**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут: **ІКНІ**

Кафедра: **Пз**

**Звіт**

До лабораторної роботи № 10

**На тему: “** ДИНАМІЧНІ СТРУКТУРИ ДАНИХ **”**

**З дисципліни:** «Основи програмування»

**Лектор:**

ст. викл. каф. ПЗ

Муха.Т.О

**Виконав:**

Ст. гр. ПЗ-15

Бабіля О.О.

**Прийняла:**

Асист. каф. Пз

Заводовська Н.О

08.12.2021

∑= \_\_\_\_ .

Львів – 2021

**Тема** **роботи**: ДИНАМІЧНІ СТРУКТУРИ ДАНИХ

**Мета** **роботи**: – оволодіти практичними прийомами створення та опрацювання динамічних списків

**Індивідуальне завдання**

Виконати завдання з лабораторної роботи № 8, організувавши послідовність структур в однозв’язний список. Реалізувати операцію вставки нового елемента у відсортований список і операцію вилучення зі списку даних, які відповідають одній з наступних умов:

8) про студентів, які не мають оцінок 4 і 5;

**Теоритичні відомості**

Зчитав студентів, просорутвав , вивів , зчитав нового , просортвував ,та знову вивів, за допомогою різних функцій.

**Код програми**

Source.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define N 500

#include "struct.h"

#include "function.h"

int main()

{

int month = 12, year = 2021, avarage,n;

struct list student[6];

Read(student, 5);

avarage=age(student, month, year);

printf(" The sorted group without changes:\n\n");

OUT(student,5);

ReadNew(student,5);

printf(" The unsorted group with new member:\n\n");

OUT(student, 6);

printf(" The sorted group with changes:\n\n");

sort(student, 6);

OUT(student, 6);

struct list\* pListHead = NULL;

pListHead = &student[0];

pListHead->next = &student[1];

for (int k = 1; k < 6; k++)

{

if (k == 5)

{

student[k].next = NULL;

}

else

{

student[k].next = &student[k + 1];

}

}

AdvOut(pListHead);

}

**Function.c**

**#define** \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define N 500

#include "struct.h"

#include "function.h"

int Read(struct list student[], int n)

{

int i = 0, a, b, c, d1, d2, d3;

FILE\* fp = fopen("list.txt", "r");

if (fp == NULL)

{

printf("Cannot open file.\n");

return 0;

}

while (!feof(fp))

{

fscanf(fp, "%s %s %d %d %d %d %d %d", student[i].name, student[i].surname, &a, &b, &c, &d1, &d2, &d3);

student[i].mark1 = a;

student[i].mark2 = b;

student[i].mark3 = c;

student[i].Bday = d1;

student[i].Bmonth = d2;

student[i].Byear = d3;

i++;

}

fclose(fp);

return 1;

}

int age(struct list student[], int month, int year)

{

int age, avg = 0, count = 0;

for (int i = 0; i < 5; i++)

{

if (student[i].Bmonth > month) age = year - student[i].Byear;

else age = year - student[i].Byear - 1;

student[i].age = age;

}

for (int i = 0; i < 5; i++)

{

avg += student[i].age;

count++;

}

return avg = avg / count;

}

void OUT(struct list student[], int n)

{

printf("| Surname | Name | Marks | Birthday|\n");

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

{

printf("|%-24s|\t%-40s|\t%d %d %d |\t%d.%d.%d|\n", student[i].name, student[i].surname, student[i].mark1, student[i].mark2, student[i].mark3, student[i].Bday, student[i].Bmonth, student[i].Byear);

}

}

void ReadNew(struct list student[], int n)

{

printf("Would you like to add someone?\nType 0 for NO or \nEnter name of the student:\n");

scanf\_s("%d", &n);

if (n == 0) return 0;

else {

gets(student[n].surname);

printf("Enter Surname of the student:");

gets(student[n].name);

int a, b, c, d1, d2, d3;

printf("Enter 3 marks of the student:");

scanf\_s("%d %d %d", &student[n].mark1, &student[n].mark1, &student[n].mark3);

printf("Enter birth date of the student:");

scanf\_s("%d %d %d", &student[n].Bday, &student[n].Bmonth, &student[n].Byear);

}

}

void sort(struct list student[], int n)

{

char temp[100];

char temp1[100];

int a, b, c, d, e, f;

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

{

for (int j = i + 1; j <= n; j++) {

if (strcmp(student[i].name, student[j].name) > 0)

{

strcpy(temp, student[i].name);

strcpy(temp1, student[i].surname);

a = student[i].mark1;

b = student[i].mark2;

c = student[i].mark3;

d = student[i].Bday;

e = student[i].Bmonth;

f = student[i].Byear;

strcpy(student[i].name, student[j].name);

strcpy(student[i].surname, student[j].surname);

student[i].mark1 = student[j].mark1;

student[i].mark2 = student[j].mark2;

student[i].mark3 = student[j].mark3;

student[i].Bday = student[j].Bday;

student[i].Bmonth = student[j].Bmonth;

student[i].Byear = student[j].Byear;

strcpy(student[j].name, temp);

strcpy(student[j].surname, temp1);

student[j].mark1 = a;

student[j].mark2 = b;

student[j].mark3 = c;

student[j].Bday = d;

student[j].Bmonth = e;

student[j].Byear = f;

}

}

}

}

void AdvOut(struct list\* pListHead)

{

struct list\* pTemp = pListHead;

printf(" The sorted group with changes and deletes:\n\n");

printf("| Surname | Name | Marks | Birthday|\n");

while (pTemp != NULL) {

if ((pTemp->mark1 == 3 || pTemp->mark1 == 2) && (pTemp->mark2 == 3 || pTemp->mark2 == 2) && (pTemp->mark3 == 3 || pTemp->mark3 == 2))

printf("|%-24s|\t%-40s|\t%d %d %d |\t%d.%d.%d|\n", pTemp->name, pTemp->surname, pTemp->mark1, pTemp->mark2, pTemp->mark3, pTemp->Bday, pTemp->Bmonth, pTemp->Byear);

pTemp = pTemp->next;

}

}

**Function.h**

int Read(struct list [], int );

int age(struct list[], int, int);

void OUT(struct list[], int );

void ReadNew(struct list[], int );

void sort(struct list[], int );

void AdvOut(struct list \*);

**Struct.h**

**struct list**

{

char name[200];

char surname[220];

int mark1;

int mark2;

int mark3;

int Bday;

int Byear;

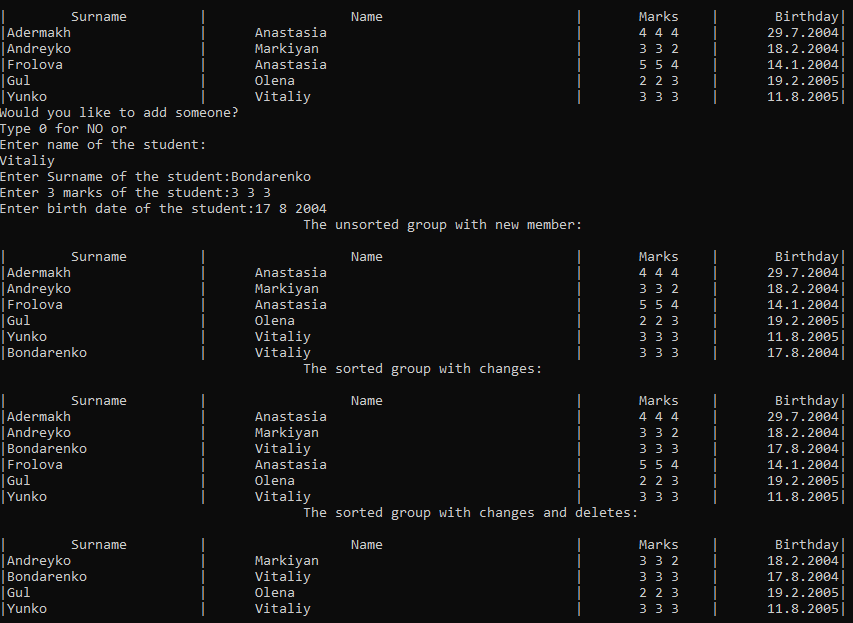
int Bmonth;

int age;

struct list\* next;

**};**

**Результат виконання програми**

****

**Висновки:** В ході даної лабораторної роботи я оволодів практичними прийомами створення та опрацювання динамічних списків