Программирование на языках высокого уровня

Лаб. работа 13. Односвязные списки

Общие сведения

На оценку «3» - задачи сложности A и B.

На оценку «4» - задачи сложности C

На оценку «5» - задачи сложности D

Требуется написать программу, реализующую задание, соответствующее варианту с использованием односвязного списка

Задача А

Вводить с клавиатуры числовые элементы до тех пор, пока не будет введен 0. После этого вывести на экран весь список введенных элементов.

Задача B

Вводить с клавиатуры строки до тех пор, пока не будет введена строка «stop». После этого вывести на экран весь список введенных элементов.

Задача C

Вводить с клавиатуры элементы телефонной книги до тех пор, пока не будет введено слово «stop» вместо имени.

После завершения ввода программа выводит запрос на ввод номера телефона. После этого пользователь вводит номер телефона и программа ищет его в списке. Если находит – выводит соответствующее имя. Не находит – выводит «Not found»

Задача D

Реализовать действия над списком, аналогичные представленным в задаче C.

Программа должна при запуске выводить меню следующего содержания:

1. Показать справочник
2. Добавить запись
3. Найти по номеру
4. Удалить по номеру
5. Очистить справочник

Подсказка

- Элемент списка реализуется в виде структуры

```
struct Elem
{
    int Data; //Данные любого типа и в любом количестве
    Elem *next;
};
```

Подсказка

- Пустой список:

```
Elem *listbegin = NULL;
```

Добавление элемента в список:

```
Elem *newElem = (Elem*)malloc (sizeof(Elem));
newElem -> next = listbegin;
newElem -> data = …;
listbegin = newElem;
```

Подсказка

Проход по списку:

```
Elem *cur;

cur = listbegin;
while (cur != NULL)
{
    … cur -> data … //обработка данных текущего элемента
    cur = cur -> next;
}
```

Подсказка

Очистка списка

```
Elem *cur;

cur = listbegin;
while (cur != NULL)
{
    Elem *del = cur;
    cur = cur -> next;
    free(del);
}
```

Программирование на языках высокого уровня

Лаб. работа 14. Двусвязные списки

Общие сведения

На оценку «3» - задачи сложности A и B.

На оценку «4» - задачи сложности C

На оценку «5» - задачи сложности D

Требуется написать программу, реализующую задание, соответствующее варианту с использованием двусвязного списка

Задания по вариантам брать из ***Л/Р №12*** за исключением того, что вывод списка должен осуществляться сначала в прямом, затем – в обратном направлении.

Подсказка

- Элемент списка реализуется в виде структуры

```
struct Elem
{
    int Data; //Данные любого типа и в любом количестве
    Elem *next;
    Elem *prev;
};
```

Подсказка

- Пустой список:

Elem *head = NULL; Elem *tail = NULL; //Голова и хвост

Добавление элемента в начало списка:

```
Elem *newElem = (Elem*)malloc (sizeof(Elem));
newElem -> next = head;
newElem -> prev = NULL;
newElem -> data = …;
head = newElem;
```

Подсказка

Добавление элемента в конец списка:

```
Elem *newElem = (Elem*)malloc (sizeof(Elem));
newElem -> next = NULL;
newElem -> prev = tail;
newElem -> data = …;
tail = newElem;
```

Нужно обратить **внимание**, что если добавляется **первый элемент в пустой список**, то нужно и head, и tail присвоить новому элементу!