Федеральное государственное автономное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ТЕЛЕКОММУНИКАЦИИ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Доцент кафедры САИТ Студент группы КТбо1-2

Лапшин В. С. Мастерских А.С.

**ЛАБОРАТОРНАЯ РАБОТА №2**

По дисциплине

«Алгоритмизация и программирование»

Вариант 8

На тему

«Динамическое распределение памяти, списки»

**Таганрог 2024**

* **Техническое задание**
* **Цель задания**

Цель данной лабораторной работы является освоение способов предоставления и обработки данных в виде списков.

Необходимо изучить материалы по темам списки, указатели, способы выделения памяти, функции, отвечающие за выделение памяти, функции *malloc(), calloc()* и их отличия, функцию *free(),* структуры данных.

* **Задача**

Написать программу для работы с односвязным и двусвязным списками в соответствии с выданным вариантом задания (№8). Предусмотреть в программах следующие функции:

* Включить новый элемент в конец списка.
* Включить новый элемент на заданное пользователе место списка.
* Включить новый элемент после элемента с заданной информационной частью.
* Включить новый элемент перед элементом с заданной информационной частью.
* Включить новый элемент в середину списка.
* Исключить элемент из середины списка.
* Исключить элемент с заданной информационной частью.
* Исключить элемент из конца списка.
* Исключить элемент из заданного пользователем места списка.
* Исключить элемент из головы списка.

Задание к Варианту №8:

Программы должны уметь обрабатывать книгу рецептов.

* **Ход работы**

Функция append() добавляет новый элемент в конец списка, сначала список проверяется на пустоту, если в нем есть элементы, то мы последовательно проходим по всем, пока не достигнем конца списка. Последний элемент списка указывает на новый элемент. Если список пустой, то указатель на начало списка, ссылается на новый элемент.

Функция add\_by\_pos() включает новый элемент на заданное пользователем место списка. Она проходит последовательно по всем элементам и увеличивает счётчик с каждым новым элементом. Если цикл смог достичь нужной позиции, то элемент перед этой позицией начинает ссылаться на новый элемент, а новый элемент на следующий.

Функция add\_by\_gos\_number\_after() добавляет новый элемент после элемента с заданным гос. номером. Функция последовательно проходит по всем элементам и сравнивает гос. номер с искомым. Если случается совпадение, то найденный элемент ссылается на новый, а новый ссылается на следующий за найденным.

Функция add\_by\_gos\_number\_before() выключает новый элемент перед элементом с заданным гос. номером. Она тоже проходит по каждому элементу, только теперь проверяем не текущий элемент, а сразу следующий за ним, чтобы иметь указатель на элемент перед искомым.

Функция add\_in\_center() добавляет новый элемент в середину списка. В ней мы манипулируем двумя указателями, мы последовательно проходим по списку и пока первый указатель делает один шаг, второй делает два шага, таким образом, когда второй указатель дойдет до конца списка, первый будет указывать на его середину. Новый элемент добавляется после найденного.

Функция delete\_from\_center() удаляет элемент из середины списка, она работает аналогично предыдущей функции, но добавляется еще один указатель, который ссылается на элемент предшествующий центральному. При достижении середины предыдущий элемент начинает ссылаться на следующий за центральным, а центральный удаляется.

Функция delete\_by\_gos\_number() удаляет элемент с заданным гос. номером, она последовательно проходит по списку и при совпадении гос. номера удаляет найденный объект.

Функция delete\_last() удаляет последний элемент списка. Она последовательно проходит по всему списку до конца, у предпоследнего элемента обнуляет указатель next() и удаляет последний элемент.

Функция delete\_by\_pos() проходит по списку, наращивая счетчик с каждым пройденным элементом, при достижении позиции перед искомой она удаляет следующий элемент и переопределяет указатель.

Функция delete\_first() удаляет элемент из начала списка. Она переносит указатель на начало списка на следующий элемент и удаляет первый.

В тесте мы создаем пять объектов и пустой список, потом последовательно вызываем каждую созданную нами функцию, выводя результат их выполнения.

**Алгоритм работы с двусвязными списками**

Реализация функций для двусвязного списка аналогична реализации для односвязного за исключением того, что теперь помимо контроля указателя на следующий элемент, необходимо контролировать указатели на предыдущий элемент, помимо этого сам список состоит из указателя на его начало, но и из указателя на его конец. Возможность проходить список в обратном направлении с одной стороны облегчает реализацию операций, но с другой стороны усложняет за счет необходимости контроля большего количества указателей.

* **Вывод**

В ходе данной работы я ознакомилась с такими понятиями как списки, указатели, функции *malloc()* и *calloc(),* структуры данных, функцию *free().* Пользуясь этими понятиями, написала программу по заданию к своему варианту. Научилась работать с односвязными и двусвязными списками.