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

Кафедра вычислительной техники

Отчет по лабораторной работе № 11

по дисциплине «Программирование»

Тема: «Линейные двусвязные списки».

Студент гр. 9305 Николаенко К.Н.

Преподаватель Перязева Ю. В.

Санкт-Петербург

2019

Содержание

Введение

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

Задание

Разработать подалгоритм удаления элементов с заданным содержимым указанного информационного поля из односвязного списка. При отсутствии таких элементов в 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 - количетсво побед, 2 - количетсво ничьих)

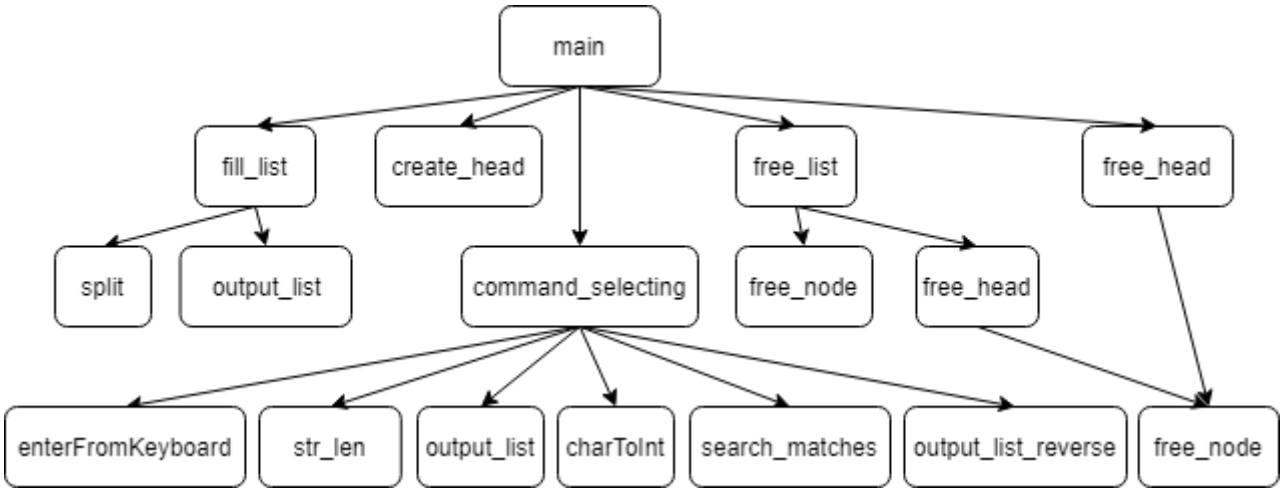
Описание структуры узла

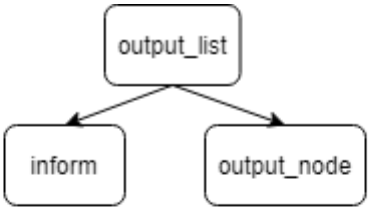
Имя поля	**Тип**	**Назначение**
id	size_t	номер узла
baza	fut	адрес на структуру данных
next	node	адрес следующего узла
prev	node	адрес предыдущего узла

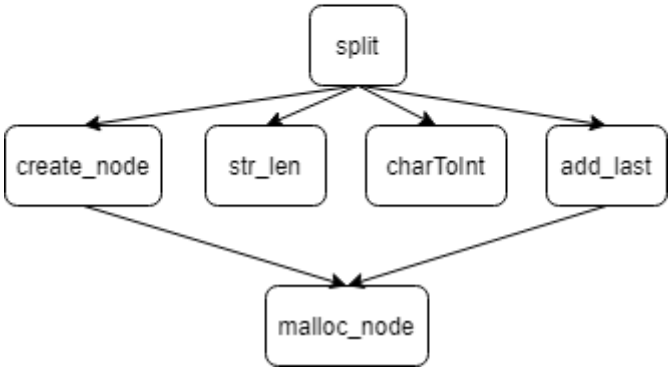
Описание структуры головы

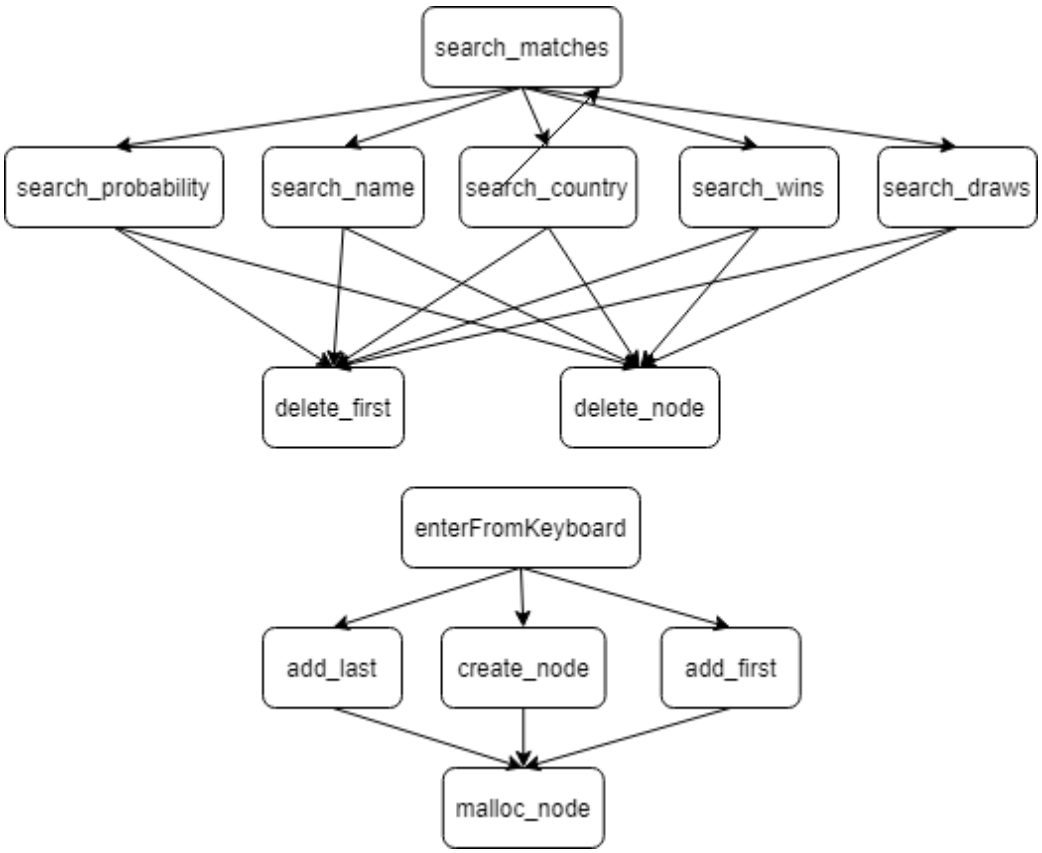
Имя поля	**Тип**	**Назначение**
N	size_t	количество узлов
first	node	адрес на первый узел
last	node	адрес на последний узел

Схемы вызова функций








search_matches
search_probability
search_name
search_country
search_wins
search_draws
delete_first
delete_node
enterFromKeyboard
add_last
create_node
add_first
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. Тоесть функция разделяет получаенную 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. search_matches

Описание:
поиск совпадений по определенному полю для удаления. По 1, 2 вводятся слова и по ним осуществляется поиск совпадений и вследсвие чего удалений, в 4, 5 вводится число от 0 до 10, происходит поиск и удалений, если есть, в 3 вводится область, тоесть от чего-то до чего-то, а так как это вероятность, то границы поиска от 0 до 1 максимум.

Прототип:
void search_matches(head *q)

Примеры вызова:
search_matches(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	k	int	команда
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

15. search_draws

Описание:
поиск по элементу поля массива statistics

Прототип:
void search_draws(head *q)

Примеры вызова:
search_draws(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	num	integer	количество совпадений
Локальная	temp	node	узел
Локальная	k	integer	значение поиска
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

16. search_wins

Описание:
поиск по элементу поля массива statistics

Прототип:
void search_wins(head *q)

Примеры вызова:
search_wins(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	num	integer	количество совпадений
Локальная	temp	node	узел
Локальная	k	integer	значение поиска
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

17. search_probability

Описание:
поиск по полю probabality

Прототип:
void search_probability(head *q)

Примеры вызова:
search_probability(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	num	integer	количество совпадений
Локальная	temp	node	узел
Локальная	beg	integer	граница поиска
Формальный аргумент	q	head	голова списка
Локальная	en	integer	граница поиска

Возвращаемое значение: отсутствует

18. search_country

Описание:
поиск по полю country

Прототип:
void search_country(head *q)

Примеры вызова:
search_country(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	num	integer	количество совпадений
Локальная	temp	node	узел
Локальная	s	char	строка поиска
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

19. search_name

Описание:
поиск по полю name

Прототип:
void search_name(head *q)

Примеры вызова:
search_name(q)

Описание переменных:

Вид переменной	**Имя переменной**	**Тип**	**Назначение**
Локальная	num	integer	количество совпадений
Локальная	temp	node	узел
Локальная	s	char	строка поиска
Формальный аргумент	q	head	голова списка

Возвращаемое значение: отсутствует

20. 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

21. 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

22. 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

23. 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

24. add_first

Описание:
добавление узла в начало списка

Прототип:
node *add_first(head *q)

Примеры вызова:
add_first(q)

Описание переменных:

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

Возвращаемое значение: temp

25. delete_first

Описание:
удаление первого узла

Прототип:
void delete_first(head *q)

Примеры вызова:
delete_first(q)

Описание переменных:

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

Возвращаемое значение: отсутствует

26. 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

27.output_list_reverse

Описание:
Печать базы данных от последней записи к первой.

Прототип:
void output_list_reverse(head *q)

Пример вызова:
output_list_reverse(q)

Описание переменных:

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

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 линейных двусвязных списках.