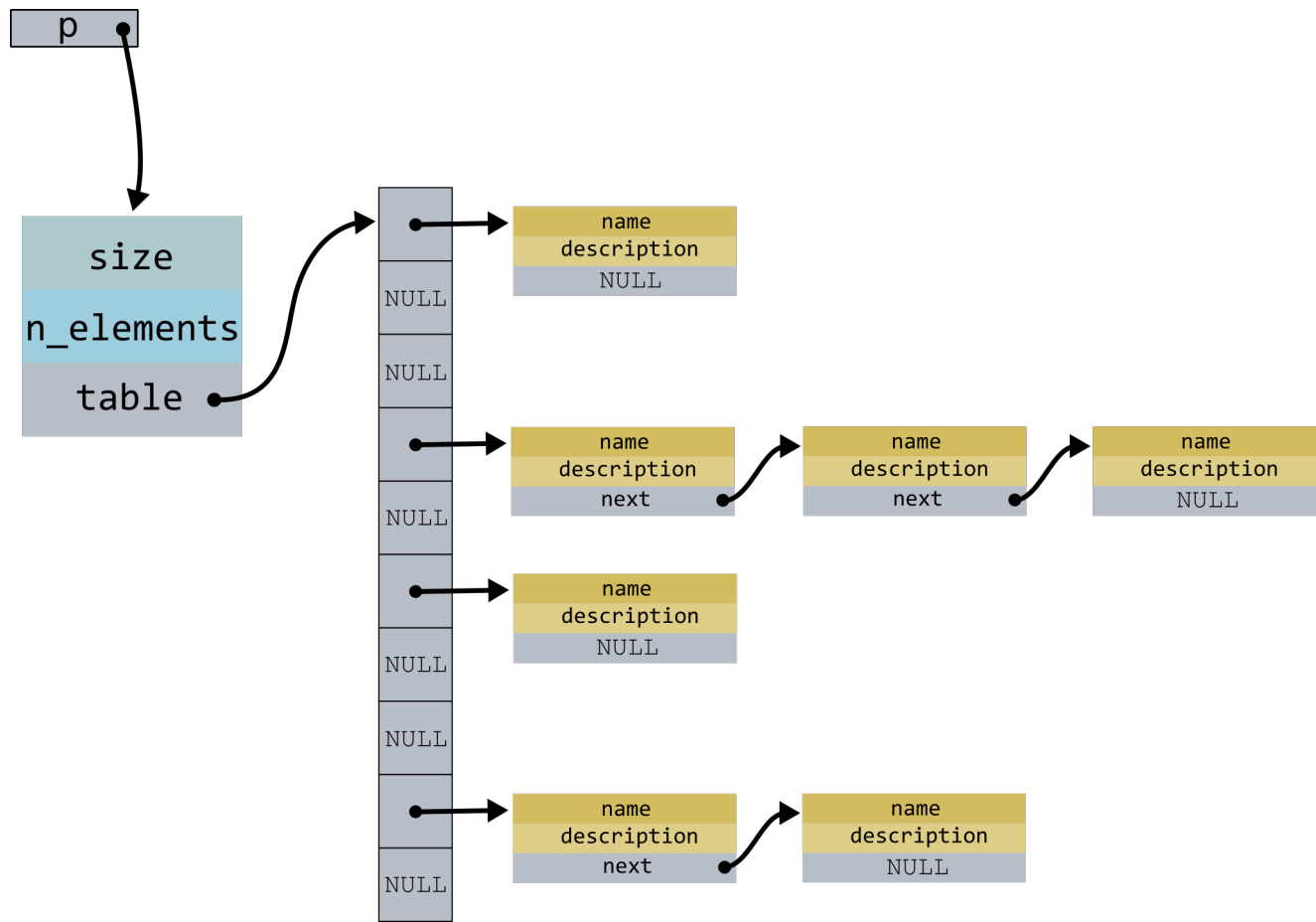


Хеш-таблицы



```
struct node
{
    char* name;
    char* description;
    struct node* next;
};
typedef struct node Node;

struct hashtable
{
    int size;
    int n_elements;
    Node** table;
};
typedef struct hashtable Hashtable;
```

```
Hashtable* hashtable_create(int size)
{
    Hashtable* ht =
        malloc(sizeof(Hashtable));

    ht->size = size;
    ht->n_elements = 0;
    ht->table = malloc(sizeof(struct node *)
        * ht->size);

    for(int i = 0; i < ht->size; i++)
        ht->table[i] = NULL;

    return ht;
}
```

Задачи:

- Написать функцию `void hashtable_insert(Hashtable* ht, char* name, char* description)`, которая будет добавлять пару (название города - его описание) в хеш-таблицу. Если город с таким названием в хеш-таблице уже есть, то ничего делать не надо.

- Написать функцию `char* hashtable_get_description(Hashtable* ht, char* name)`, которая будет находить город по его названию. Если города нет в хеш-таблице, то эта функция должна вернуть `NULL`.
- Написать программу, которая будет в бесконечном цикле требовать у пользователя название города и печатать его описание.
- Написать функцию `void hashtable_print(Hashtable* ht)`, которая будет печатать всё содержимое хеш-таблицы.
- Написать функцию `int hashtable_remove(Hashtable* ht, char* name)`, которая будет находить город в хеш-таблице и удалять его. Если города нет в хеш-таблице, то эта функция должна вернуть `0`, иначе - `1`.
- Написать функцию `void hashtable_destroy(Hashtable* ht)`, которая будет удалять все элементы хеш-таблицы и освобождать всю память.
- Написать функцию `float get_load_factor(Hashtable* ht)`, которая будет вычислять фактор загруженности хеш-таблицы (среднее количество элементов в связанном списке).
- Видоизмените функцию `hashtable_insert` так, чтобы она проверяла загруженность таблицы при каждом добавлении элемента. Если загруженность таблицы больше чем `MAX_LOAD_FACTOR`, то таблица должна будет вырасти в `GROWTH_FACTOR` раз. При этом положения каждого элемента таблицы могут измениться. Проще всего создать новую таблицу и добавить туда все элементы из старой.
- Заполните таблицу `1000000` случайными парами - название и описание. Протестируйте различные хеш-функции.