

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

Отчет по лабораторной работе № 12
по дисциплине «Программирование»
Тема: «Кольцевые списки».

Студент гр. 9305

Николаенко К.Н.

Преподаватель

Перязева Ю. В.

Санкт-Петербург

2019

Содержание

Введение	2
Задание	2
Описание структур	3
Схема вызова функций	5
Функций	7
Заключение	25

Введение

Получить практические навыки в разработке алгоритма и написании программы на языке Си. Для ознакомления работы с кольцевыми списками, а также правилами их написания на языке Си.

Задание

Разработать подалгоритм удаляющий в односвязном кольцевом списке заданный по номеру элемент. Номер элемента задается с «конца» списка. При недостаточном количестве элементов в списке удалить элемент из начала списка. При пустом списке вывести соответствующее сообщение.

Описание структур

Описание структуры данных

Имя поля	Тип	Назначение
name	char	Название футбольного клуба
country	char	Название страны, в котором находится данный клуб
probability	float	вероятность выхода в Лигу Чемпионов
statistics	float	массив из 2 элементов(1 - количество побед, 2 - количество ничьих)

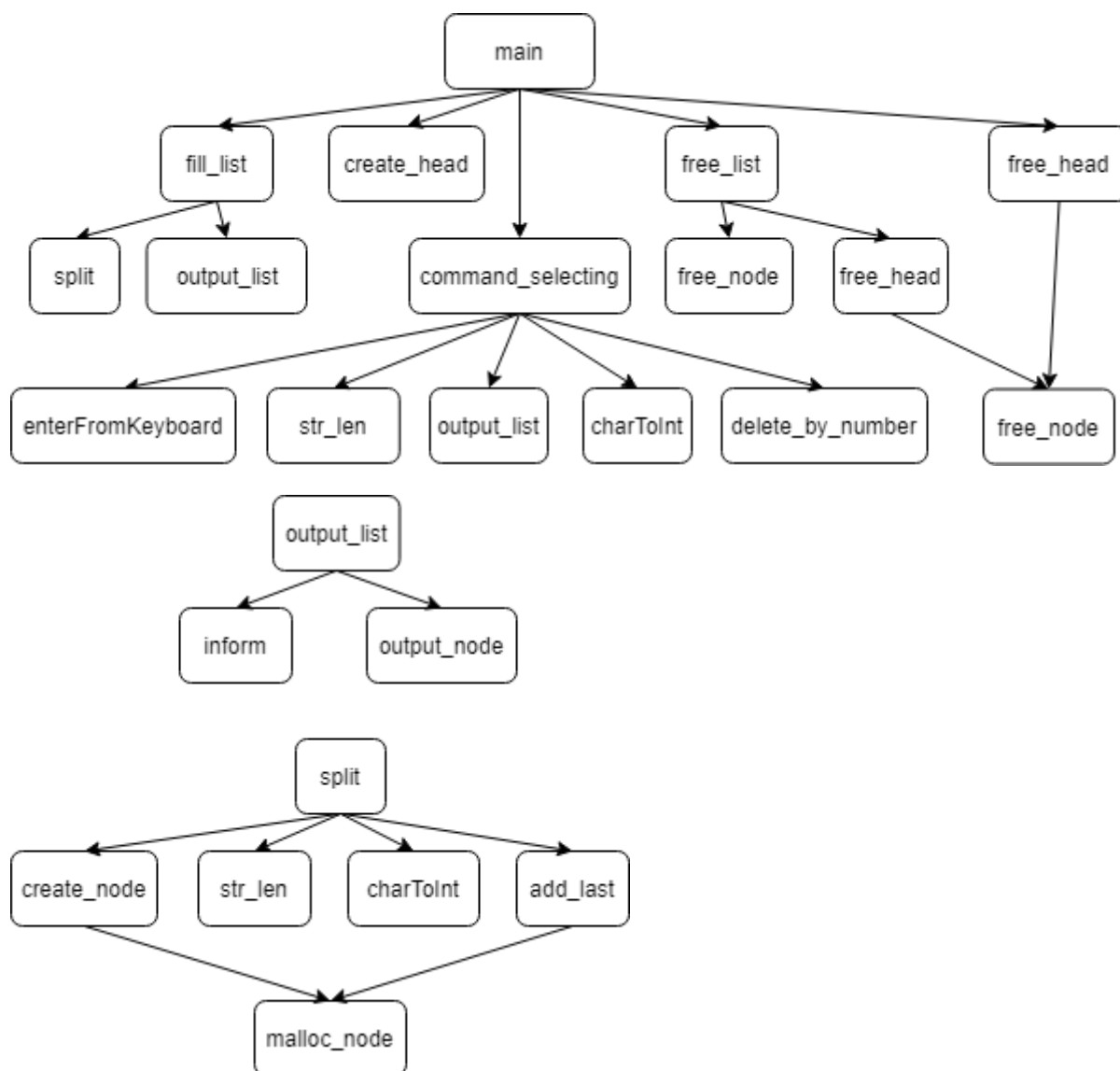
Описание структуры узла

Имя поля	Тип	Назначение
id	size_t	номер узла
baza	int	адрес на структуру данных
next	node	адрес следующего узла

Описание структуры головы

Имя поля	Тип	Назначение
N	size_t	количество узлов
first	node	адрес на первый узел

Схемы вызова функций



delete_by_number



delete_node

enterFromKeyboard



add_last

create_node



malloc_node

Функции

1. main

Описание:

Вызов функций считывания из файла, работы с программой и очищения.

Прототип:

int main()

Примеры вызова:

main()

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная переменная	q	head	голова списка

Возвращаемое значение: 0

2.fill_list

Описание:

Считывание информации из файла и забивания его в массив структуры. Пока строка не совпадет с предыдущей, программа ее разделяет и записывает в поля списка.

Прототип:

```
void fill_list(head *q)
```

Примеры вызова:

```
fill_list(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	message	char	одна из строк файла
Локальная	str	char	одна из строк файла
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

3.enterFromKeyboard

Описание:

Ввод информации в базу данных с клавиатуры: вначало списка или в конец.

Прототип:

```
void enterFromKeyboard(head *q)
```

Примеры вызова

```
enterFromKeyboard(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная переменная	temp	node	узел
Локальная переменная	k	integer	команда
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

4.output_list

Описание:

Печать базы данных с первой записи до последней.

Прототип:

```
void output_list(head *q)
```

Пример вызова:

```
output_list(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	q	head	голова списка
Формальный аргумент	temp	node	узел

Возвращаемое значение: отсутствует

5.inform

Описание:

информация о столбцах базы данных.

Прототип:

```
void inform()
```

Пример вызова:

```
inform()
```

Описание переменных: отсутствуют.

Возвращаемое значение: отсутствует.

6.command_selecting

Описание:

работа с пользователем, тоесть программа не завершается пока пользователь не захочет этого (введет нужную команду), тоесть для доабвления элемента - 1, для удаления - 2, для вывода списка - 3, выхода из программы 4. Тоесть работает по не введено 4.

Прототип:

```
void command_selecting(head *q)
```

Пример вызова:

```
command_selecting(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	q	head	голова списка
Локальная переменная	s	char	команда
Локальная переменная	f	integer	индификатор команды

Возвращаемое значение: отсутствует

7.output_node

Описание:

ВЫВОД узла.

Прототип:

```
void *output_node(node *temp)
```

Пример вызова:

```
output_node(temp)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	temp	node	узел

Возвращаемое значение: отсутствует

8.str_len

Описание:
длина строки.

Прототип:
int str_len(char *s)

Пример вызова:
str_len(s)

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	s	char	строка
Локальная переменная	r	integer	количество символов

Возвращаемое значение: r

9.charToInt

Описание:

из char в int, номер символа в инт минус 48.

Прототип:

```
int charToInt(char numeric)
```

Пример вызова:

```
charToInt(numeric)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	numeric	char	СИМВОЛ

Возвращаемое значение: цифру в int

10.split

Описание:

выделение нужной информации из строки и запись в базу данных. То есть функция разделяет полученную строку на 4 части и записывает в список.

Прототип:

```
void split(char *mes, person *mass, int r, int q, int n)
```

Пример вызова:

```
split(s, futClub, r, q, n)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	q	head	голова списка
Локальная переменная	temp	node	узел
Формальный аргумент	mes	char	строка из файла
Формальный аргумент	n	integer	вспомогательная переменная для выделения информации
Локальная переменная	s	char	выделяемая подстрока из строки файла
Локальная переменная	k	integer	длина выделяемой подстроки

Возвращаемое значение: отсутствует

11. free_head

Описание:

очищение головы списка.

Прототип:

```
void free_head(head *q)
```

Примеры вызова:

```
free_head(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

12. free_node

Описание:

очищение узла.

Прототип:

```
void free_node(node *temp)
```

Примеры вызова:

```
free_node(temp)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

13. free_list

Описание:

очищение всего списка, пока у последнего узла не будет ссылка на NULL.

Прототип:

```
void free_list(head *q)
```

Примеры вызова:

```
free_list(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	p	node	предыдущий узел
Локальная	temp	node	узел
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

14. malloc_node

Описание:

выделяет память полям узла.

Прототип:

```
void malloc_node(node *temp)
```

Примеры вызова:

```
malloc_node(temp)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Формальный аргумент	temp	node	узел

Возвращаемое значение: отсутствует

15. create_head

Описание:

создание головы списка.

Прототип:

head *create_head()

Примеры вызова:

create_head()

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	p	head	голова списка

Возвращаемое значение: p

16. create_node

Описание:

создание узла.

Прототип:

```
node *create_node(head *q)
```

Примеры вызова:

```
create_node(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	temp	node	узел
Формальный аргумент	q	head	голова списка

Возвращаемое значение: temp

17. add_last

Описание:

добавление узла в конец списка.

Прототип:

```
void add_last(head *q)
```

Примеры вызова:

```
add_last(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	temp	node	узел
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

18. delete_node

Описание:

удаление любого узла, кроме первого.

Прототип:

```
void delete_node(node *p, head *q)
```

Примеры вызова:

```
delete_node(p, q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	temp	node	узел
Формальный аргумент	q	head	голова списка
формальный аргумент	p	node	предыдущий узел

Возвращаемое значение: отсутствует

19. delete_by_number

Описание:

поиск узла, который нужно удалить.

Прототип:

```
void delete_by_number(head *q)
```

Примеры вызова:

```
delete_by_number(q)
```

Описание переменных:

Вид переменной	Имя переменной	Тип	Назначение
Локальная	p	node	узел
Формальный аргумент	q	head	голова списка
Локальный	k	integer	номер удаляемого элемента
Локальный	i	integer	счетчик

Возвращаемое значение: отсутствует

Заключение

Выводы:

При выполнении лабораторной работы были получены практические навыки в разработке алгоритма и написании программы на языке Си, а также получена информация о кольцевых списках.