

## Лабораторна робота №10

### Тема: Розробка та реалізація програм із використанням лінійних динамічних списків.

Ціль роботи: отримати навички створення та реалізації програм, що використовують структури лінійного динамічного списку та реалізують операції з ними.

#### Завдання до лабораторної роботи

**Завдання 10.1.** Запишіть рядки, які будуть виведені на екран дисплею внаслідок виконання фрагментів, поданих у варіантах (табл. 10.1), при наступному початку програми (Вказівка: замість N підставити номер варіанта за списком групи)

```
#include <cstdlib>
#include <cstdio>
#include <cstring>
struct lnk{
    char name[10];
    int ph;
    lnk *next;
};
int main(){
int i, nr; short int n; short int *k, *p;
lnk *cR,*fst,*p1; char *nAr[3]; int pAr[3];
scanf ("%hd", &n); k=&n; p=k; *p=*p+2;
printf ("%p %hd\n", k, *p);
nAr[0]="AAA"; nAr[1]="BBBB"; nAr[2]="CCCC";
pAr[0]=2222; pAr[1]=333; pAr[2]=4444;
nr=sizeof (lnk);
fst=NULL;
for (i=0; i<3; i++){
    cR = (lnk *) malloc(sizeof(lnk)*1);
    strcpy(cR->name, nAr[i]);
    cR->ph=pAr[i];
    cR->next=fst; fst=cR;
}
```

Таблиця 10.1 – Варіанти завдання 10.1

№	Фрагмент програми
1-5	<pre>fst-&gt;next=fst-&gt;next-&gt;next;  printf("%d \n",cR-&gt;ph); cR=fst;  while (cR!=NULL){      printf("%s %d\n",cR-&gt;name,cR-&gt;ph);      cR=cR-&gt;next;  }  printf("nr=%d",nr);  printf(" %s\n",fst-&gt;name);</pre>
6-10	<pre>fst=fst-&gt;next;  printf("%d \n",cR-&gt;ph);cR=fst;  while ( cR!=NULL){  printf("%s %d \n",cR-&gt;name,cR-&gt;ph);  cR=cR-&gt;next;  }  printf("nr=%d",nr);  printf(" %s\n",fst-&gt;name);</pre>
11-15	<pre>p1=(lnk *) malloc(sizeof(lnk)*1);  strcpy(p1-&gt;name,nAr[0]);p1-&gt;ph=pAr[0];  p1-&gt;next=fst-&gt;next; fst-&gt;next=p1;  printf("%d\n",cR-&gt;ph);cR=fst;  for (i=2;i&gt;0;i--){      printf("%s %d\n",cR-&gt;name,cR-&gt;ph);      cR=cR-&gt;next;  }</pre>

	<pre>printf("nr=%d",nr); printf("  %s\n",fst-&gt;name);</pre>
16-20	<pre>p1=(lnk *) malloc(sizeof(lnk)*1); strcpy(p1-&gt;name,nAr[1]);p1-&gt;ph=pAr[1]; p1-&gt;next=fst-&gt;next; fst-&gt;next=p1; printf("%d\n",cR-&gt;ph);cR=fst; for (i=0;i&lt;3;i++){     printf("%s  %d\n",cR-&gt;name,cR-&gt;ph);     cR=cR-&gt;next; } printf("nr=%d",nr); printf("  %s\n",fst-&gt;name);</pre>
21-25	<pre>printf("%d\n",cR-&gt;ph);cR=fst; for (i=1;i&lt;3;i++){     printf("%s  %d\n",cR-&gt;name,cR-&gt;ph);     cR=cR-&gt;next; } printf("nr=%d",nr); printf("  %s\n",fst-&gt;name);</pre>
26-30	<pre>printf("%d\n",cR-&gt;ph);cR=fst; for (i=2;i&gt;=0;i--){     printf("%s  %d\n",cR-&gt;name,cR-&gt;ph);     cR=cR-&gt;next; } printf("nr=%d",nr); printf("  %s\n",fst-&gt;name);</pre>

**Завдання 10.2. Створити лінійний динамічний список на основі структур та файлів, які були створені в процесі виконання лабораторної роботи №9.**

**Завдання 10.3. Створити функцію для виведення інформації про елемент лінійного списку за вказівником на нього та виконати пункт (d) із використанням такої функції. Також створити функції додавання запису у список після вказаного елемента та перед вказаним елементом.**

**Реалізувати можливість видалення вказаного елемента.**

1. **Student:** id, Прізвище, Ім'я, По батькові, Дата народження, Адреса, Телефон, Факультет, Курс, Група.

Створити лінійний динамічний список структур. Вивести:

- a) список студентів вказаного факультету;
- b) список студентів, що народились після вказаного року;
- c) список навчальної групи;
- d) Повну інформацію про першого знайденого студента із вказаним прізвищем.

2. **Customer:** id, Прізвище, Ім'я, По батькові, Адреса, Номер кредитної картки, Номер банківського рахунку.

Створити лінійний динамічний список структур. Вивести:

- a) список покупців;
- b) список покупців, у яких номер кредитної картки знаходиться в заданому інтервалі;
- c) список покупців, чия у яких номер банківського рахунку закінчується на вказану цифру;
- d) повну інформацію про покупця із вказаним номером кредитної картки.

3. **Patient:** id, Прізвище, Ім'я, По батькові, Адреса, Телефон, Номер медичної картки, Діагноз.

Створити лінійний динамічний список структур. Вивести:

- a) список пацієнтів, що мають вказаний діагноз;
- b) список пацієнтів чий номер медичної картки при діленні на 7 дасть вказану остачу;
- c) список пацієнтів, номер медичної карти яких знаходиться в заданому інтервалі;
- d) повну інформацію про пацієнта із вказаним прізвищем та номером телефону.

4. **Abiturient**: id, Прізвище, Ім'я, По батькові, Адреса, Телефон, Оцінки.

Створити лінійний динамічний список структур. Вивести:

- a) список абітурієнтів, що мають незадовільні оцінки;
- b) список абітурієнтів, у яких сума балів вище заданої;
- c) вибрати вказану кількість n абітурієнтів, що мають найбільшу суму балів;
- d) повну інформацію про абітурієнта за вказаними Прізвищем, ім'ям та по батькові.

5. **Book**: id, Назва, Автор (и), Видавництво, Рік видання, Кількість сторінок, Ціна, Тип папітурки.

Створити лінійний динамічний список структур. Вивести:

- a) список книг заданого автора;
- b) список книг, що видані вказаним видавництвом;
- c) список книг, що видані після заданого року;
- d) повну інформацію про книгу із вказаним id.

6. **House**: id, Номер квартири, Площа, Поверх, Кількість кімнат, Вулиця, Тип будівлі, Термін експлуатації.

Створити лінійний динамічний список структур. Вивести:

- a) список квартир, які мають задану кількість кімнат;
- b) список квартир, що мають вказану кількість кімнат і розташованих між вказаними поверхами;
- c) список квартир, які мають площу, що більше заданої;
- d) повну інформацію про квартиру із вказаним id.

7. **Phone**: id, Прізвище, Ім'я, По батькові, Адреса, Номер кредитної картки, Дебет, Кредит, Час

міських та міжнародних розмов.

Створити лінійний динамічний список структур. Вивести:

- a) відомості про всіх абонентів;
- b) відомості про абонентів, які користувались міжміським зв'язком;
- c) відомості про абонентів, у яких час міських розмов перевищує вказаний;
- d) повну інформацію про абонента із вказаним номером кредитної картки

8. **Car**: id, Марка, Модель, Рік випуску, Колір, Ціна, Реєстраційний номер.

Створити лінійний динамічний список структур. Вивести:

- a) список автомобілів заданої марки;
- b) список автомобілів заданої моделі, які експлуатуються більше  $n$  років;
- c) список автомобілів вказаного року випуску, ціна яких більше вказаної;
- d) повну інформацію про автомобіль із вказаним реєстраційним номером.

9. **Product**: id, Найменування, Тип, Виробник, Ціна, Термін зберігання, Кількість.

Створити лінійний динамічний список структур. Вивести:

- a) список товарів заданого найменування;
- b) список товарів заданого найменування, ціна яких не більше заданої;
- c) список товарів, термін зберігання яких більше заданого;
- d) повну інформацію про товар із вказаним id.

10. **Train**: Пункт призначення, Номер поїзда, Час відправлення, Число місць (загальних, плацкарт, купе, люкс).

Створити лінійний динамічний список структур. Вивести:

- a) список поїздів, які прямують до заданого пункту призначення;
- b) список поїздів, які прямують до заданого пункту призначення та відправляються після вказаної години;
- c) список поїздів, які відправляються до заданого пункту призначення та мають загальні місця;
- d) повну інформацію про поїзд за його номером

### Приклад виконання роботи

**Завдання 10.1.** Визначити дію фрагмента програми

```
printf("%s\n", cR->name); p1=fst->next;
while (p1->next){
    printf("%s  %d\n", p1->name, p1->ph);
    p1=p1->next;
}
printf("nr=%d", nr);
printf(" %s  %d\n", p1->name, p1->ph);
```

## Розв'язання

Цей фрагмент програми виводить на екран у першому рядку вміст змінної  $k$ , а саме адресу, та значення за покажчиком  $p$ , що вказує на  $n$ . У другому рядку виводиться значення інформаційного поля останнього елементу списку, далі виводяться значення двох інформаційних полів попереднього елементу. В останньому рядку наведено розмір елементу списку, та значення полів першого елементу списку.

```
0028FF02 6
CCCC
BBBB 333
nr=20 AAA 2222
```

**Завдання 10.2.** Розробити програму, яка оброблює лінійний список: вивести на екран всі елементи, що більші за середнє арифметичне значення.

## Розв'язання

### 1. Постановка задачі

Розробити програму, яка оброблює лінійний список: видалити всі елементи, що більші за середнє арифметичне значення.

### 2. Алгоритм розв'язання задачі

Алгоритм розв'язання задачі можна представити у вигляді такої послідовності дій:

Дія 1. Створити список.

Дія 2. Вивести список на екран.

Дія 3. Знайти середнє арифметичне значення елементів списку.

Дія 4. Повторювати *доки не кінець списку*:

Дія 4.1. Знайти адресу елементу списку, значення інформаційного поля якого більше за середнє арифметичне значення;

Дія 4.2. Якщо адреса *не нуль*, тоді вивести на екран елемент списку за цією адресою;

Дія 4.3. Якщо адреса *не нуль*, тоді перейти до наступного елементу списку;

## Дія 5. Видалити весь список.

### 3. Текст програми

```
#include <iostream>
#include <stdlib.h>
using namespace std;
struct element {
    int x;
    element *next;
};
void add(int x, element *head){
    element * t = new element;
    t->x = x;
    t->next = head->next;
    head->next = t;
}
void print(element *head) {
    cout << "-----" << endl;
    element * t = head->next;
    while (t != NULL) {
        cout << t->x << " ";
        t = t->next;
    }
    cout << endl << "-----" << endl;
}
element * findAbove(element *head, float value) {
    element *p;
    p = head;
    while(p && (p->x <= value)) {
        p = p->next;
    }
    return p;
}
float average(element *head) {
    element *p;
    int sum=0;int k=0;
```



```

        p = head->next;
while (p) {
    sum+=p->x;k++;
    p=p->next;
}
return float(sum)/k;
}
void deleteList(element *head) {
while (head->next) {
    element * t = head->next;
    head->next = t->next;
    delete t;
}
}

int main() {
element *head = new element;
head->next=NULL;
int k;
for (int i=0;i<5;i++){
    cin>>k;
    add(k, head);
}
print(head);
float av=average(head);
cout<<"\n average="<<av<< endl;
element *temp=head->next;
while (temp!=NULL){
    temp = findAbove(temp, av);
    if (temp) cout<<"\n aver >"<<temp->x<< endl;
    if (temp) temp=temp->next;
}

deleteList(head);
return 0;
}

```

#### 4. Результати роботи програми

-----

4 2 8 8 4 9 0 4 7 1

-----

average=4.7

-----

4 2 4 0 4 1

-----