Міністерство освіти і науки України

Дніпропетровський національний університет

імені Олеся Гончара

ФФЕКС

Кафедра ЕОМ

**Звіз з лабораторної роботи № 2**

Виконав: студент 2 курсу

Шейко Р.О.

Науковий керівник:

Викладач кафедри EOM

Спірінцева Ольга Володимирівна

м. Дніпро

2022

Завдання 1

Визначити структурний тип, що містить такі поля:

-символьний масив, що використовується для зберігання рядка, наприклад, з ім'ям студента,

-покажчик на тип char – для організації динамічного масиву, що зберігає рядок, наприклад, із прізвищем студента.

Інші поля вибрати за варіантами, наведеними в таблиці 2.

Використовувати одну із змінних зберігання деякого ідентифікатора (номера);

покажчик не символьний тип – для організації динамічного масиву цілих чи плаваючих чисел;

іншу змінну – зберігання розміру цього масиву.

Доповнити структурний тип будь-якими полями на свій вибір.

Використовувати одну із змінних зберігання деякого ідентифікатора (номера); покажчик не символьний тип – для організації динамічного масиву цілих чи плаваючих чисел; іншу змінну – зберігання розміру цього масиву.

Доповнити структурний тип будь-якими полями на свій вибір.

2. Визначити функції:

ініціалізації структури;

заповнення масиву чисел;

виведення на екран масиву чисел;

введення інформації до рядків імені та прізвища та інші поля;

виведення на екран усіх полів структури, крім масиву чисел.

функцію звільнення динамічної пам'яті.

У половини функцій, на вибір студента, одним з аргументів має бути покажчик на структурний тип, в іншої половини – посилання на структурний тип.

3. Визначити функцію main(), у якій створити:

o об'єкт раніше визначеного структурного типу

o покажчик цього структурний тип.

За допомогою покажчика створити динамічний масив об'єктів структурного типу із 3-х – 4-х елементів.

Для об'єкта послідовно викликати функції ініціалізації, заповнення масиву чисел, введення даних до інших полів, показу масиву, показу полів.

Для кожного елемента масиву структур виконати у циклі (for) функції ініціалізації, заповнення масиву та введення даних.

Вивести на екран вміст полів кожного елемента масиву структур у циклі (for) за допомогою відповідних функцій.

Наприкінці функції main() викликати функцію звільнення пам'яті об'єкта структурного типу й у циклі кожного елемента масиву об'єктів.

Видалити динамічний масив.

Хід виконання програми контролювати за повідомленнями, що виводяться на екран.

Зберегти файл з текстом програми для наступної роботи.

Пояснювальна записка до коду та її функцій

//Тип даних:

typedef struct {

char Name[N];

char\* SecondName;

float Number;//nomer//Переменная плавающего типа.

unsigned long SizeOfMassive;//Переменная беззнакового длинного целого типа;

long\* Massive;//Указатель на длинный целый тип;

} List;

// Функція перепису з масиву b до масиву a

void writeToFrom(char a[], char b[]) {

for (int i = 0; i < strlen(b) + 1; i++)

a[i] = b[i];

}

// Функція, яка зберігає сейвово цифру, що була введена користувачем

void safeScanf(int\* a) {

char temp[N];

int error = 0;

do {

error = 0;

scanf\_s("%s", &temp, N - 1);

for (int a = 0; a < strlen(temp); a++)

if (atoi(temp) == 0 && !(('0' <= temp[a]) && (temp[a] <= '9')))

error = 1;

if (error == 1)

printf("Вводите только числа!\n");

else {

\*a = atoi(temp);

break;

}

} while (1);

}

//Функція повертає тип даних List ,який потім записується

List TakeElement() {

List list;

printf("\nВведите название: ");

scanf\_s("%s", &list.Name, N);

char SName[N];

printf("\nВведите фамилию: ");

scanf\_s("%s", &SName, N - 1);

list.SecondName = (char\*)malloc((strlen(SName) + 1)\* sizeof(char));

writeToFrom(list.SecondName, SName);

printf("\nВведите номер: ");

scanf\_s("%f", &list.Number);

list.SizeOfMassive = 0;

return list;

}

//Функція повертає масив List кількості dimension.

List\* initArray(int dimension) {

List\* massive = (List\*)malloc(dimension \* sizeof(List));

if (massive == NULL)

return NULL;

for(int i = 0; i < dimension; i++)

massive[i] = TakeElement();

return massive;

}

//Фукція відповідає за визов функції, що виводить info одного екземпляра

void ViewInfo(List\* lists, int dimension) {

for(int i = 0; i < dimension; i++)

ViewInfoElement(i, lists[i]);

}

// Фукція виводить info одного екземпляра

void ViewInfoElement(int i, List list ) {

printf("%d | %s - %s - %f\n", i, list.Name, list.SecondName, list.Number);

}

//Функція відповідає за запис у масив чисел вибраного елемента числа

List Change(List list\_) {

List list;

int choose = 0;

do {

printf("Ввведите количество цифр, что хотите записать в массив: ");

safeScanf(&choose);

if (choose >= 1)

break;

} while (1);

list.SizeOfMassive = (long)choose;

list.Massive = (long\*)malloc(list.SizeOfMassive \* sizeof(long));

for (int i = 0, num = 0; i < list.SizeOfMassive; i++) {

safeScanf(&num);

list.Massive[i] = num;

}

return list;

}

//Функція відповідає за вибір елементу для зміни його масиву чисел long

void ChooseChangeMassiveInLists(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для изменений: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

lists[choosed] = Change(lists[choosed]);

}

// Функція виводу масиву чисел елемента, що був переданий

void MassiveOutPut(List list) {

puts("Массив чисел: ");

for (int i = 0; i < list.SizeOfMassive; i++)

printf("%ld ", list.Massive[i]);

}

//Функція вибору елементу для виводу масиву чисел long

void ChooseMassiveOutPut(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для вывода: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

if (lists[choosed].SizeOfMassive > 0)

MassiveOutPut(lists[choosed]);

else puts("Пустой массив чисел.");

}

//Функція вибору елементу

void ChooseElementToChange(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для изменений: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

lists[choosed] = TakeElement();

}

//Функція для вибору розмірності масиву(не використовується)

int TakeDimension() {

int chooseNum = 0;

do {

safeScanf(&chooseNum);

if (chooseNum > 0 && chooseNum < 10)

break;

else if (chooseNum >= 10)

printf("\nНе нужно делать больше 10, неудобно потом будет)");

} while (1);

return chooseNum;

}

// головна функція програми.

void main()

{

setlocale(LC\_ALL, "Rus");

List\* massiveStruct = NULL;

int choose = 0;

int dimension = 4;

do {

puts("\nМеню: \

\n1. Инициализация структуры;\

\n2. Заполнение массива чисел;\

\n3. Вывода на экран массива чисел;\

\n4. Ввода информации в строки имени и фамилии и другие поля;\

\n5. Bывода на экран всех полей структуры, кроме массива чиcел;\

\n6. Функцию освобождения динамической памяти;\

\n0. Exit");

safeScanf(&choose);

if (choose == 1) {

if (massiveStruct == NULL)

massiveStruct = initArray(dimension);

else puts("Массив не равен NULL!");

}

else if (choose == 2 && massiveStruct != NULL)

ChooseChangeMassiveInLists(massiveStruct, dimension);

else if (choose == 3 && massiveStruct != NULL)

ChooseMassiveOutPut(massiveStruct, dimension);

else if (choose == 4 && massiveStruct != NULL)

ChooseElementToChange(massiveStruct, dimension);

else if (choose == 5 && massiveStruct != NULL)

ViewInfo(massiveStruct, dimension);

else if (choose == 6)

{

free(massiveStruct);//очищення пам'яті

puts("Память очищена!");

massiveStruct = NULL;

}

} while (choose!= 0);

}

Ілюстрація роботи

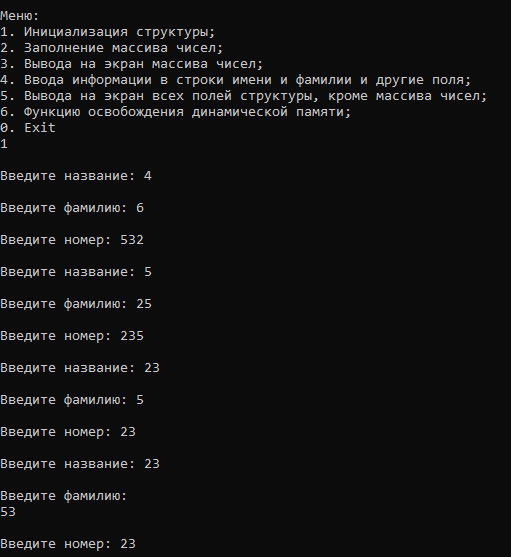


Рис 1.

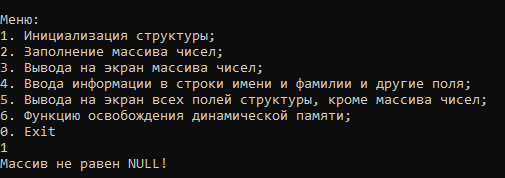


Рис 2. (Якщо ми заповнили масив, то функція не може спрацювати)

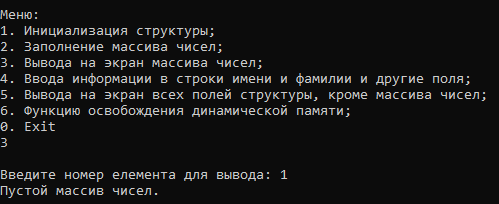


Рис 3. Якщо масив чисел пустий, то не можливо вивести його цифри.

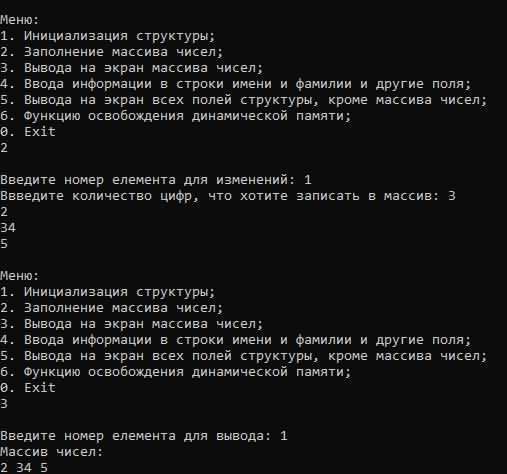


Рис 4. Масив можливо вивести

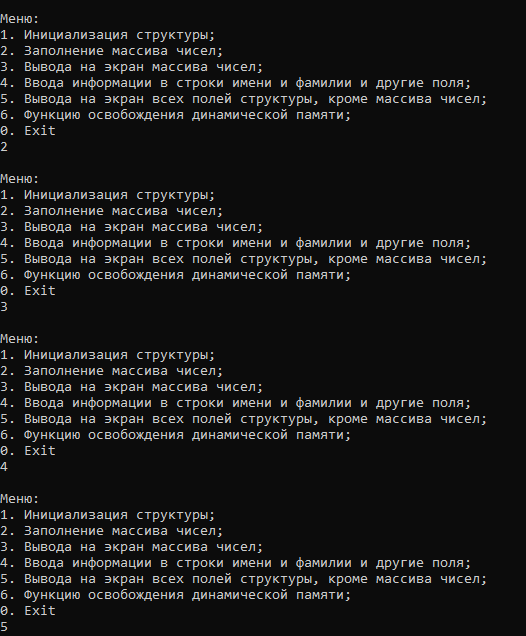


Рис 5. Якщо масив не ініцілізовано, то робота пунктів 2, 3, 4, 5 паралізовано.

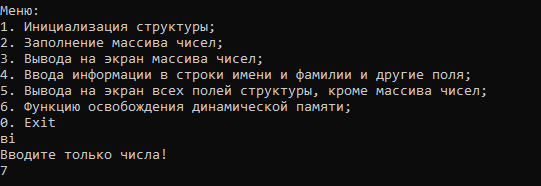


Рис 6. Обробка некорректного вибору пункту меню

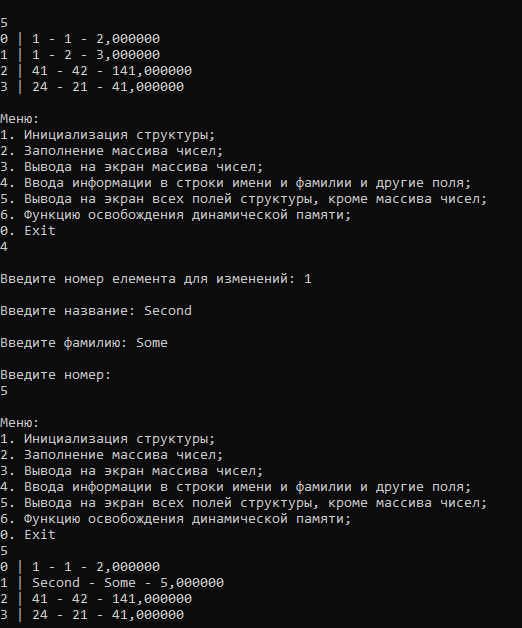


Рис 7. Ілюстрація зміни 1 елементу масиву структур.

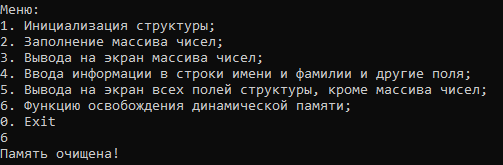


Рис 8. Пам'ять очищена!

Висновок: Структури даних дуже вдобний інструмент для манюпулювання даними.

Лістинг з лабораторної роботи. Файл <main.c>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <locale.h>

#define N 50

typedef struct {

char Name[N];

char\* SecondName;

float Number;//nomer//Переменная плаваю- щего типа.

unsigned long SizeOfMassive;//Переменная беззнакового длинного целого типа;

long\* Massive;//Указатель на длинный целый тип;

} List;

void writeToFrom(char a[], char b[]) {

for (int i = 0; i < strlen(b) + 1; i++)

a[i] = b[i];

}

void safeScanf(int\* a) {

char temp[N];

int error = 0;

do {

error = 0;

scanf\_s("%s", &temp, N - 1);

for (int a = 0; a < strlen(temp); a++)

if (atoi(temp) == 0 && !(('0' <= temp[a]) && (temp[a] <= '9')))

error = 1;

if (error == 1)

printf("Вводите только числа!\n");

else {

\*a = atoi(temp);

break;

}

} while (1);

}

List TakeElement() {

List list;

printf("\nВведите название: ");

scanf\_s("%s", &list.Name, N);

char SName[N];

printf("\nВведите фамилию: ");

scanf\_s("%s", &SName, N - 1);

list.SecondName = (char\*)malloc((strlen(SName) + 1)\* sizeof(char));

writeToFrom(list.SecondName, SName);

printf("\nВведите номер: ");

scanf\_s("%f", &list.Number);

list.SizeOfMassive = 0;

return list;

}

List\* initArray(int dimension) {

List\* massive = (List\*)malloc(dimension \* sizeof(List));

if (massive == NULL)

return NULL;

for(int i = 0; i < dimension; i++)

massive[i] = TakeElement();

return massive;

}

void ViewInfo(List\* lists, int dimension) {

for(int i = 0; i < dimension; i++)

ViewInfoElement(i, lists[i]);

}

void ViewInfoElement(int i, List list ) {

printf("%d | %s - %s - %f\n", i, list.Name, list.SecondName, list.Number);

}

List Change(List list\_) {

List list;

int choose = 0;

do {

printf("Ввведите количество цифр, что хотите записать в массив: ");

safeScanf(&choose);

if (choose >= 1)

break;

} while (1);

list.SizeOfMassive = (long)choose;

list.Massive = (long\*)malloc(list.SizeOfMassive \* sizeof(long));

for (int i = 0, num = 0; i < list.SizeOfMassive; i++) {

safeScanf(&num);

list.Massive[i] = num;

}

return list;

}

void ChooseChangeMassiveInLists(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для изменений: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

lists[choosed] = Change(lists[choosed]);

}

void MassiveOutPut(List list) {

puts("Массив чисел: ");

for (int i = 0; i < list.SizeOfMassive; i++)

printf("%ld ", list.Massive[i]);

}

void ChooseMassiveOutPut(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для вывода: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

if (lists[choosed].SizeOfMassive > 0)

MassiveOutPut(lists[choosed]);

else puts("Пустой массив чисел.");

}

void ChooseElementToChange(List\* lists, int dimension) {

int choosed = 0;

do {

printf("\nВведите номер елемента для изменений: ");

safeScanf(&choosed);

if (choosed >= 0 && choosed < dimension)

break;

} while (1);

lists[choosed] = TakeElement();

}

int TakeDimension() {

int chooseNum = 0;

do {

safeScanf(&chooseNum);

if (chooseNum > 0 && chooseNum < 10)

break;

else if (chooseNum >= 10)

printf("\nНе нужно делать больше 10, неудобно потом будет)");

} while (1);

return chooseNum;

}

void main()

{

setlocale(LC\_ALL, "Rus");

List\* massiveStruct = NULL;

int choose = 0;

int dimension = 4;

do {

puts("\nМеню: \

\n1. Инициализация структуры;\

\n2. Заполнение массива чисел;\

\n3. Вывода на экран массива чисел;\

\n4. Ввода информации в строки имени и фамилии и другие поля;\

\n5. Bывода на экран всех полей структуры, кроме массива чиcел;\

\n6. Функцию освобождения динамической памяти;\

\n0. Exit");

safeScanf(&choose);

if (choose == 1) {

if (massiveStruct == NULL)

massiveStruct = initArray(dimension);

else puts("Массив не равен NULL!");

}

else if (choose == 2 && massiveStruct != NULL)

ChooseChangeMassiveInLists(massiveStruct, dimension);

else if (choose == 3 && massiveStruct != NULL)

ChooseMassiveOutPut(massiveStruct, dimension);

else if (choose == 4 && massiveStruct != NULL)

ChooseElementToChange(massiveStruct, dimension);

else if (choose == 5 && massiveStruct != NULL)

ViewInfo(massiveStruct, dimension);

else if (choose == 6)

{

free(massiveStruct);//очищення пам'яті

puts("Память очищена!");

massiveStruct = NULL;

}

} while (choose!= 0);

}