Лабораторна робота № 5

«Операції з линійними списками»

Теоретичні відомості та рекомендації:

Вказівник – тип даних, який описує адресу змінної іншого типу в пам`яті. Розмір вказівника залежить лише від того, в ближній чи дальній зоні пам`яті розташована змінна. Тип даних, на які вказує вказівник визначає розмір блоку пам`яті, який записується або читається за вказаною адресою (напр. *int*, char*, far double**).

До вказівників застосовуються дві основні операції:

- адресування

```
int a; int* b; b = &a;
```

- разадресація

```
int a; int* b = &a;  *b = 3;
```

Структура – тип даних, який складається з кількох полів різних типів, розташованих у певному порядку. Розмір структури дорівнює сумі розмірів полів. Поля структури мають власні імена. Доступ до полів здійснюється через точку для змінної типа структура, та через знак –> для вказівника на структуру. Наприклад:

```
struct pair { int first; int second; };
struct pair x, *p;
x.first = 1; x.second = 2;
p = &x;  p->first = 3;
```

Список – це структура, що містить один чи більше вказівників на таку саму структуру (сусідів по списку). Списки бувають:

- линійні однозв’язні – містять один вказівник на наступний елемент списку
- линійні двозв’язні – містять два вказівники на наступний та попередній елемент списку
- линійні кільцьові – вказівники першого та останнього елементу об’єднані
- дерева – містять кілька вказівників на сусідні (похідні) елементи дерева.

Списки передбачають такі основні операції:

- ітерування – переміщення окремо виділеного вказівника між елементами списку шляхом присвоювання йому значення вказівника на сусіда відповідного елементу списка

- додавання елементу у кінець, початок чи середину списку – відбувається шляхом виділення пам’яті під нову структуру елементу і присвоювання вказівника на неї відповідному вказівнуку на сусіда елемента списку
- вилучення елементу з початку, кінця чи середини – відбуувається шляхом переприсвоювання вказівників на сусідів сусідніх елементів списку, та звільнення пам’яті елементу, що видаляється
- перестановка елементів у списку – відбувається шляхом переприсвоювання вказівників на сусідів відповідно до бажаного порядку

Приклади для линійного однозв’язного списку:

```
typedef struct _list { double data; struct _list *next; } list;
list *head = (list*)malloc(sizeof(list));
head->data = 0;
// тут вказівник head зберігає адресу першого елеемнту списку,
// для однозв’язного списку рухатись можна тільки у одному напрямку
//// додавання п’яти елементів:
list *x = head;
for(i=0; i<5; i++){
      x->next = (list*)malloc(sizeof(list));
      x->next->data = i;
      x = x->next;
}
//// ітерування списком та друк елементів
x = head;
while(x){
      printf(“%f\n”,x->data);
      x = x->next;
}
//// обмін другого та третього елементів
x = head->next->next;
list *y = head;
y->next->next = x->next;
x->next = y->next;
y->next = x;
//// видалення третього елементу
x = head->next->next;
y = x->next;
x->next = y->next;
free(y);
```

Завдання:

1. Описати структуру *vector*, що складається з реальних полів *x,y,z* та є елементом однозв'язного списку бо містить вказівник на наступний елемент.
2. Обрати за номером варіанту файл (l5vXX.txt, де XX – номер варіанту) з каталогу вхідних даних для лабораторних робіт №5.
3. Прочитати з файлу групи по три реальних числа, вважати їх координатами *x,y,z* та створювати та додавати до списку нові елементи з такими координатами.
4. Надрукувати елементи списку та їх підраховану кількість.
5. Скласти функцію, яка обчислює скалярний добуток двох векторів, переданих вказівниками, та повертає реальне значення.
6. Скласти функцію, яка обчислює кут між двох векторів, переданих вказівниками, та повертає реальне значення.
7. Скласти функцію, яка обчислює суму двох векторів, переданих вказівниками та записує у третій вектор, який приймає через вказівник.
8. Скласти функцію, яка обчислює векторний добуток двох векторів, переданих вказівниками та записує у третій вектор, який приймає через вказівник.
9. Скласти функцію, яка обчислює добуток вектора, поданого вказівником, на скаляр та записує результат у інший вектор, поданий вказівником. Якщо другий вектор не заданий (NULL) – записати результат у перший вектор.
10. Останні три функції повинні повертати той вказівник на структуру, у який вони записують результат, або *NULL*, якщо операція не виконана.
11. Виконати зі списком операцію відповідно до номеру варіанту та надрукувати результат.

Варіанти:

№ варіанту	Задача
1	Знайти два сусідніх вектора, кут між якими найменьший
2	Знайти суму усіх векторів списку
3	Знайти векторний добуток першого та останнього елемента списку
4	Знайти скалярний добуток першого та останнього елемента списку
5	Знайти усі вектори, що є локальними максимумами за модулем. Надрукувати їх координати та номери.
6	Знайти найбільший та найменьший за модулем вектори та їх скалярний

	добуток.
7	Знайти найбільший за модулем вектор, надрукувати його номер та координати
8	Знайти два сусідніх вектора, кут між якими найбільший
9	Знайти найбільший та найменьший за модулем вектори та їх векторний добуток.
10	Знайти найменьший за модулем вектор, надрукувати його номер та координати
11	Знайти усі вектори, що є локальними мінімумами за модулем. Надрукувати їх координати та номери.
12	Знайти добуток суми усіх векторів та склярного добутку першого та останнього вектора
13	Знайти найменьший за модулем вектор, надрукувати його номер та координати
14	Знайти скалярний добуток першого та останнього елемента списку
15	Знайти два сусідніх вектора, кут між якими найменьший
16	Знайти усі вектори, що є локальними максимумами за модулем. Надрукувати їх координати та номери.
17	Знайти найбільший за модулем вектор, надрукувати його номер та координати
18	Знайти добуток суми усіх векторів та склярного добутку першого та останнього вектора
19	Знайти два сусідніх вектора, кут між якими найбільший
20	Знайти найбільший та найменьший за модулем вектори та їх векторний добуток.
21	Знайти усі вектори, що є локальними мінімумами за модулем. Надрукувати їх координати та номери.
22	Знайти векторний добуток першого та останнього елемента списку
23	Знайти суму усіх векторів списку
24	Знайти найбільший та найменьший за модулем вектори та їх скалярний добуток.
25	Знайти суму усіх векторів списку