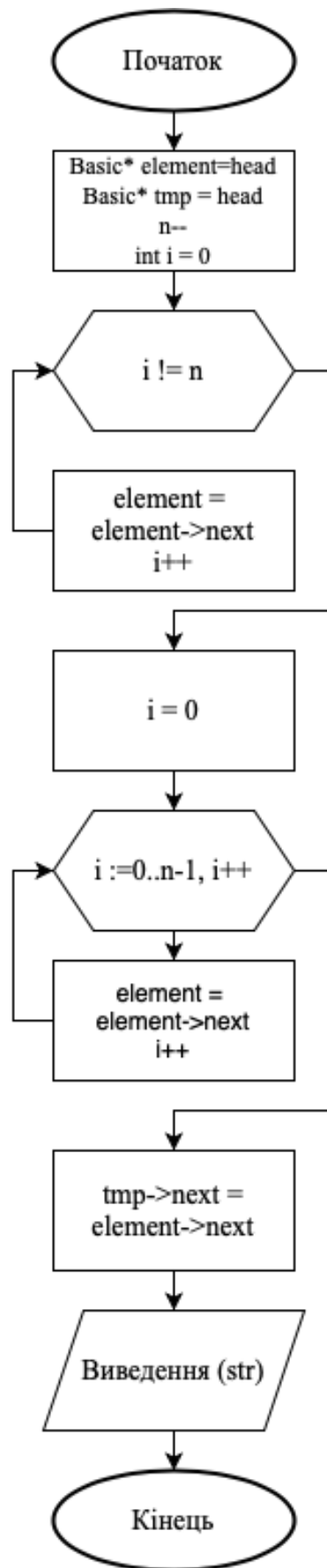


Рисунок 8 – схема алгоритму функції `delete_once`

2.2.9 Функція виводу у файл списку

```
void output_file_element (Basic* head, FILE *out_info);
```

Призначення: виведення у файл динамічний список.

Схема алгоритму функції подана на рис. 9

Опис роботи: виведення у заданий файл динамічного списки елементів структури.

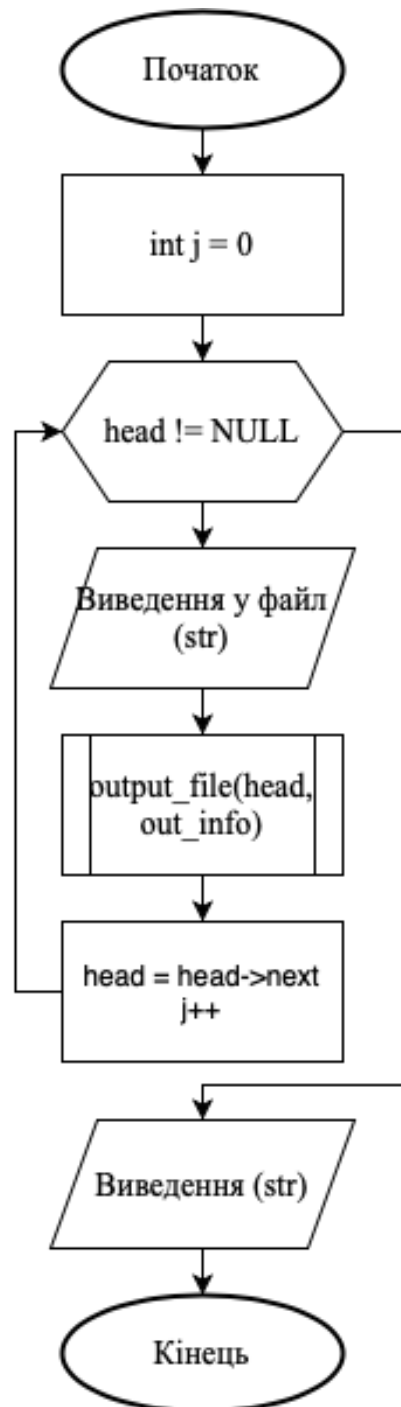


Рисунок 9 – схема алгоритму функції `output_file_element`

2.2.10 Функція виводу на екран для списку

```
void output_monitor_element (Basic* head);
```

Призначення: виведення на екран динамічний список.

Схема алгоритму функції подана на рис. 10

Опис роботи: виведення на екран динамічного списки елементів структури.

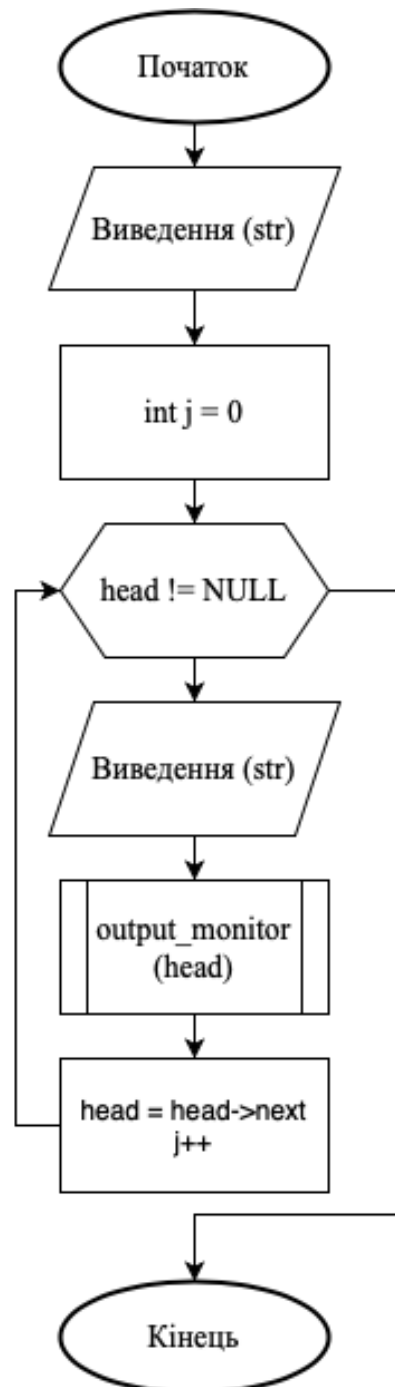


Рисунок 10 – схема алгоритму функції `output_monitor_element`

2.2.11 Функція сортування для списку

```
void sort_by_age (Basic* head);
```

Призначення: сортування динамічного списку.

Схема алгоритму функції подана на рис. 11

Опис роботи: сортування за відповідним критерієм (в цьому випадку за роками життя) динамічного списку елементів структури від меншого до більшого. Сортування виконується методом «бульбашки».

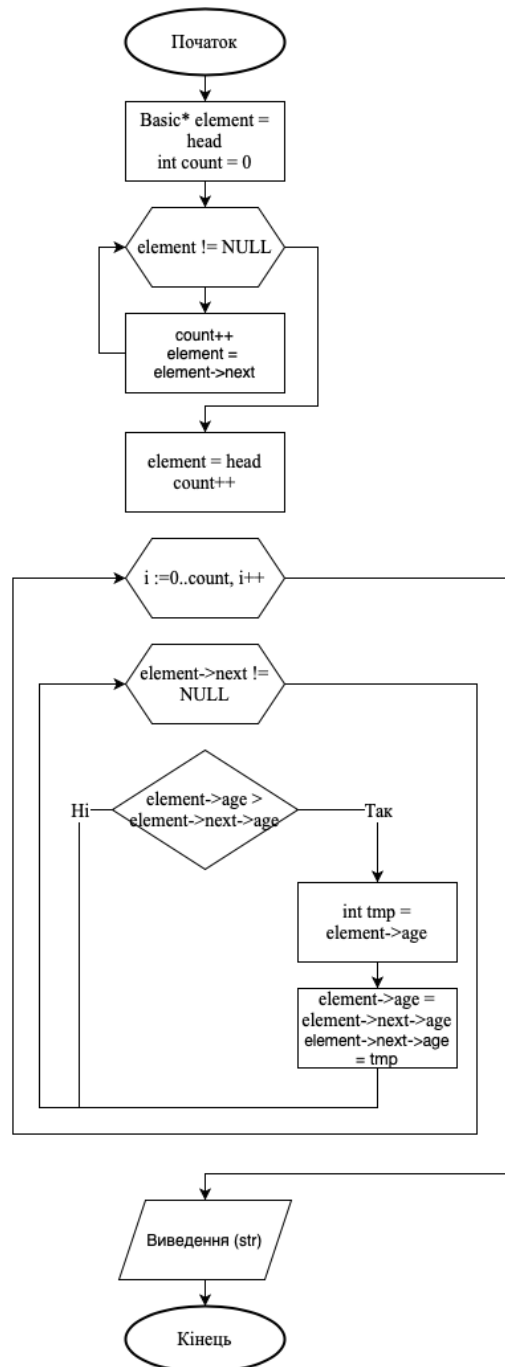
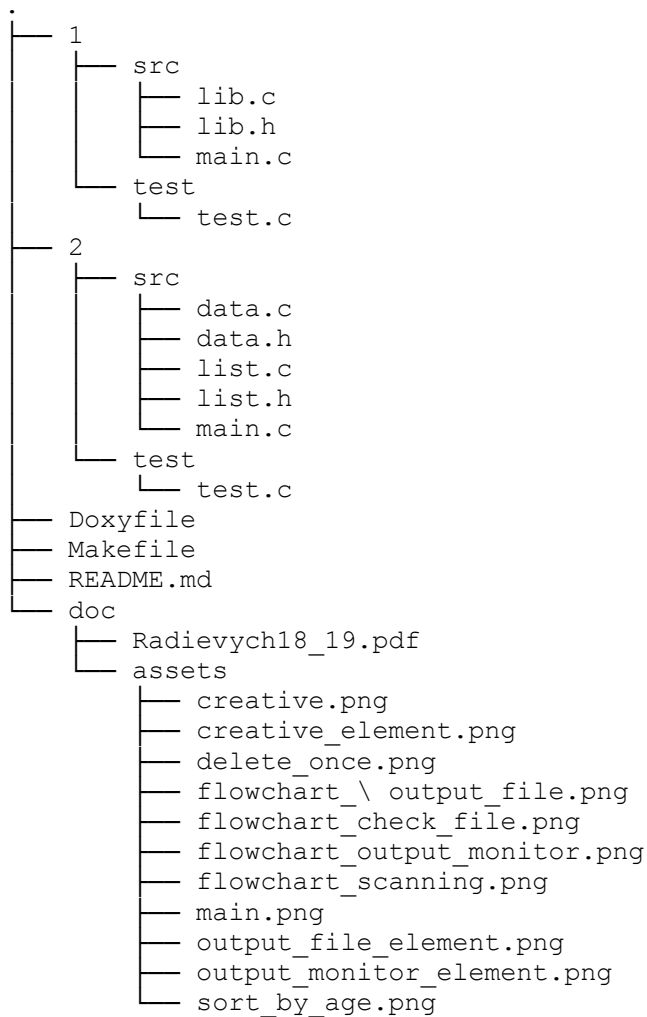


Рисунок 11 – схема алгоритму функції sort_by_age

2.3 Структура проекту



3 ВАРІАНТИ ВИКОРИСТАННЯ

Цю програму можна використовувати за для перепису усіх зареєстрованих птахів в окремий файл на комп'ютері або для заповнення списку, заздалегідь давши про них певну інформацію.

Результат роботи з doxygen продемонстровано на рисунку 12, рисунку 13 та рисунку 14, виконання модульних тестів на рисунку 15 та демонстрація відсутності витоків пам'яті на рисунку 16..

Lab18_19 1.0

Динамичні масиви та динамічні списки

Титульна сторінка	Додаткова інформація	Структури даних ▾	Файли ▾	Q* Пошук
Файли				
Повний список файлів.				
[Рівень елемента 1 2 3]				
▼ 1				
▼ src				
lib.c	Файл з реалізацією функцій			
lib.h	Файл з описом структури птахів, перерахуванням критеріїв птахів та функцій оперування птахами			
main.c	Файл з демонстрацією роботи функцій з використанням динамічних масивів			
▼ test				
test.c	Файл з тестами на реалізації функцій оперування структурами			
▼ 2				
▼ src				
data.c	Файл з реалізацією функцій для data.h			
data.h	Файл з описом структури птахів, перерахуванням критеріїв птахів та функцій оперування птахами			
list.c	Файл з реалізацією функцій для list.h			
list.h	Файл з описом структури птахів, перерахуванням критеріїв птахів та функцій оперування птахами			
main.c	Файл з демонстрацією роботи односпрямованого списку елементів розробленої структури Basic та методів оперування ним			
▼ test				
test.c	Файл з тестами на реалізації функцій оперування динамічним списком елементів структурами			

Створено системою **doxygen** 1.9.1

Рисунок 12 – робота з doxygen

Динамичні масиви та динамічні списки

Титульна сторінка

Додаткова інформація

Структури даних

Файли

Q*

Пошук

2

>

src

<

Функції

Файл list.h

Файл з описом структури птахів, перерахуванням критеріїв птахів та функцій оперування птахами

Детальніше...

```
#include "data.h"
#include <stdio.h>
#include <errno.h>
#include <stdbool.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
```

Діаграма включених заголовочних файлів для list.h:

Граф файлів, які включають цей файл:

Див. вихідні тексти.

Функції

Basic *

creative (Basic *head, FILE *birds_info)

Функція вводу даних з файлу

Детальніше...

void

creative_element (Basic *head, int position)

Функція створення одного елемента

Детальніше...

void

delete_once (Basic *head, int n)

Функція видалення елемента

Детальніше...

void

output_file_element (Basic *head, FILE *out_info)

Функція виводу у файл

Детальніше...

void

output_monitor_element (Basic *head)

Функція виводу на екран

Детальніше...

void

sort_by_age (Basic *head)

Функція сортування

Детальніше...

Рисунок 13 – робота з doxygen

Динамічні масиви та динамічні списки

Титульна сторінка

Додаткова інформація

Структури даних

Файли

Пошук

2

src

Структури даних | Визначення типів | Переліки | Функції

Файл data.h

Файл з описом структури птиць, перерахуванням критеріїв птиць та функцій оперування птахами

Детальніше...

#include <stdio.h>

Діаграма включених заголовочних файлів для data.h:

Граф файлів, які включають цей файл:

Див. вихідні тексти.

Структури даних

struct Feature

Структура домівки птаха

Детальніше...

struct Basic

Структура "Птах".

Детальніше...

Визначення типів

typedef struct Basic Basic

Структура "Птах".

Детальніше...

Переліки

enum Yes_no { Так , Ні , Так , Ні }

Так або ні

Детальніше...

enum Sex { Чоловіча , Жіноча , Чоловіча , Жіноча }

Стать птиці

Детальніше...

Функції

void check_file (FILE *fmatrix)

Функція перевірки файлу

Детальніше...

Рисунок 15 – робота з doxygen

```

whatislove@MacBook-Air-Vladislav dist % ./test2.bin
Запуск тесту test_creative_element ...
Елемент додано до списку!

Запуск тесту test_delete_once ...
Елемент видалено!

Запуск тесту test_sort_by_age ...
Дані відсортовано!

Модульні тести пройдено успішно!%
whatislove@MacBook-Air-Vladislav dist % █
  
```

Рисунок 15 – робота з модульними тестами

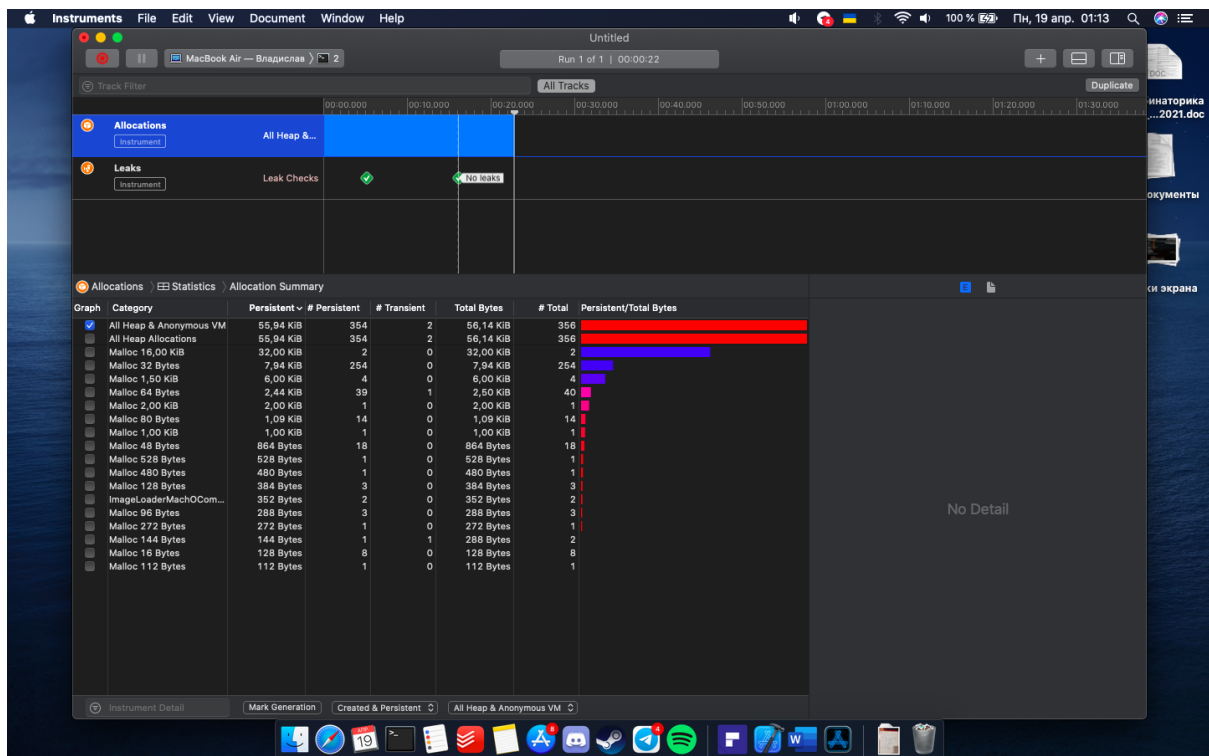


Рисунок 16 – демонстрація відсутності витоків пам'яті

4 ВИСНОВОК

При виконанні даної лабораторної роботи я закріпив набуті мною навички, створення та взаємодію з динамічними масивами та динамічними структурами.

Посилання на GitHub, де знаходяться усі програми:
https://github.com/KotKHPI/Programming_Radievych