

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра системного програмування і спеціальних комп’ютерних систем

**Лабораторна робота №6**

з дисципліни ***“ СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ - 2.  
СКЛАДНІ СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ”***

Тема: «**Зв`язані динамічні структури даних. Списки**»

**Варіант № 6**

Виконав:

студент 1 курсу ФПМ

групи КВ-41:

Горпинич-Радуженко

Іван Олександрович

**Перевірено:**

Київ 2015

*Мета лабораторної роботи*

Метою лабораторної роботи №2.6 є засвоєння теоретичного матеріалу та набуття практичного досвіду використання зв’язаних динамічних структур даних у вигляді одно- та двозв ’язаних списків при складанні різних алгоритмів.

*Постановка задачі*

1. Створити список з n (n>0) елементів (n вводиться з клавіатури), якщо інша кількість елементів не вказана у конкретному завданні,  
 2. Тип ключів (інформаційних полів) задано за варіантом.  
3. Значення елементів списку взяти самостійно такими, щоб можна було продемонструвати коректність роботи алгоритму програми. Введення значень елементів списку можна виконати довільним способом (випадкові числа, формування значень за формулою, введення з файлу чи з клавіатури).  
4. Вид списку (черга, стек, дек, прямий однозв’язний лінійний список, обернений однозв’язний лінійний список, двозв’язний лінійний список, однозв’язний кільцевий список, двозв’язний кільцевий список,) вибрати самостійно з метою найбільш доцільного рішення поставленої за варіантом задачі.  
5. Виконати над створеним списком дії, вказані за варіантом, та коректне звільнення пам’яті списку.  
6. При виконанні заданих дій, виводі значень елементів та звільненні пам’яті списку вважати, що довжина списку (кількість елементів n чи 2n) невідома на момент виконання цих дій.  
7. Повторювані частини алгоритму необхідно оформити у вигляді процедур або функцій (для створення, обробки, виведення та звільнення пам’яті списків) з передачею списку за допомогою параметра(ів).

***Завдання за варіантом***

Варіант 6

Ключами елементів списку є дійсні числа. Відсортувати елементи списку за незбільшенням, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом обміну (“бульбашки”) з використанням «прапорця».

***Код програми***

**program** LABO6;

**type**

TPtr = ^TElem;

TElem = **record**

Inf: integer;

Link: TPtr;

**end**;

**var**

BegL, EndL: Tptr;

n, k: integer;

**procedure** input(n: integer; **var** BegL, EndL: Tptr);

**var**

p: Tptr;

i: integer;

**begin**

BegL := nil;EndL := nil;

**new**(p);

p^.Inf := random(n);

p^.Link := nil;

BegL := p;

EndL := p;

**for** i := 2 **to** n **do**

**begin**

**new**(p);

p^.Inf := random(n);

p^.Link := nil;

EndL^.Link := p;

EndL := p;

**end**;

**end**;

**procedure** output(**var** BegL, EndL: Tptr);

**var**

p: Tptr;

**begin**

p := BegL;

**while** p <> nil **do**

**begin**

write(p^.inf:5);

p := p^.Link;

**end**;

**end**;

**procedure** sort(**var** BegL, EndL: Tptr);

**var**

f: boolean;

p: Tptr;

value: integer;

**begin**

f := true;

**while**(f = true) **do**

**begin**

f := false;p := BegL;

**while** p^.Link <> nil **do**

**begin**

**if** p^.Inf < p^.Link^.Inf **then**

**begin**

value := p^.Inf;

p^.Inf := p^.Link^.Inf;

p^.Link^.Inf := value;

f := true;

**end**;

p := p^.Link;

**end**;

**end**;

**end**;

**procedure** clean(**var** BegL, EndL: Tptr);

**var**

p: Tptr;

**begin**

**while** BegL^.Link <> nil **do**

**begin**

p := BegL;BegL := BegL^.Link;

**if** BegL = nil **then** EndL := nil;

dispose(p);

**end**;

**end**;

**begin**

write('Vvedite kolichestvo elementov v spiske:');

readln(n);

input(n, BegL, EndL);

output(BegL, EndL);

sort(BegL, EndL);

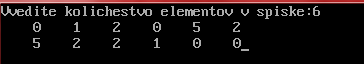
writeln;

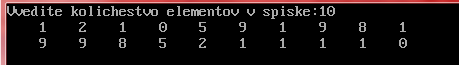
output(BegL, EndL);

clean(BegL, EndL);

**end**.

***Тестування програми***







*Контрольні питання*

1. Чи існують обмеження на кількість елементів у списку, якщо так, то які?
2. Які є види спискових структур даних з точки зору їх логічного використання (стек, черга, тощо). Поясніть їх особливості та відмінності між ними.
3. Які є вимоги до структури елемента зв’язних динамічних даних?
4. В чому полягає особливість опису типів для створення зв’язних динамічних даних?
5. Якою повинна бути структура елементу лінійного двозв’язного списку?
6. Скільки вказівників і якого призначення необхідно для роботи з чергою?
7. Скільки вказівників і якого призначення необхідно для

роботи з стеком?

1. Скільки вказівників і якого призначення необхідно для

роботи з деком?

1. Скільки вказівників і якого призначення необхідно для

роботи з лінійними однозв’язними списками?

1. Скільки вказівників і якого призначення необхідно для

роботи з лінійними двозв’язними списками?