**Министерство высшего образования и науки Российской Федерации**

**ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)**

**Электротехнический факультет**

**Кафедра Информационные технологии и автоматизированные системы**

**Отчёт по теме**

**«Поиск данных с помощью хеш-таблиц»**

**Выполнил:**

**Студент 1 курса**

**Группы ИВТ-22-2б**

**Корючкин Савелий**

**Научный руководитель:**

**Доцент кафедры ИТАС**

**Полякова Ольга Андреевна**

**Пермь 2022**

**Постановка задачи**

Необходимо организовать поиск с помощью хеш-таблицы.

**Алгоритм решения**

Хеш-таблица - структура данных, где каждый элемент хранится в виде пары ключ-значение. Ключ - это уникальный номер.

*Устройство хеш-таблицы:*

1. Функция, которая возвращает некоторое неотрицательное [по индексу] значение, позволяющее осуществлять движение по хеш-таблице
2. Массив для хранения данных в хеш-таблице

*Свойства хорошей хеш-функции*

1. Должны быть только данные, прошедшие через хеш-функцию
2. Используются все данные, которые были хешированны
3. Детерминированный алгоритм - одни и те же данные вернут то же значение
4. Максимально разный хеш для мало отличающихся значений
5. Равномерное распределение данных (уменьшение коллизий)

*Методы разрешения коллизий*

1. Линейное пробирование (метод открытой адресации): коллизия занимает индекс hash+1, если он занят, то hash+2 и так далее.

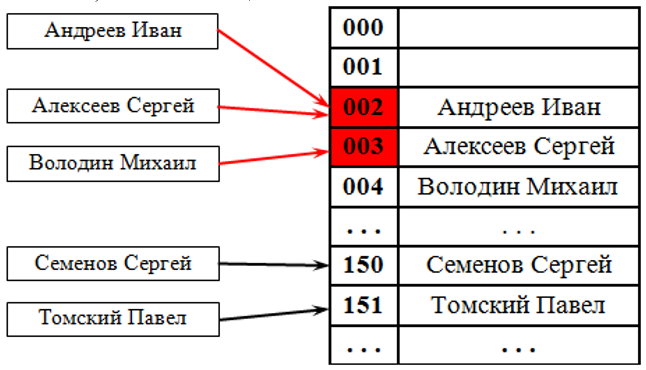


Рис. 1 - Принцип работы метода открытой адресации

1. Метод цепочек. Каждый элемент является указателем на начало связанного списка, и при коллизии элемент добавляют в начало списка.

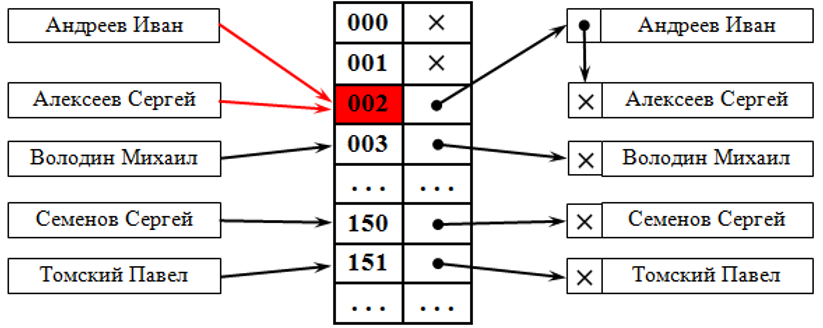


Рис. 2 - Принцип работы метода цепочек

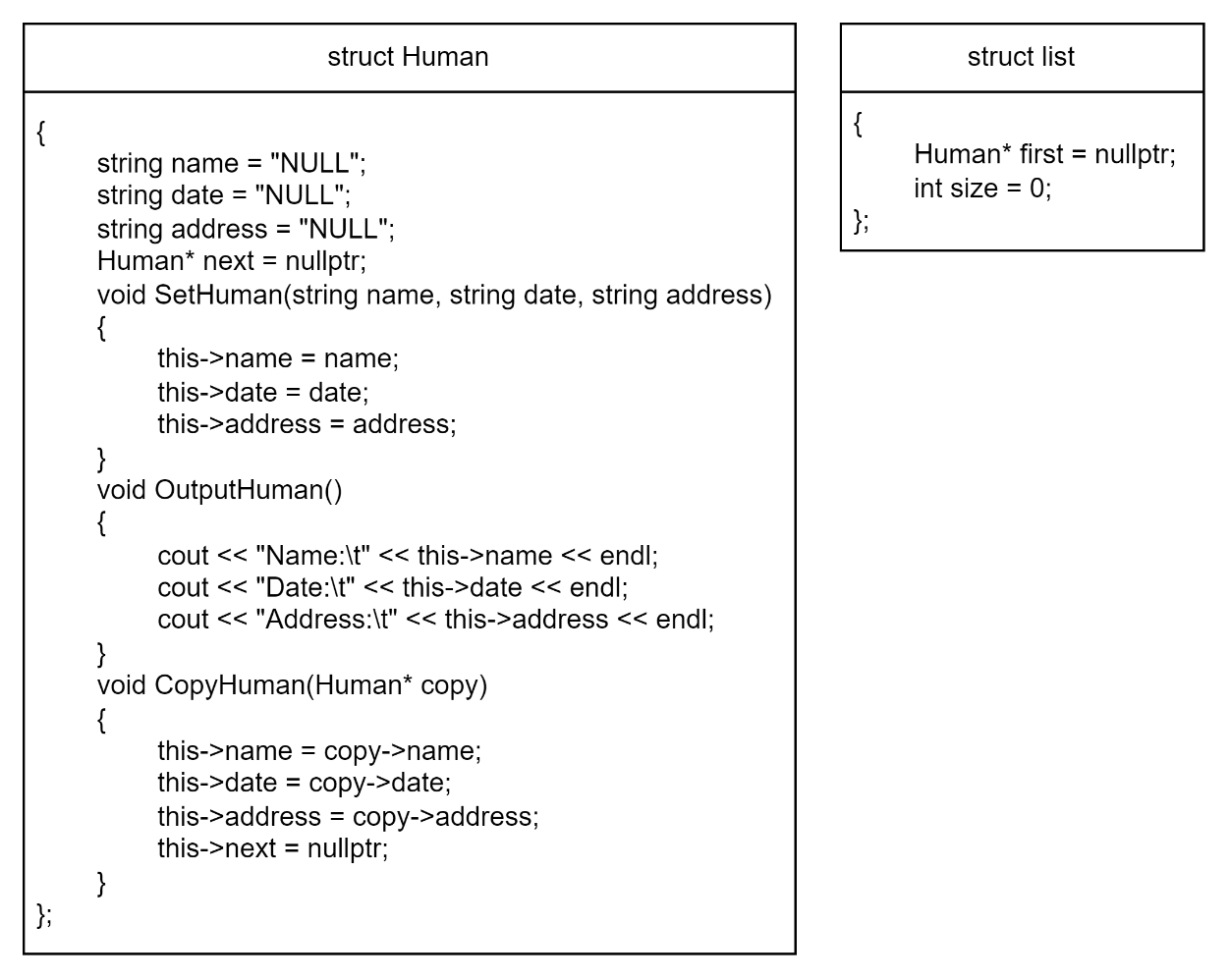


Рис. 3 – Структура struct Human и struct list()

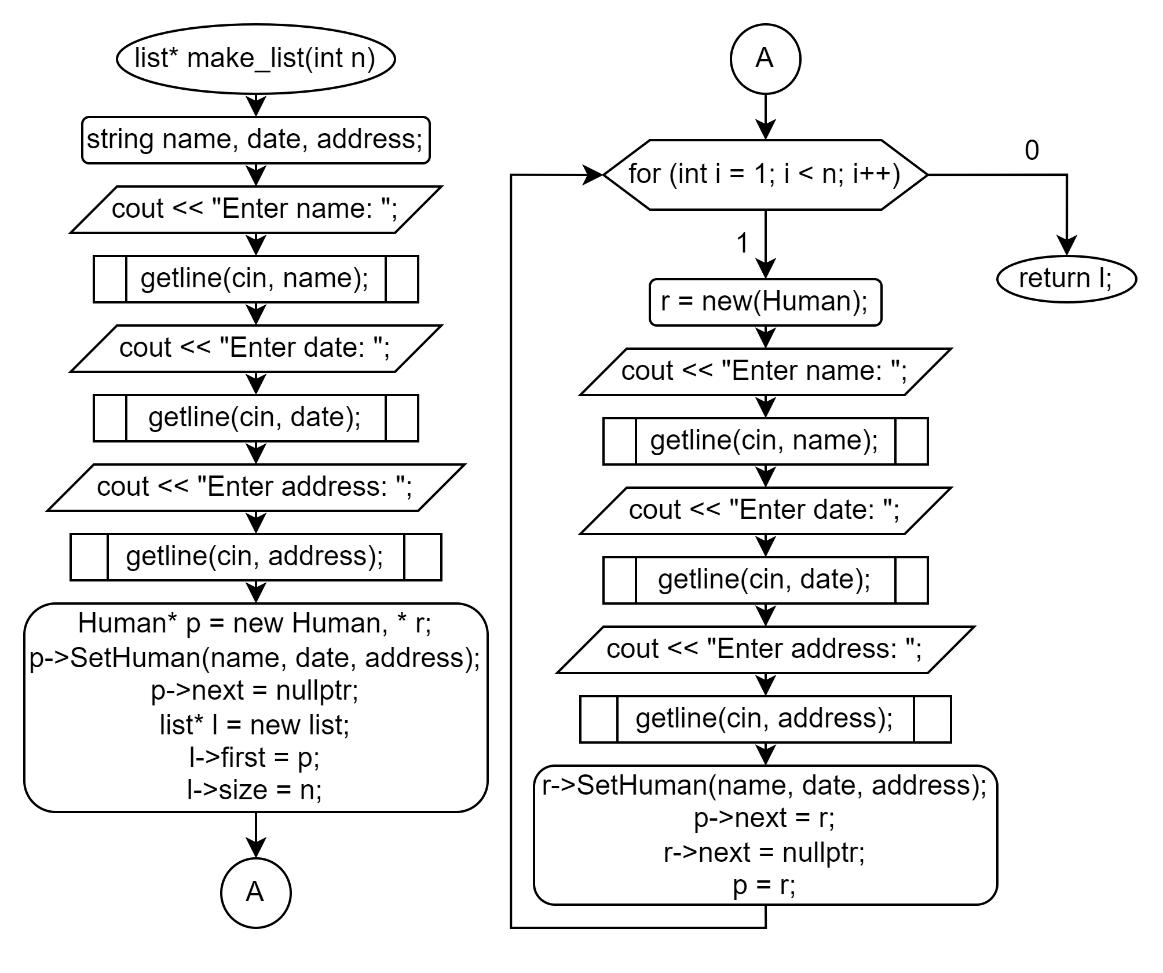


Рис. 4 – Функция list\* make\_list()

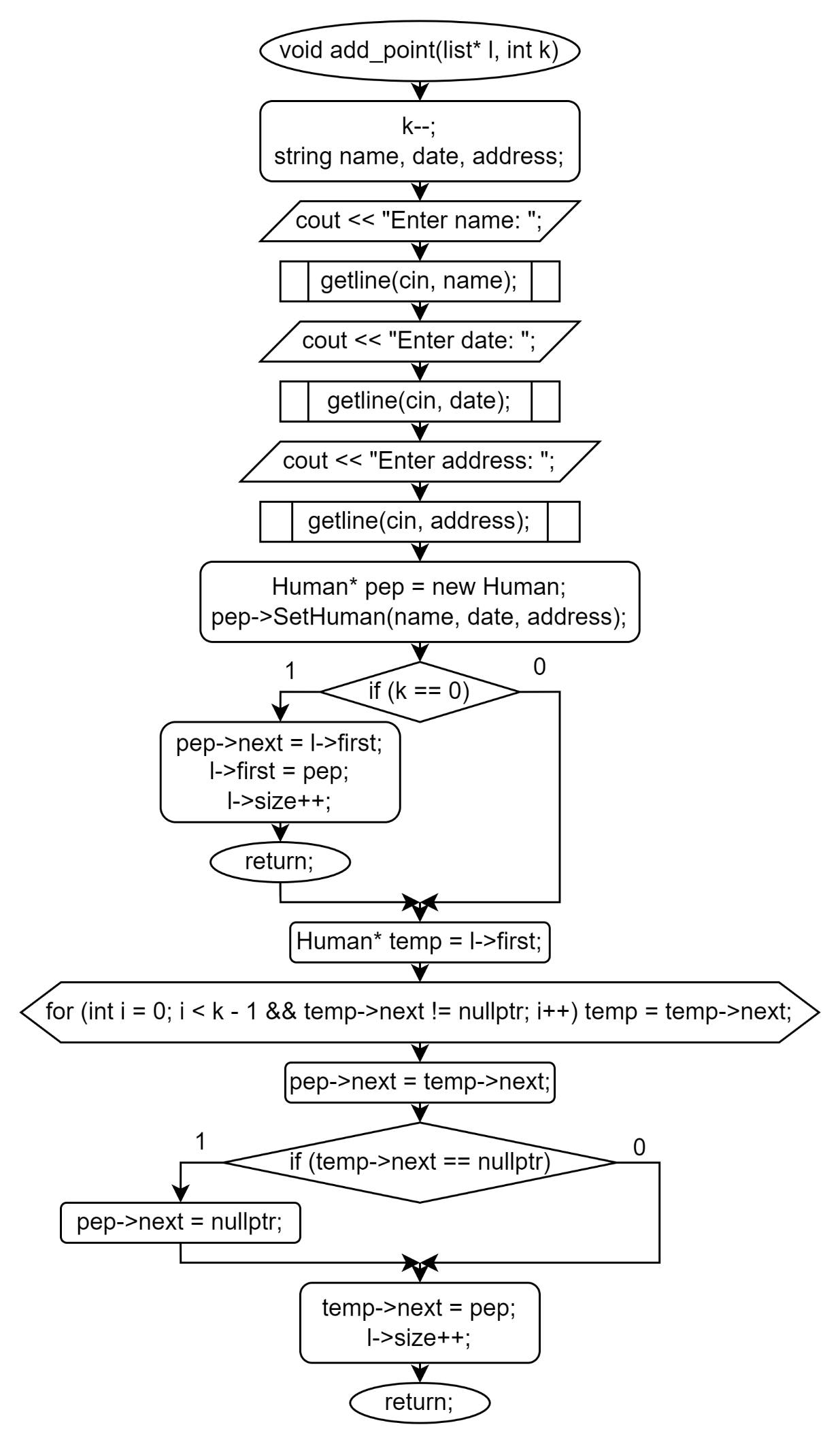


Рис. 5 – Функция void add\_point()



Рис. 6 – Функция void del\_point()

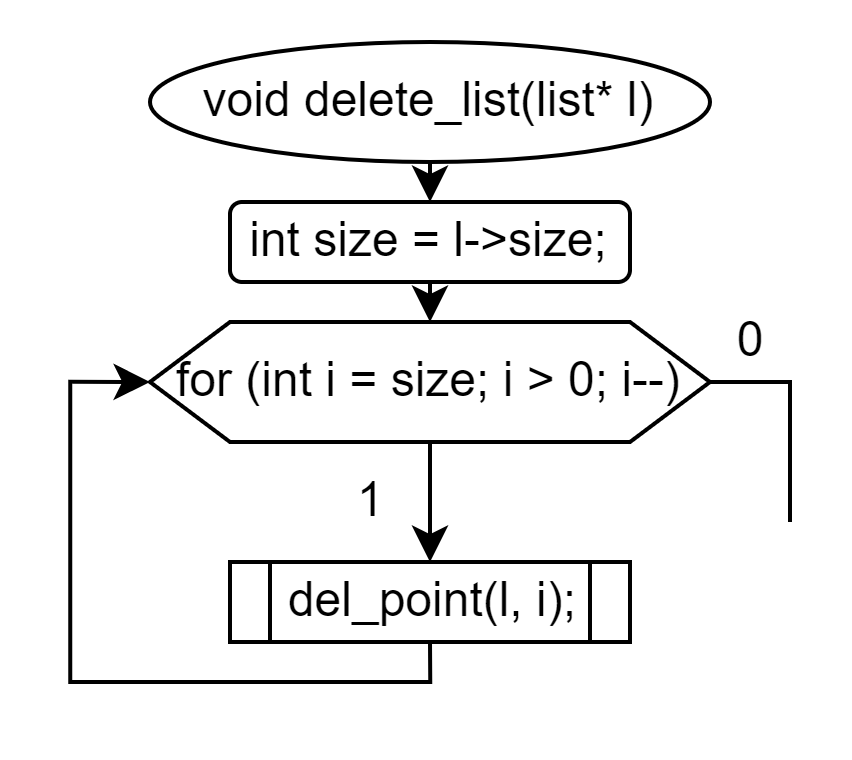


Рис. 7 – Функция void delete\_list()

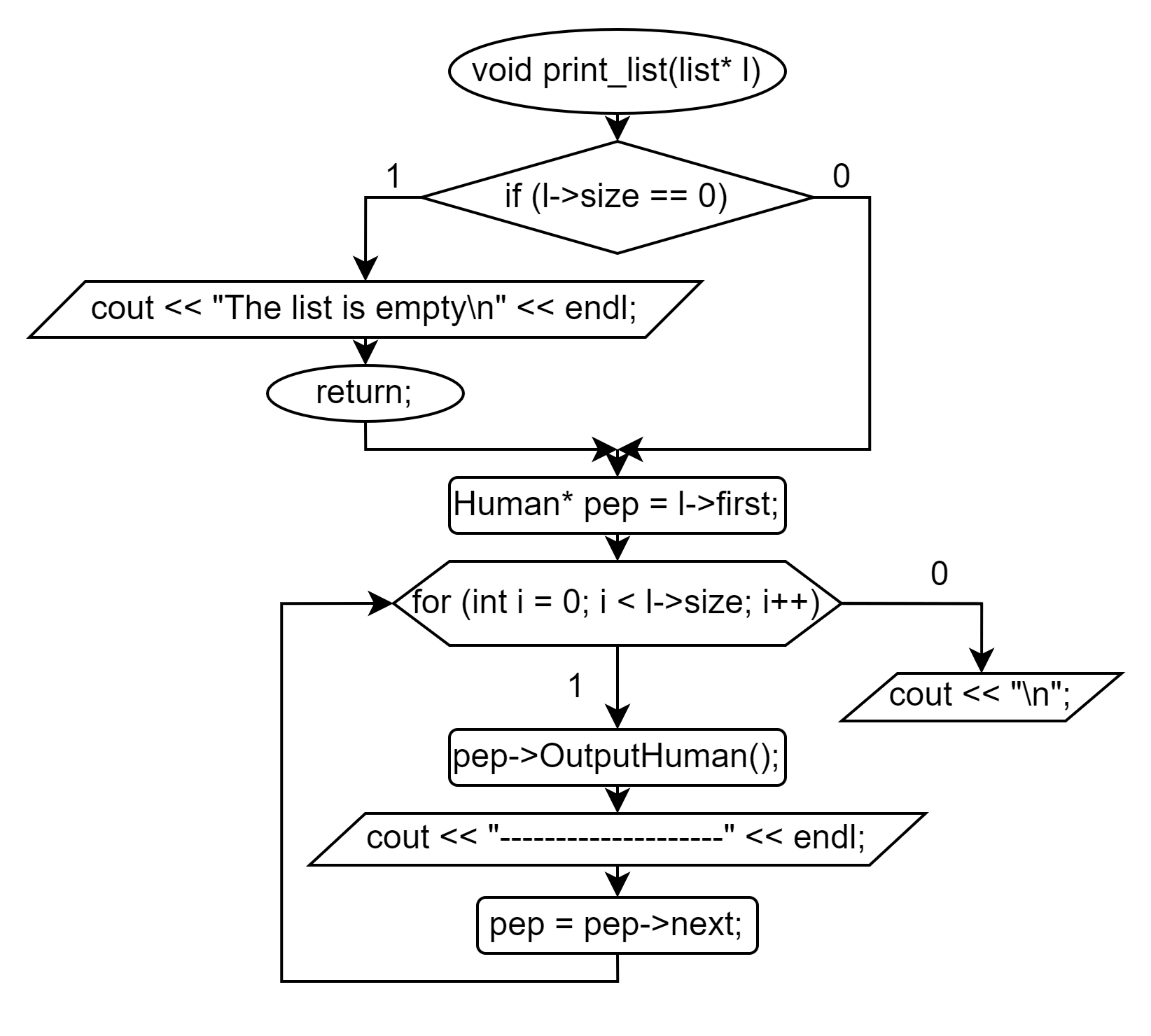


Рис. 8 – Функция void print\_list()

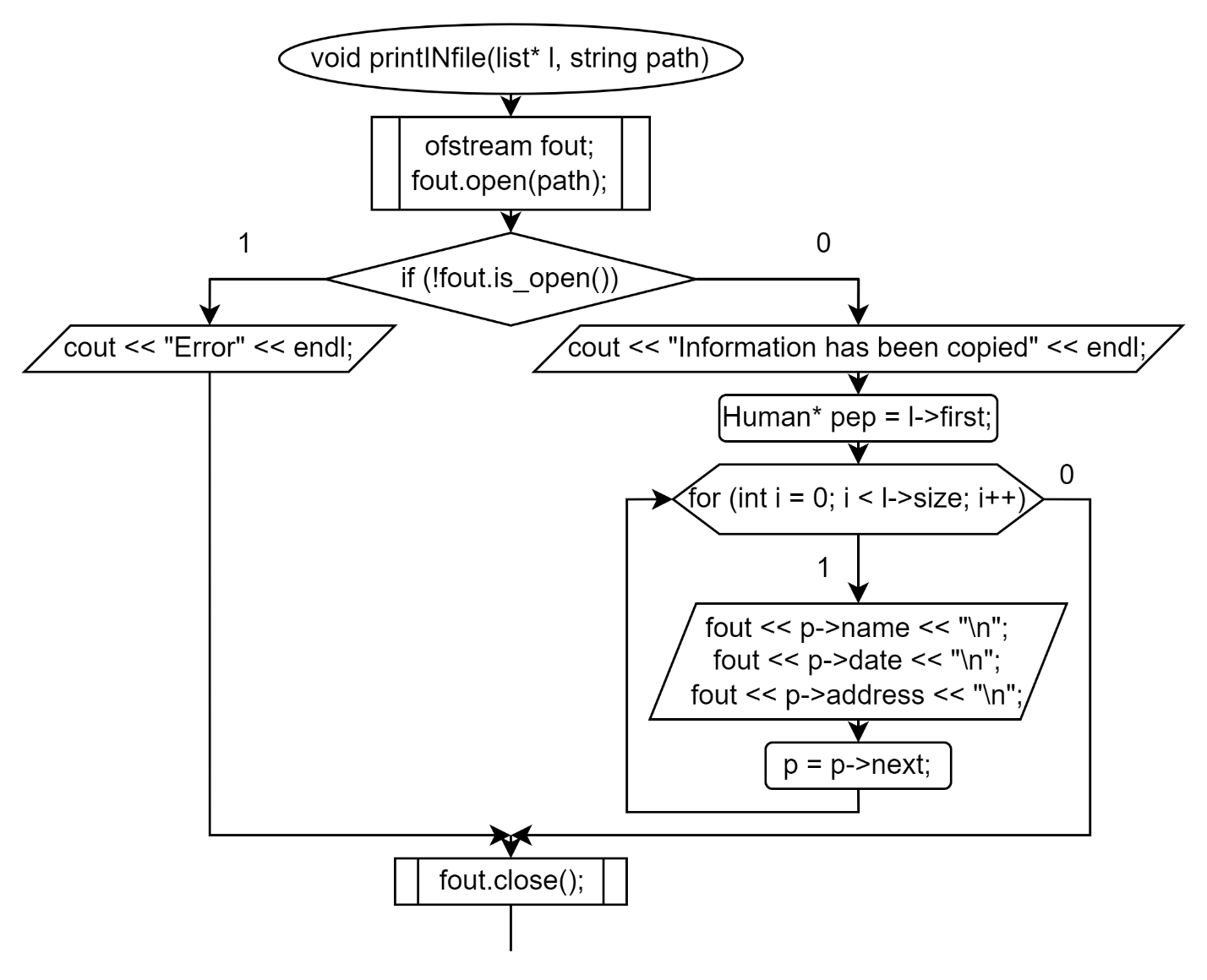


Рис. 9 – Функция void printINfile()

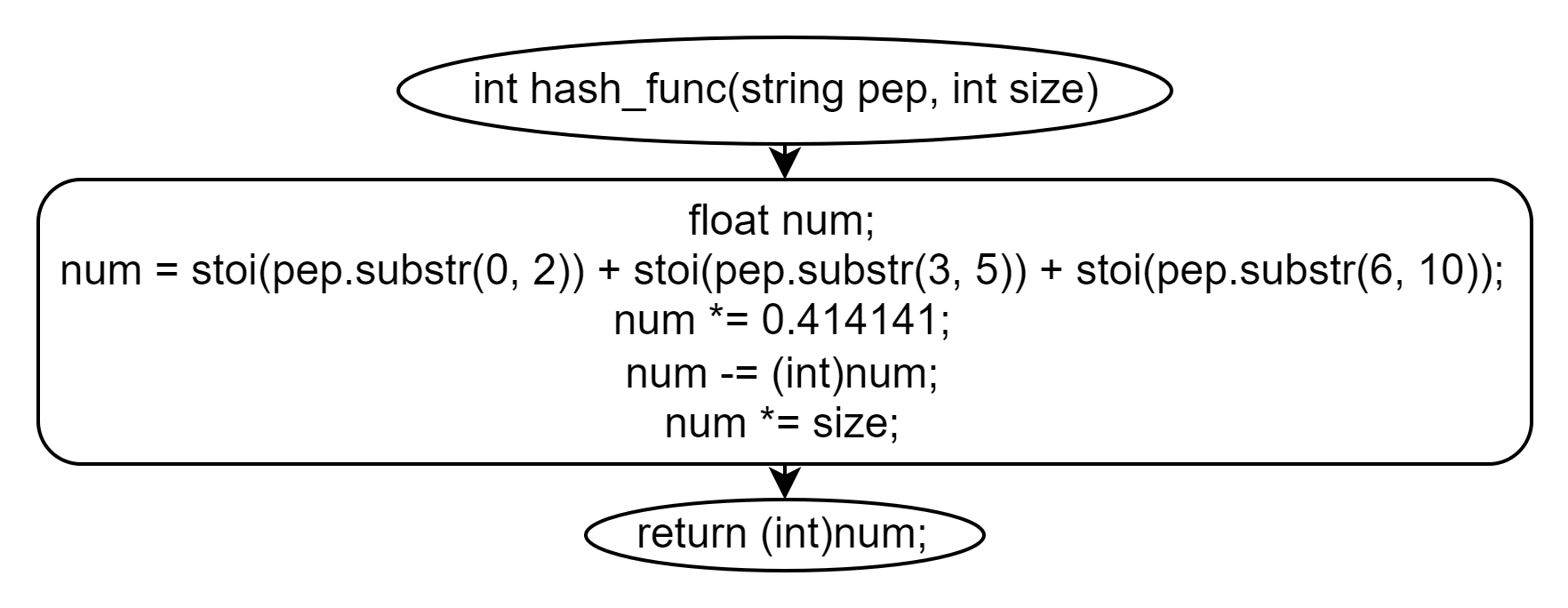


Рис. 10 – Функция void int hash\_func ()

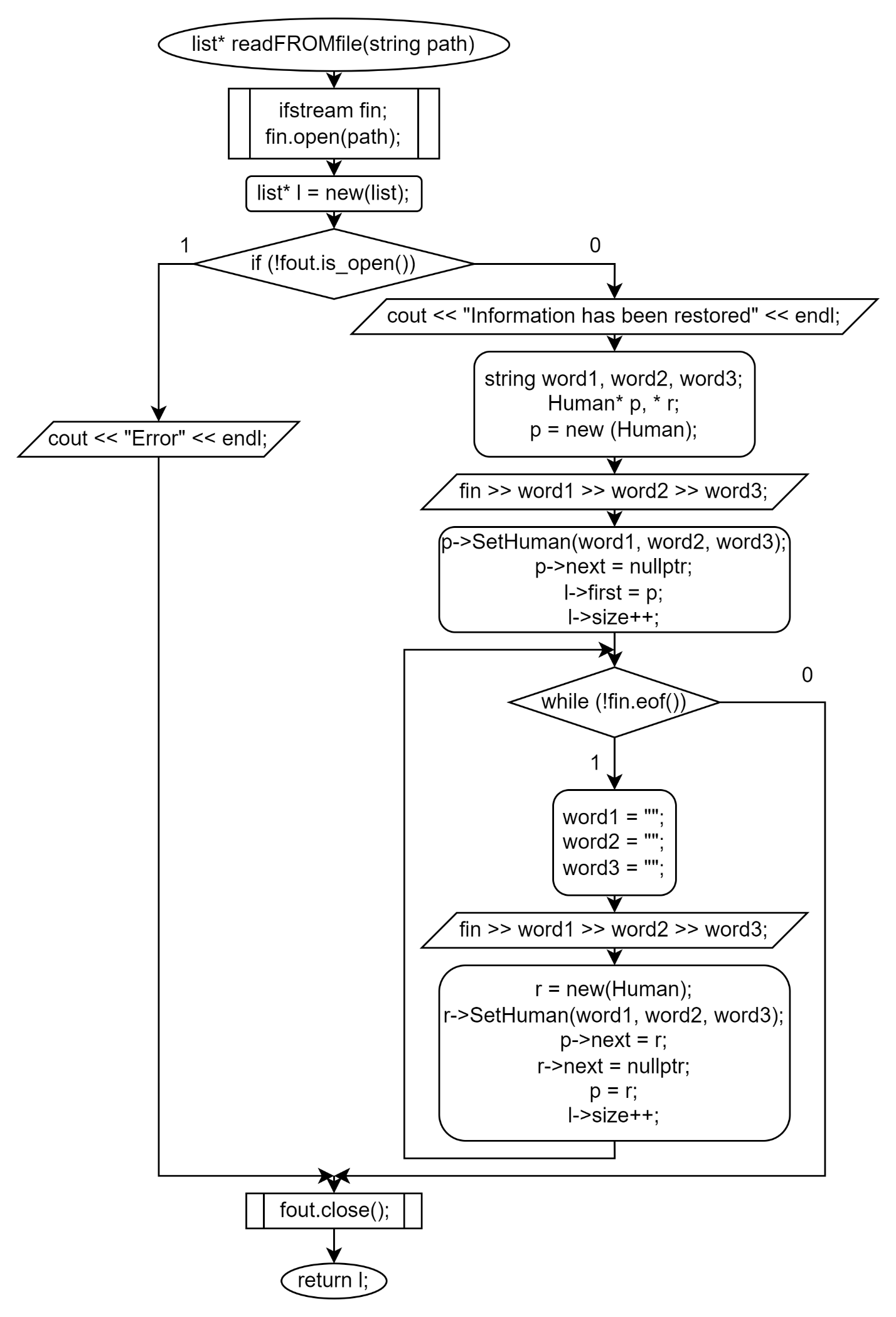


Рис. 11 – Функция list\* readFROMfile()

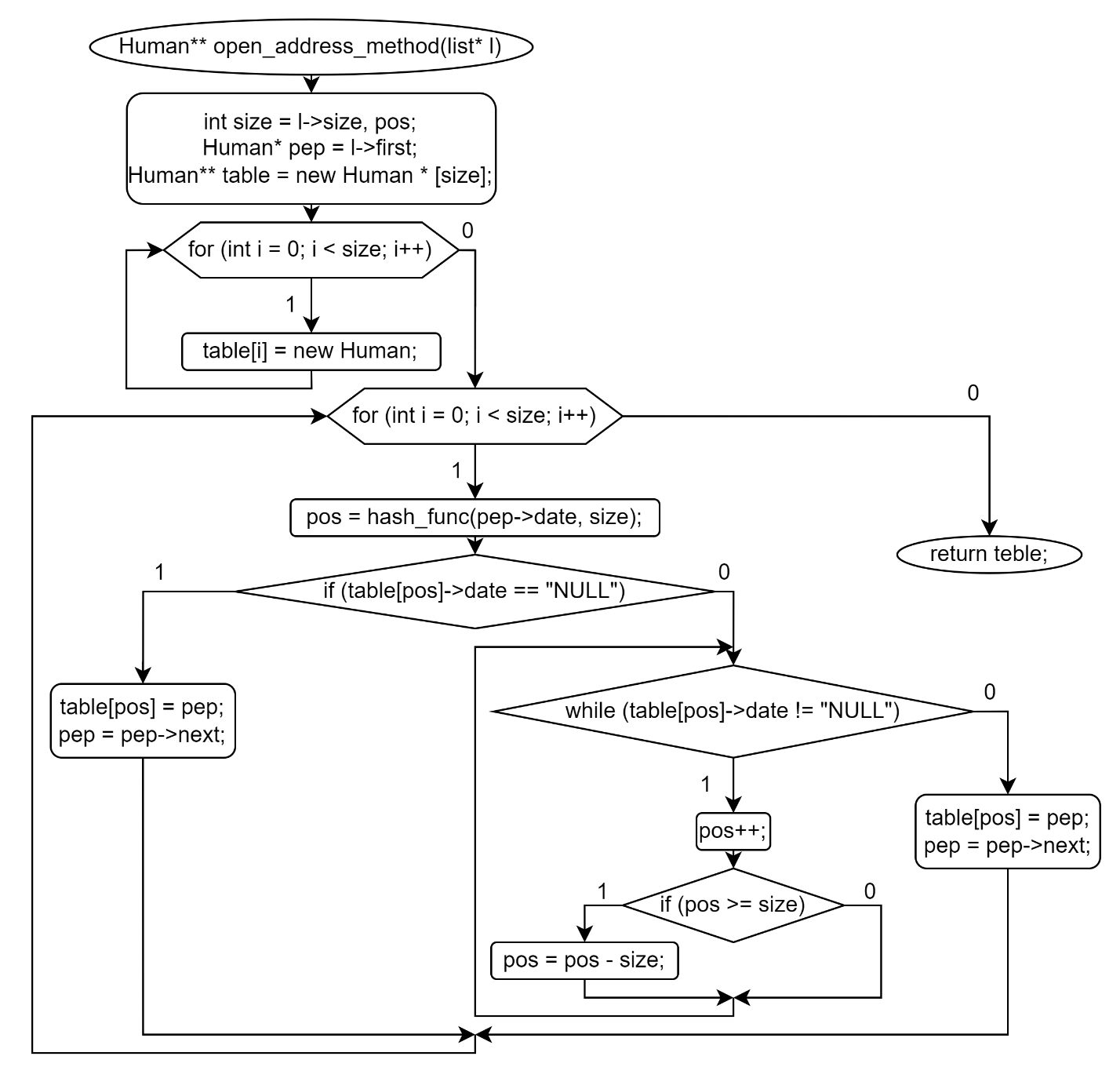


Рис. 12 – Функция Human\*\* open\_address\_method()



Рис. 13 – Функция Human\*\* chain\_method()



Рис. 14 – Функция void print\_hash()

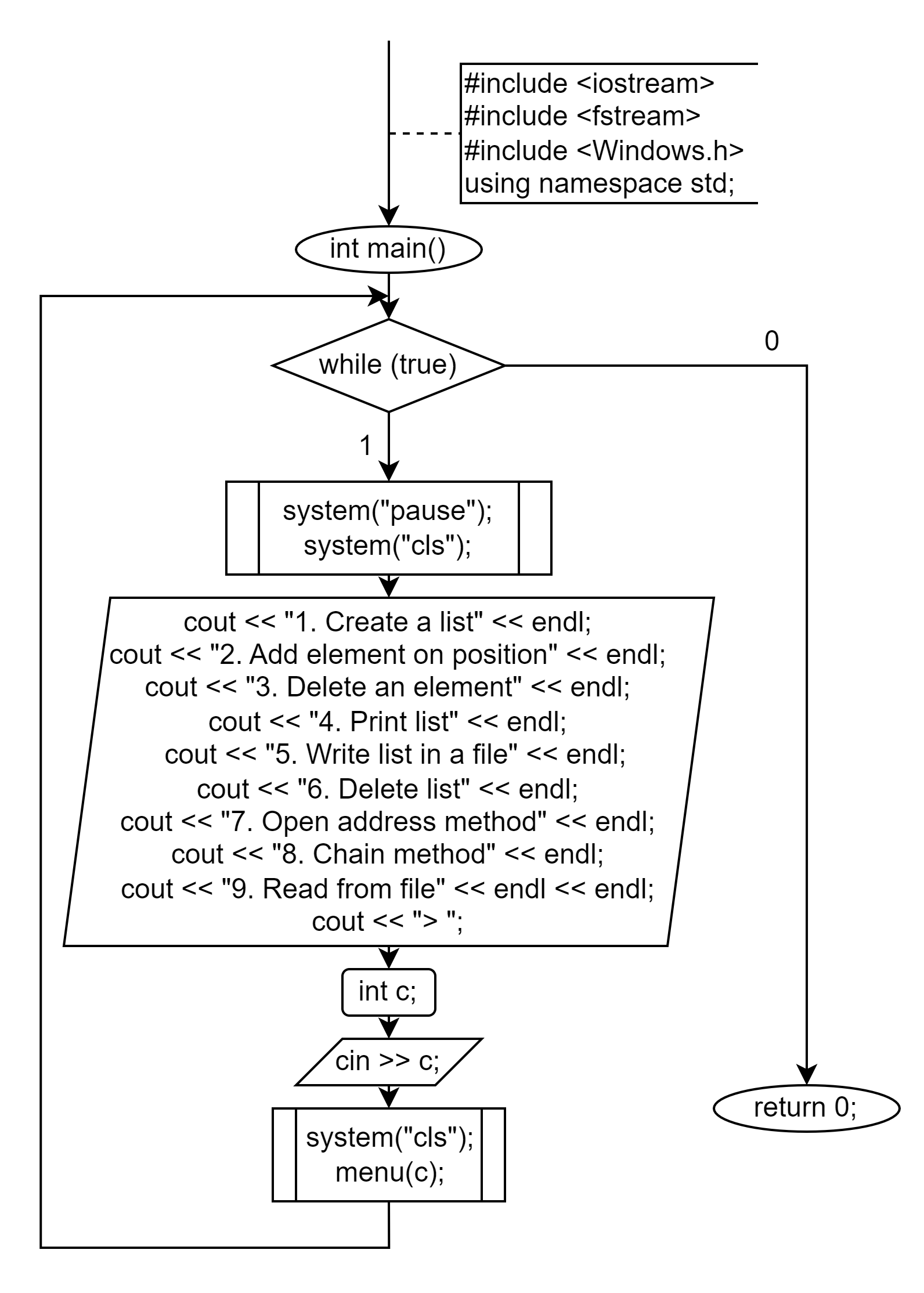


Рис. 15 – Функция int main()

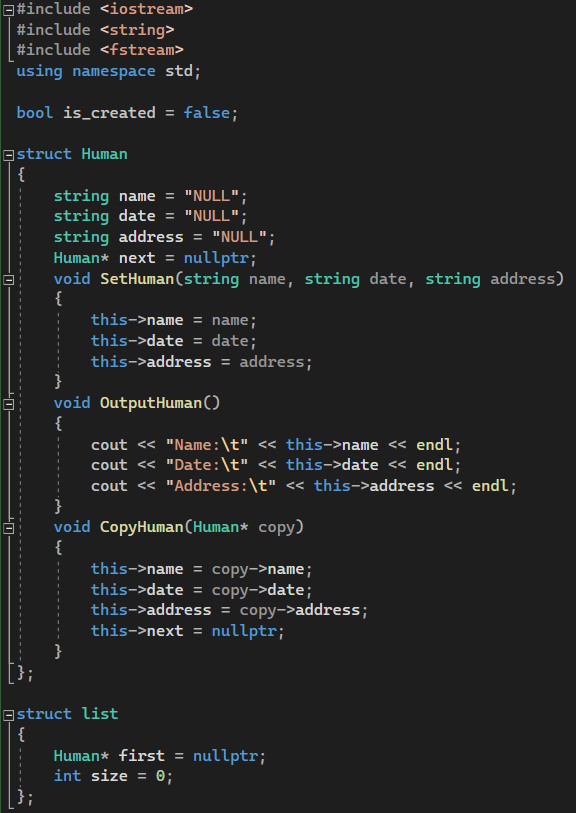


Рис. 16 – Код программы на языке С++



Рис. 17 – Код программы на языке С++

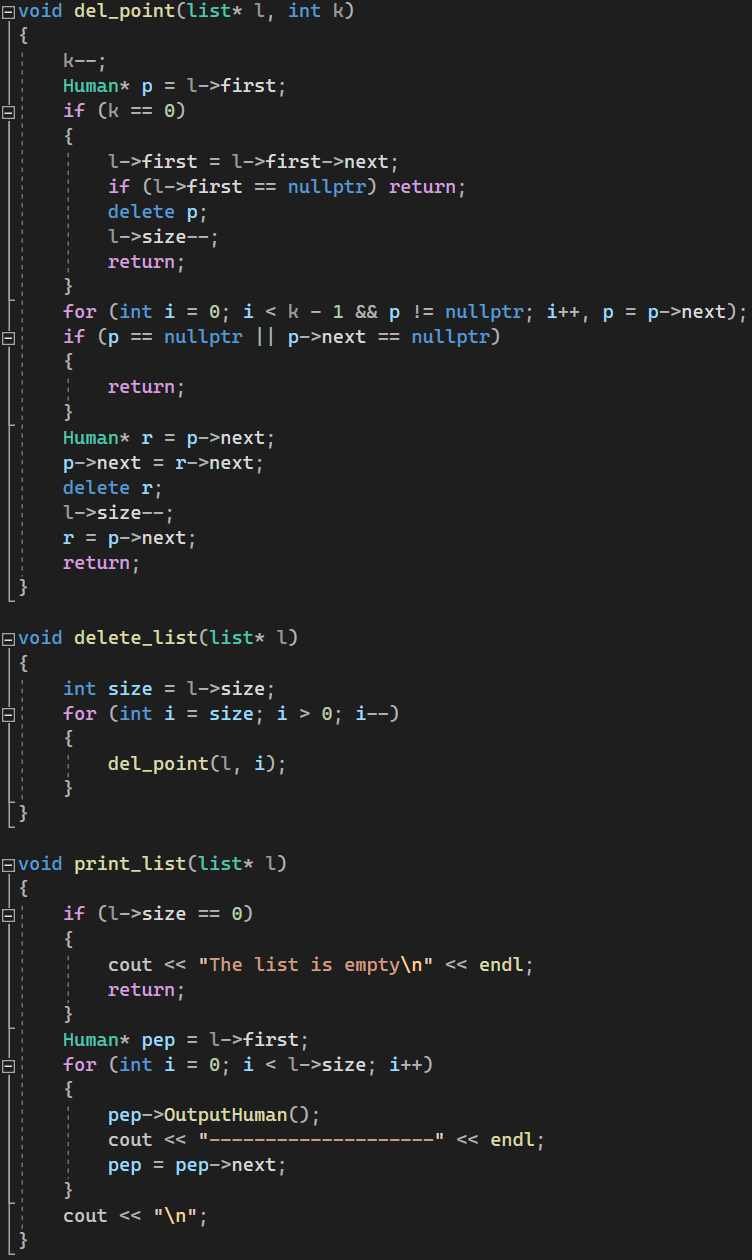


Рис. 18 – Код программы на языке С++

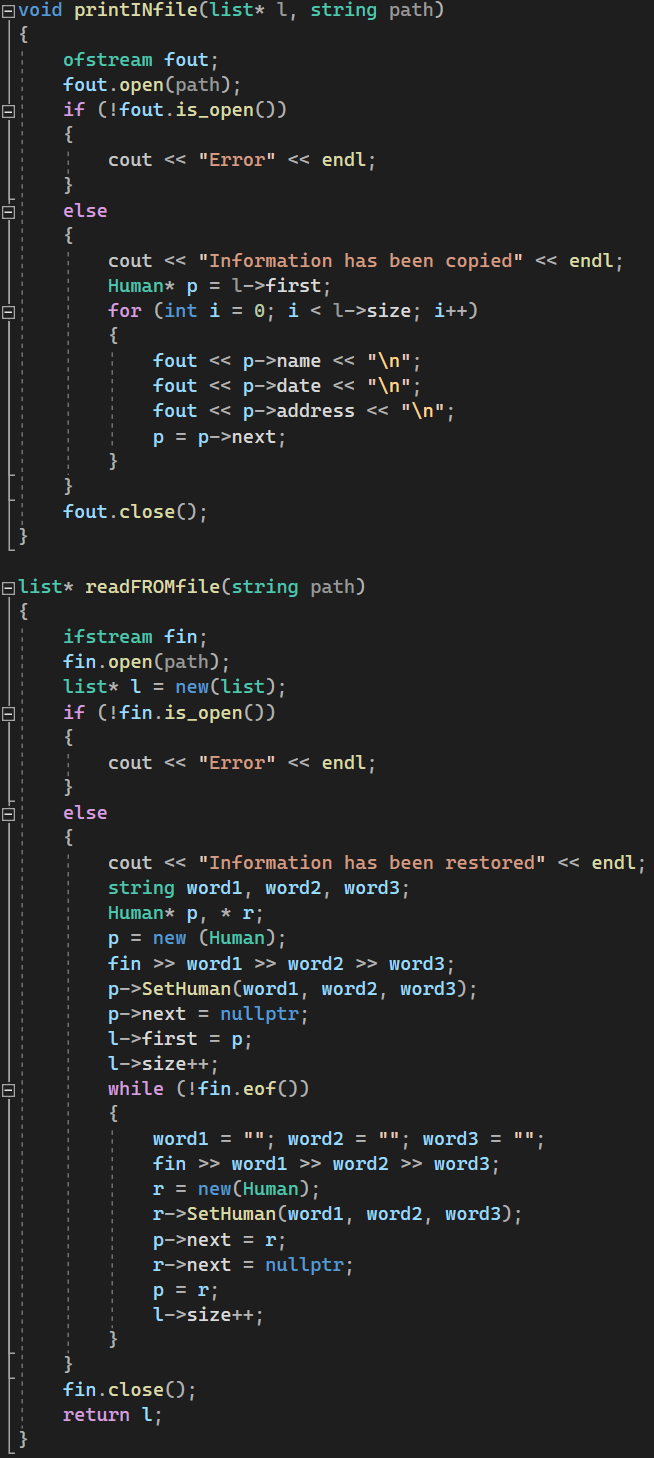


Рис. 19 – Код программы на языке С++

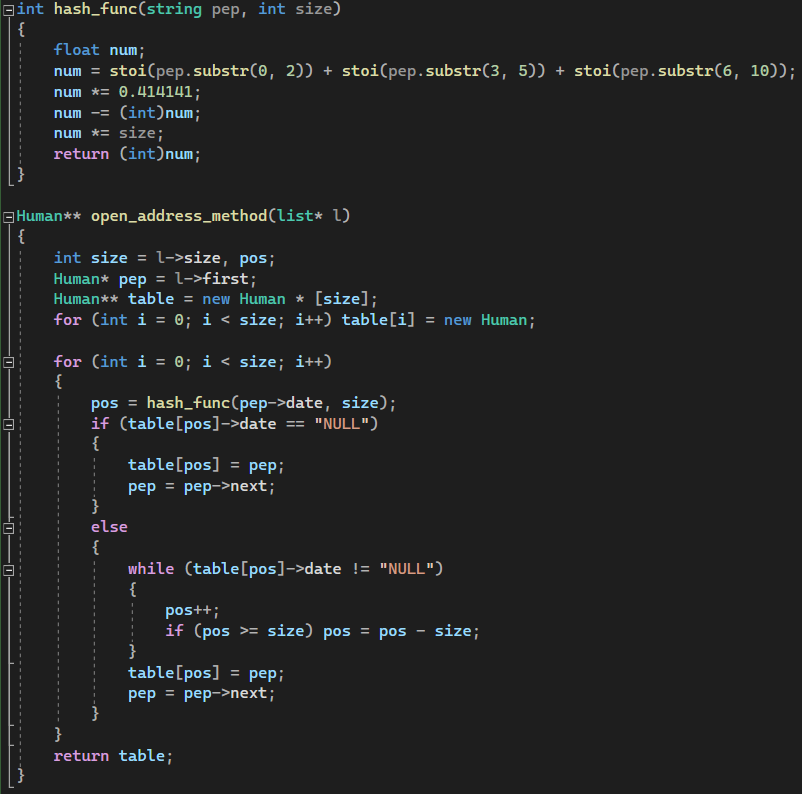


Рис. 20 – Код программы на языке С++

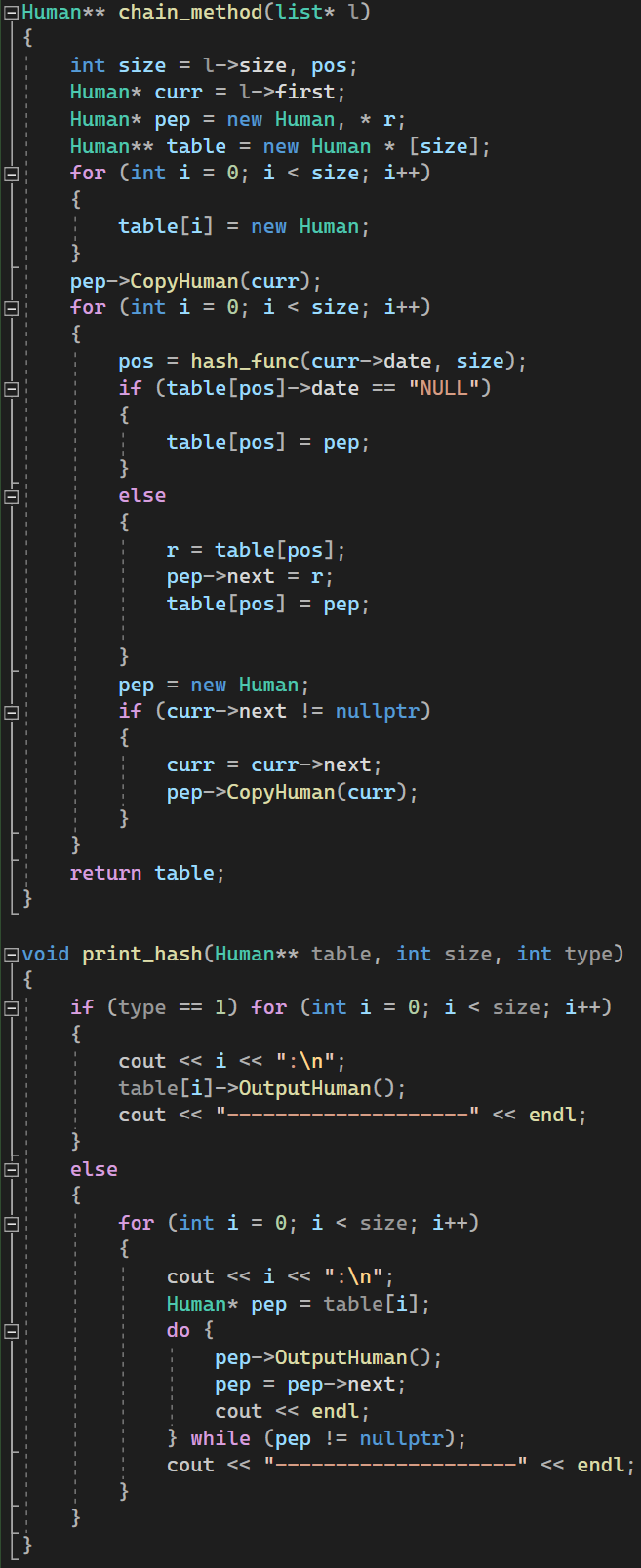


Рис. 21 – Код программы на языке С++

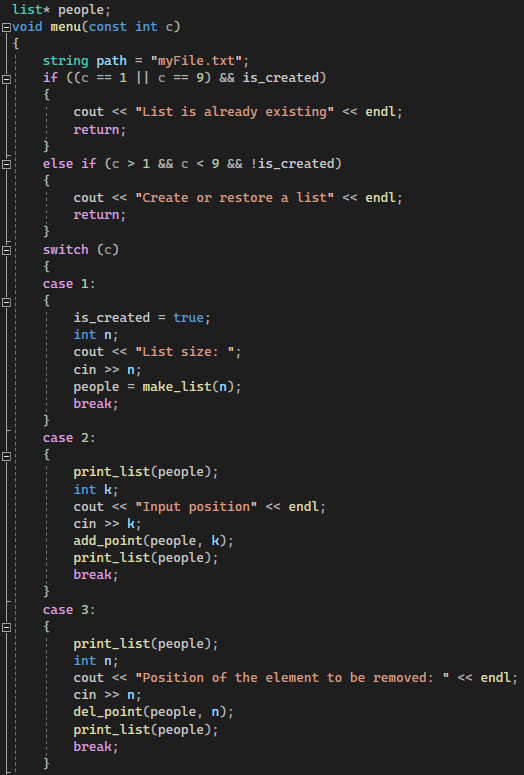


Рис. 22 – Код программы на языке С++

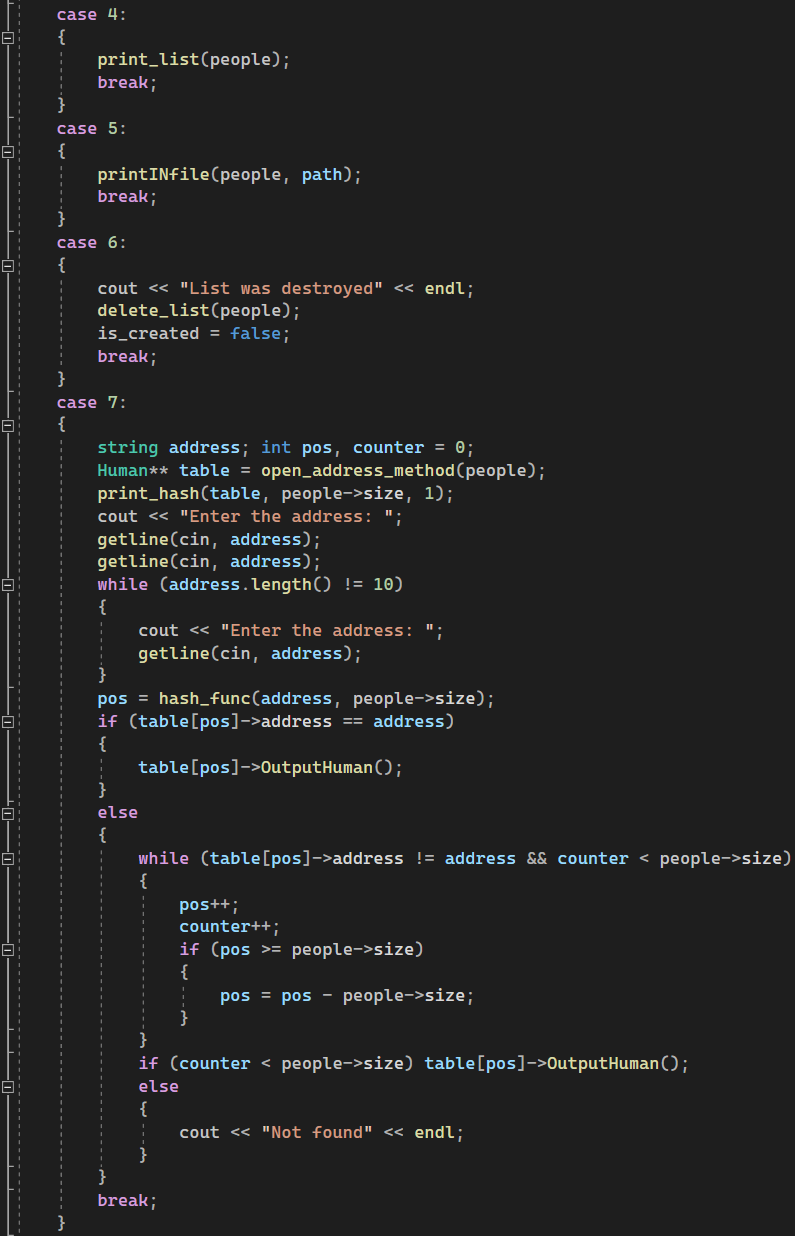


Рис. 23 – Код программы на языке С++

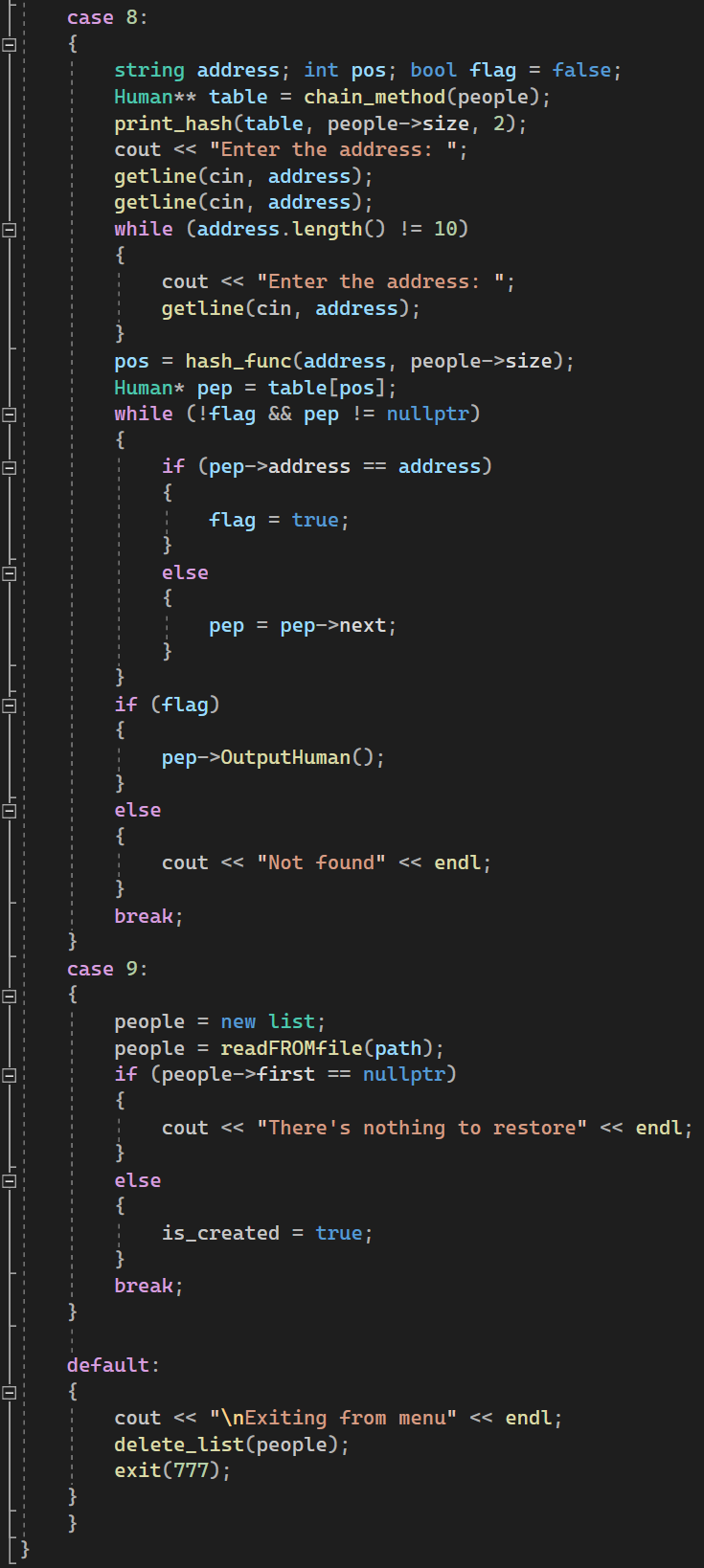


Рис. 24 – Код программы на языке С++

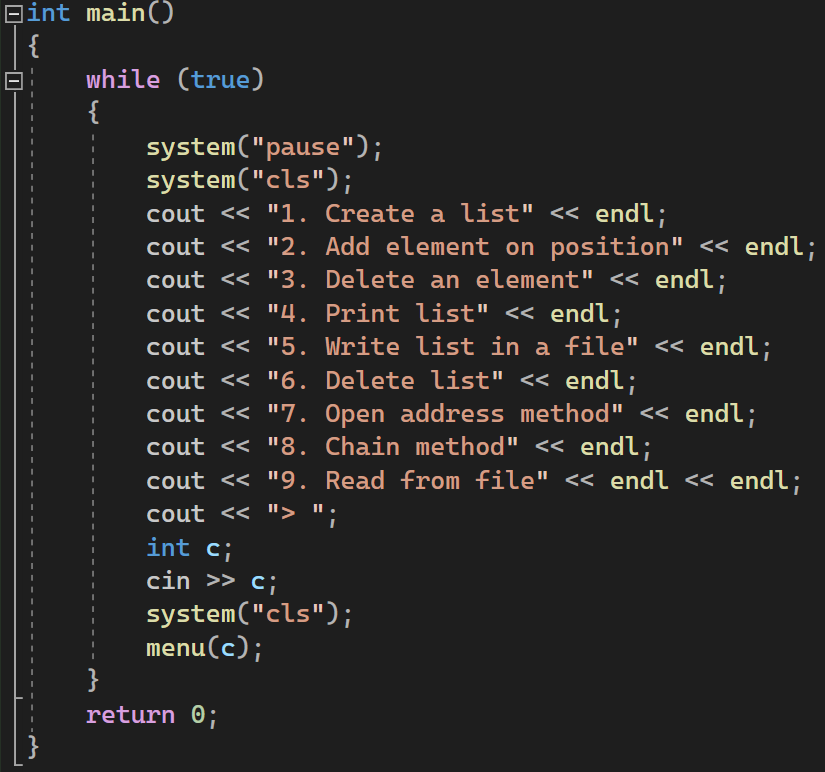


Рис. 25 – Код программы на языке С++



Рис. – Функция void menu()