НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2»

*Звіт з лабораторної роботи №18-19*

*Тема: «Динамічні масиви та динамічні списки»*

Виконав:

ст. гр. КІТ-120А

Старовойтов Н.А.

Перевірив:

Челак В.В.

Харків – 2021

***Мета:*** Отримати навички роботи із динамічними масивами та динамічними списками.

***Індивідуальне завдання***

Робота на оцінку “відмінно”.

Лабораторна робота №18:

1. Розробити функцію, яка реалізує вставку в рядок “s” другий рядок “s2” в “і”-у позицію рядка “s”.
2. Розробити функцію видалення з рядка “s” усіх символів з індексами в заданому діапазоні.
3. За допомогою функцій memcpy, memset створити функції додання та видалення елементів з динамічного масиву вашої прикладної області.
4. Додати модульні тести, що демонструють коректність розроблених функцій

Лабораторна робота №19:

1. Розробити функцію, яка зчитує список з файлу
2. Розробити функцію, яка записує дані(список) у файлу;
3. Розробити функцію, яка виводить список на екран;
4. Розробити функцію, яка буде сортувати список за заданим критерієм;
5. Розробити функцію, яка буде знаходити заданий елемент у списку;
6. Розробити функцію, яка видаляє елемент зі списку;
7. Розробити функцію, яка вставляє новий елемент по індексу;
8. Реалізувати діалоговий режим спілкування з користувачем

**Опис програми**



Рисунок 1 — Cтруктура Bird



Рисунок 2 — Всі функції лабораторної роботи №18



Рисунок 3 — Всі функції лабораторної роботи №19

**Схеми алгоритмів функцій**

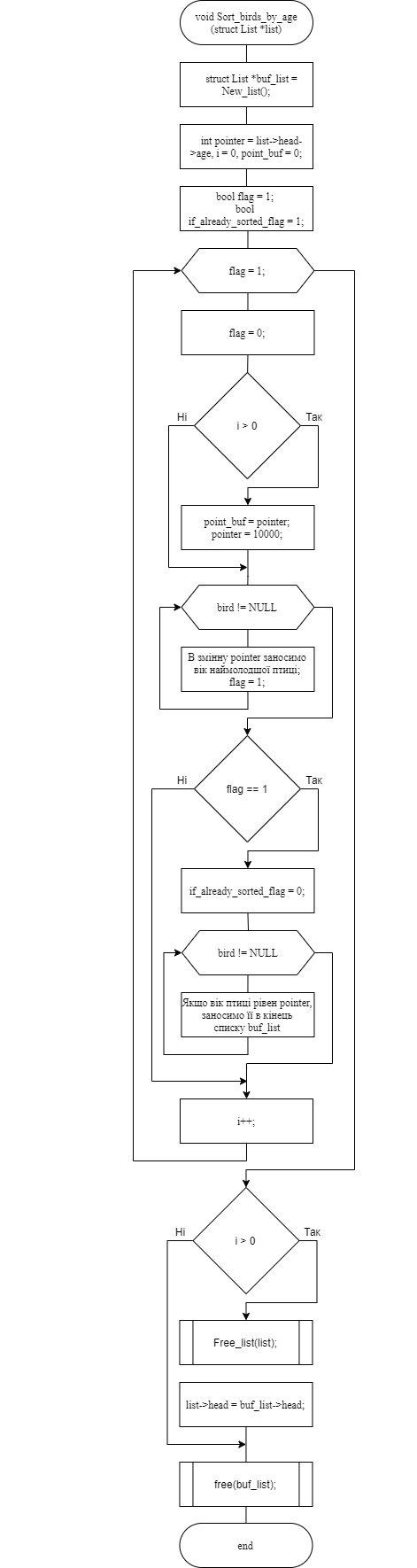


Рисунок 4 — Блок-схема функції Sort\_birds\_by\_age

**Текст програми**

Лабораторна робота №18:

#include "lib.h"

int main() {

Demonstration\_of\_strings();

struct Bird \*birds = malloc(N \* sizeof(struct Bird));

read\_from\_file(birds);

random\_age\_generation(birds);

print\_in\_console(birds, N);

struct Bird paste\_bird = {1, "Some\_bird", 27, 48, 49, 2, 0, Male};

int position, a;

printf("Select add/remove (1/0): ");

scanf("%d", &a);

if (a == 1) {

printf("Enter a position to paste: ");

scanf("%d", &position);

birds = add\_bird(birds, paste\_bird, position);

printf("\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

print\_in\_console(birds, N + 1);

}

if (a == 0) {

printf("Enter a position to remove: ");

scanf("%d", &position);

birds = delete\_bird(birds, position);

printf("\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

print\_in\_console(birds, N - 1);

}

free(birds);

return 0;

}

Лабораторна робота №19:

#include "list.h"

#include "data.h"

int main() {

system("chcp 65001");

printf("\n");

struct List \*list = New\_list();

struct Bird struct\_paste\_bird = {0, "SOME\_BIRD", 27, 250, 40, 2, 0, 1};

struct Bird \*paste\_bird = malloc(sizeof(struct Bird));

\*paste\_bird = struct\_paste\_bird;

Read\_list\_from\_file(list);

while (true) {

int action;

printf("\n|--------------------------------------------------------------------|\n");

printf(" Выберите действие: \n");

printf("\t1. Вывести список на экран\n");

printf("\t2. Вывести список в файл\n");

printf("\t3. Вставить новый элемент в список\n");

printf("\t4. Удалить элемент из списка\n");

printf("\t5. Найти в списке элемент\n");

printf("\t6. Сортировать список по возрасту птицы\n");

printf("\t7. Выход\n");

printf("|--------------------------------------------------------------------|\n\n");

printf(" --> ");

scanf("%d", &action);

switch (action) {

case 1:

printf("\nСписок:\n\n");

Show\_list(list);

break;

case 2:

Print\_list\_in\_file(list);

printf("\n Список занесен в файл 'result.txt'\n\n");

break;

case 3:

printf("\nЭлемент, который будет вставлен: \n\n");

Print\_bird(paste\_bird);

printf("\nВведите позицию, на которую нужно вставить элемент: \n");

printf(" --> ");

int pos;

scanf("%d", &pos);

Insert\_element(list, pos, paste\_bird);

printf("\nЭлемент вставлен\n\n\n");

break;

case 4:

printf("\nВведите позицию элемента, который будет удалён: \n");

printf(" --> ");

int num;

scanf("%d", &num);

Delete\_element(list, num);

printf("\nЭлемент удалён\n\n\n");

break;

case 5:

printf("\nВведите критерий, по которому искать птицу: \n");

printf("\t1. С кольцом или без\n");

printf("\t2. Имя\n");

printf("\t3. Возраст\n");

printf("\t4. Площадь скворечника\n");

printf("\t5. Высота скворечника\n");

printf("\t6. Кол-во кормушек скворечника\n");

printf("\t7. Является ли скворечник гнездом\n");

printf("\t8. Пол птицы\n");

printf(" --> ");

int number;

scanf("%d", &number);

printf("\n");

Find\_bird(list, number);

break;

case 6:

Sort\_birds\_by\_age(list);

printf("\nСписок отсортирован \n");

break;

case 7:

Free\_list(list);

free(list);

printf("\nЗавершение работы программы...\n");

return 0;

default:

printf("\nERROR: Вы ввели некорректный вариант\n");

break;

}

}

}

**Результати роботи програми**

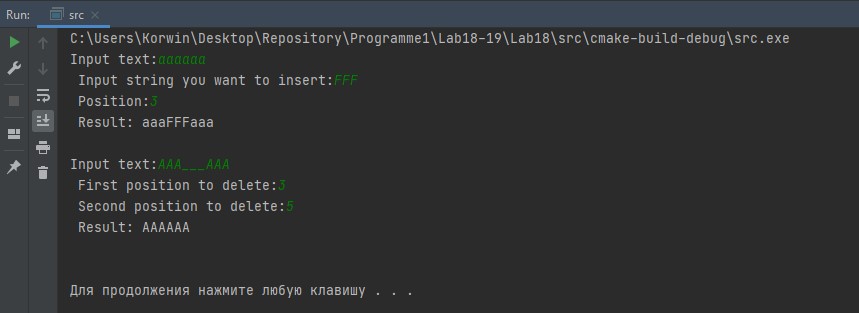


Рисунок 5 — Результат виконання перших двох завдань лабораторної роботи №18

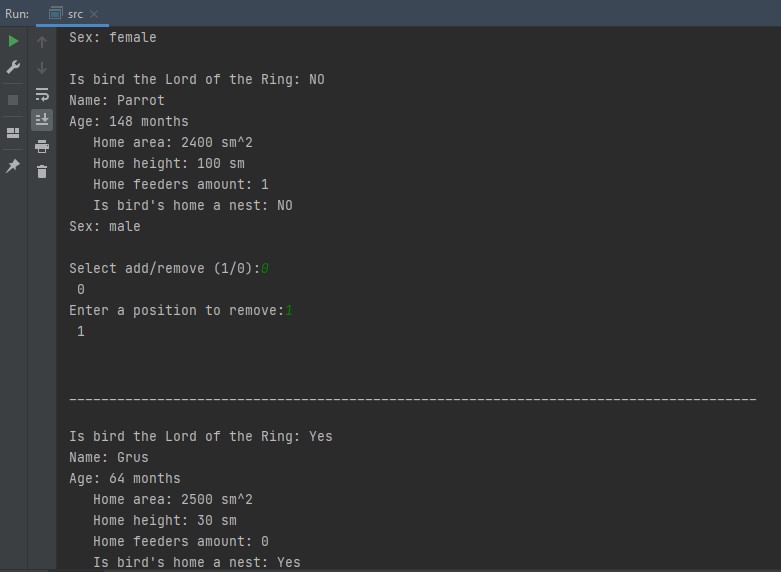


Рисунок 6 — Результат виконання третього завдання лабораторної роботи №18

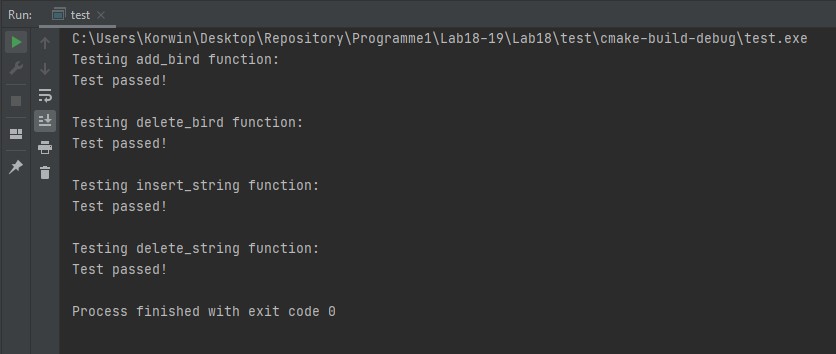


Рисунок 7 — Результат успішного виконання тестів лабораторної роботи №18

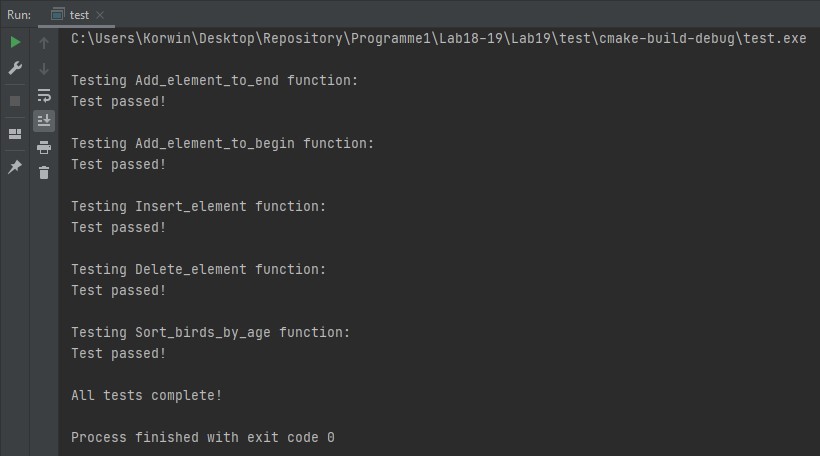


Рисунок 8 — Результат успішного виконання тестів лабораторної роботи №19

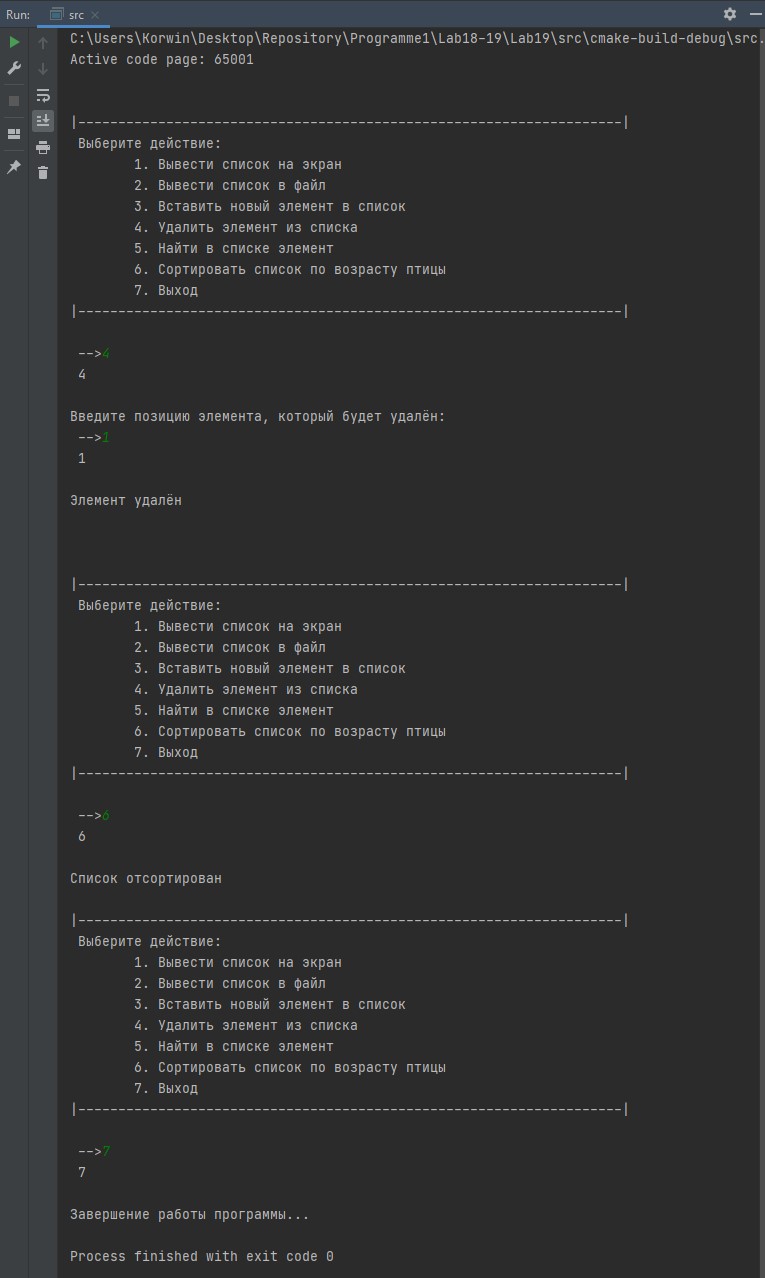


Рисунок 9 — Результат успішного виконання програми лабораторної роботи №19

**Висновки**

Під час виконання даної лабораторної роботи було отримано навички роботи із динамічними масивами та динамічними списками.