**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе № 8**

**по дисциплине «Программирование»**

**Тема: «Линейные односвязные списки»**

| Студент гр. 3311 |  | Сапронов К.Д. |
| --- | --- | --- |
| Преподаватель |  | Хахаев И.А. |

Санкт-Петербург

2024

**Цель работы.**

Цель данной работы состоит в изучении линейных односвязных списков на языке программирования С.

**Задание (вариант 8).**

Для выбранной предметной области создать динамический массив структур, содержащих характеристики объектов предметной области.

Обязательный набор полей:

* динамический массив символов, включая пробелы (name)
* произвольный динамический массив символов
* числовые поля типов int и float (не менее двух полей каждого типа)
* поле с числовым массивом.

Написать программу, обеспечивающую начальное формирование массива структур при чтении из файла (текст с разделителями — CSV) с последующим возможным дополнением элементов массива при вводе с клавиатуры.

Разработать подалгоритм добавления элемента в односвязный список после первого с «головы» элемента с заданным значением символьного поля и в конец списка, если такого элемента нет

**Постановка задачи и описание решения**

Для решения этой задачи можно воспользоваться следующим алгоритмом:

1. Создается список, заполняемый структурами из файла (предметная область - характеристики российских городов - название, регион, код региона, широта, долгота, площадь и население). При заполнении также заполняется список всех имен городов.
2. После ввода имени города оно в цикле сравнивается с каждым именем из списка имен. При каждой итерации цикла увеличивается счетчик id. При нахождении совпадения цикл завершается. Это сделано для того, чтобы при отсутствии совпадений id был равен количеству элементов в списке.
3. Создается новый элемент, который заполняется данными, введёнными с клавиатуры. Если id равен размеру списка, то новый элемент становится последним, иначе он встает после элемента, который был id по счету с головы.
4. Обновляется список имен городов.

**Описание переменных.**

**main()**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | ph | head | Указатель на заголовок списка |
| 2 | current | CITY\_node | Указатель на текущий узел |
| 3 | file | FILE | Файл для чтения |
| 4 | id | int | Номер элемента, после которого произойдет вставка |
| 5 | code | int | Код региона нового города |
| 6 | stat1 | int | Площадь нового города |
| 7 | stat2 | int | Население нового города |
| 8 | i | int | Переменная в циклах |
| 9 | answer | int | Ответ пользователя |
| 10 | latitude | float | Широта нового города |
| 11 | longitude | float | Долгота нового города |
| 12 | name | char | Имя нового города |
| 13 | region | char | Счетчик строк в цикле |
| 14 | name\_base | char | Список имён городов |

**make\_head()**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | ph | head | Указатель на заголовок списка |

**append()**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | new\_CITY | CITY\_node | Указатель на новый узел |
| 2 | list | head | Указатель на заголовок списка |
| 3 | name | char | Имя нового города |
| 4 | region | char | Счетчик строк в цикле |
| 5 | code | int | Код региона нового города |
| 6 | latitude | float | Широта нового города |
| 7 | longitude | float | Долгота нового города |
| 8 | stat1 | int | Площадь нового города |
| 9 | stat2 | int | Население нового города |

**insert()**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | new\_CITY | CITY\_node | Указатель на новый узел |
| 2 | target | CITY\_node | Указатель на узел, после которого производится вставка |
| 3 | list | head | Указатель на заголовок списка |
| 4 | i | int | Переменная внутри цикла |
| 5 | name | char | Имя нового города |
| 6 | region | char | Счетчик строк в цикле |
| 7 | code | int | Код региона нового города |
| 8 | latitude | float | Широта нового города |
| 9 | longitude | float | Долгота нового города |
| 10 | stat1 | int | Площадь нового города |
| 11 | stat2 | int | Население нового города |

**print\_list()**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | list | head | Указатель на заголовок списка |
| 2 | current | CITY\_node | Указатель на текущий узел |

**struct CITY\_node**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | name | char | Название |
| 2 | region | char | Регион |
| 3 | code | int | Код региона |
| 4 | latitude | float | Долгота |
| 5 | longitude | float | Широта |
| 6 | stats | int | Массив из двух характеристик: площадь и население |
| 7 | next | CITY\_node\* | Указатель на следующий узел списка |

**struct head**

| **№** | **Имя переменной** | **Тип** | **Назначение** |
| --- | --- | --- | --- |
| 1 | cnt | int | Количество элементов в списке |
| 2 | first | CITY\_node\* | Указатель на первый элемент списка |
| 3 | last | CITY\_node\* | Указатель на последний элемент списка |

**Контрольные примеры.**

**Пример 1.**

Ввод: 0

Kazan

Rostov-on-Don; South; 61; 47.23; 39.70; 354; 1135968;

0

Rostov-on-Don

Arkhangelsk; Northwest; 29; 64.54; 40.55; 294; 303357;

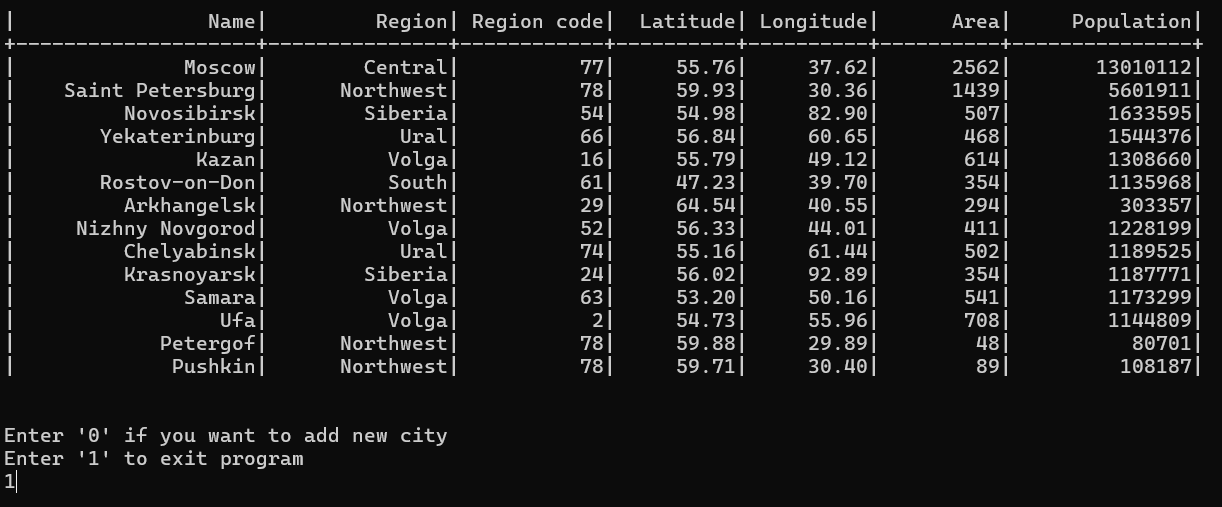
0

Bryansk

Pushkin; Northwest; 78; 59.71; 30.40; 89; 108187;

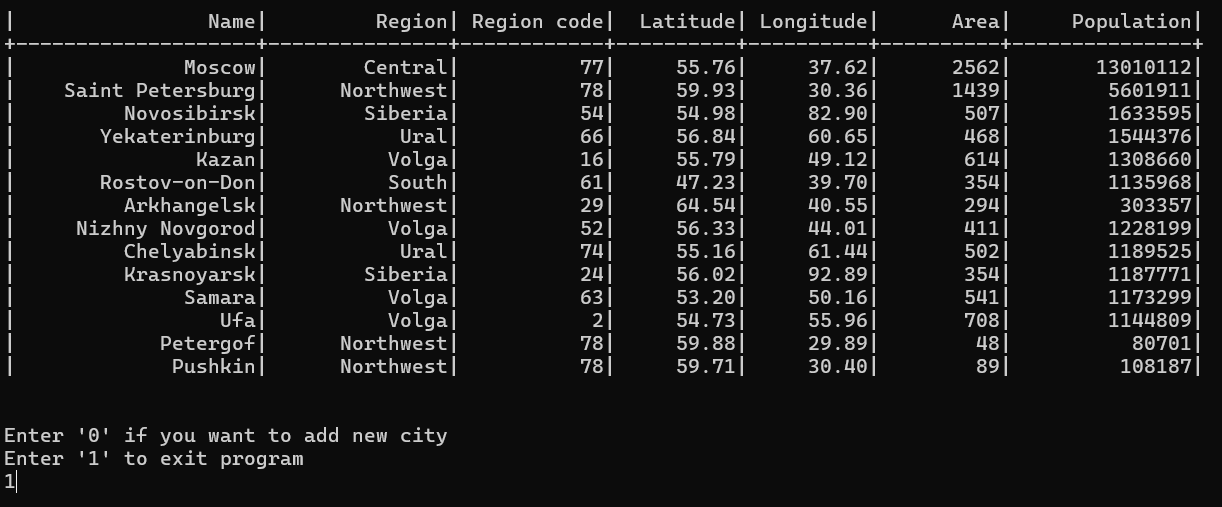
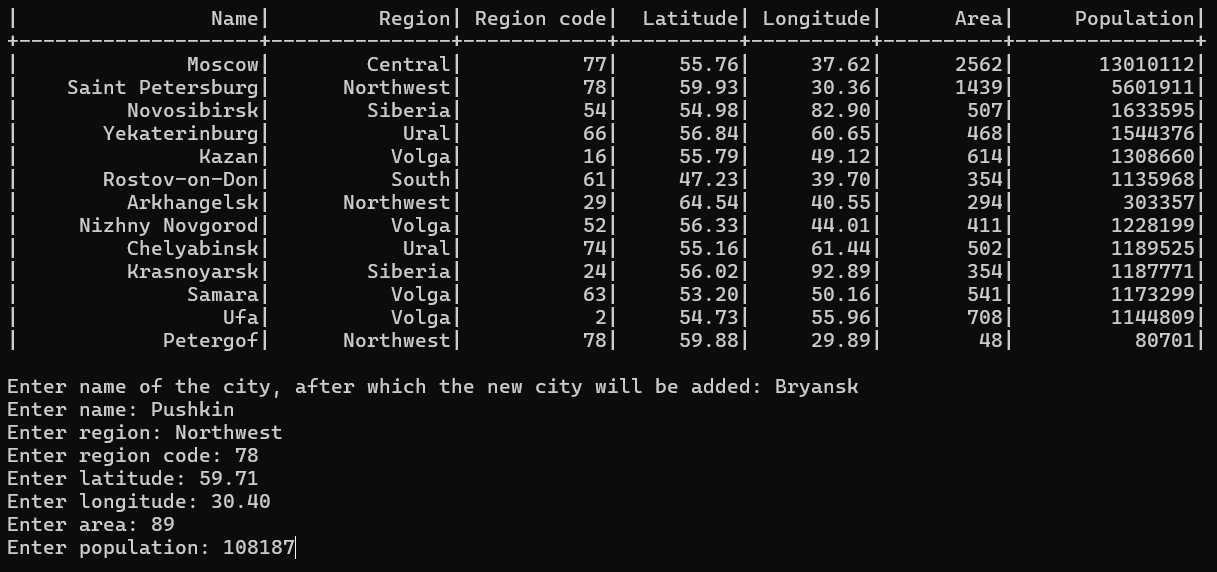
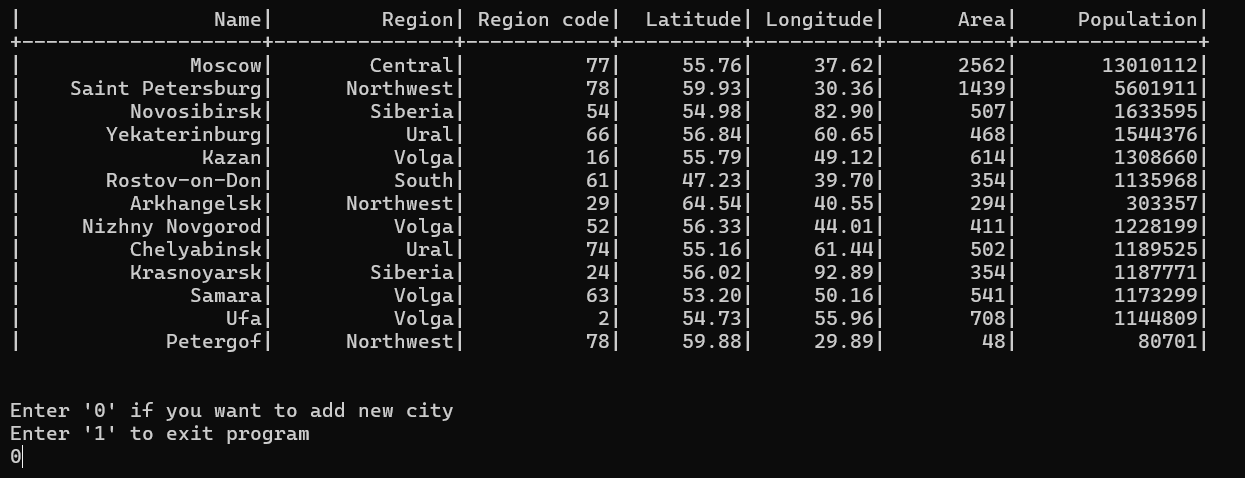
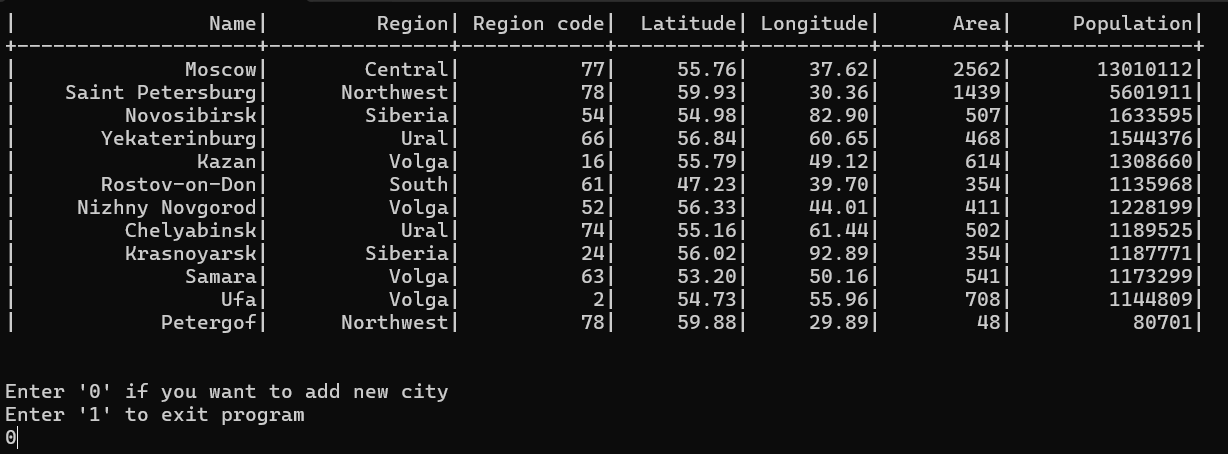
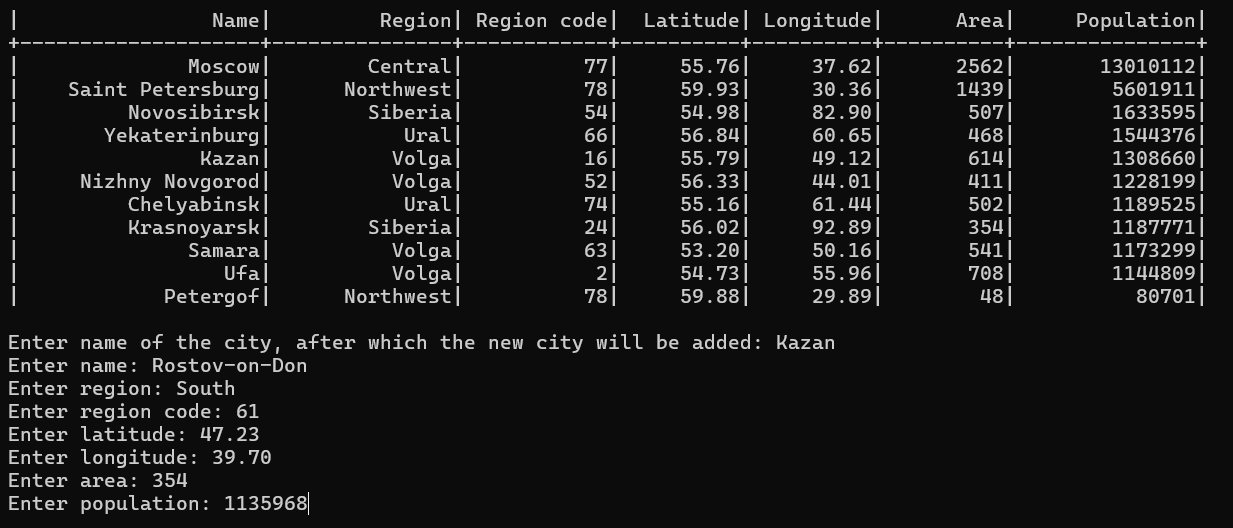
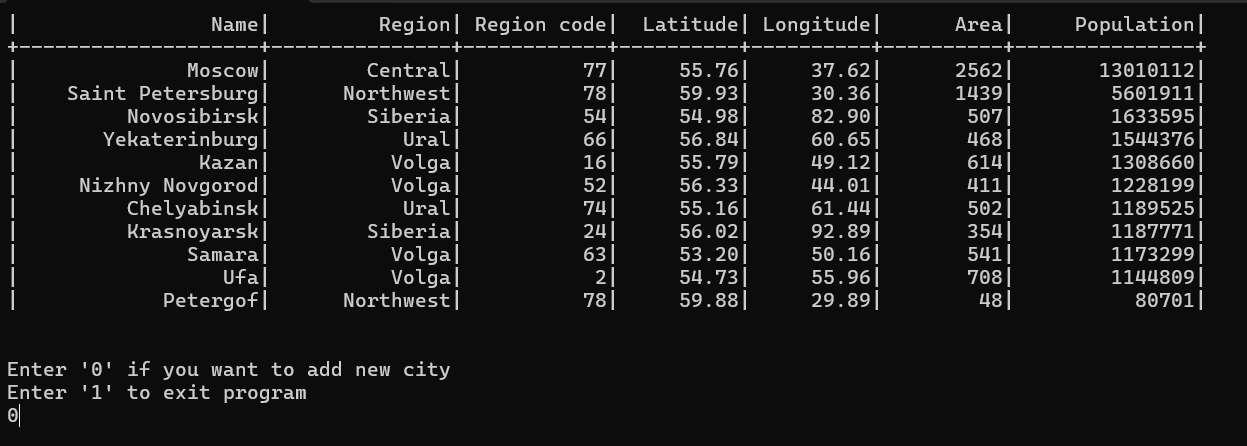
1

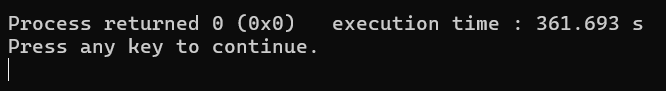
Вывод:



**Примеры выполнения программы.**

**Пример 1.**

****

****

**Выводы.**

В результате выполнения работы изучены методы работы с односвязными списками; получены практические навыки при программировании на языке С.