

## Сдать решение задачи В0-Максимальный блок

Ограничение времени: 1 с

Ограничение реального времени: 5 с

Ограничение памяти: 64M

### Максимальный блок

Описана структура данных

```
typedef struct list {  
    uint64_t address;  
    size_t size;  
    char comment[64];  
    struct list *next;  
} list;
```

Требуется реализовать только одну функцию, которая в данном списке находит адрес блока памяти занимающий больше всего места.

Адрес хранится в поле address, поле size - соответствующий размер данного блока. Если список пустой, то функция должна возвращать 0. Если есть несколько таких блоков, то вернуть адрес любого из них.

Прототип функции:

```
uint64_t findMaxBlock(list *head)
```

### Примеры

#### Входные данные

```
3  
140525067852320 10  
140525067852350 30  
140525067852900 100
```

#### Результат работы

```
140525067852900
```

## Сдать решение задачи В1-Всего памяти

Ограничение времени:	1 с
Ограничение реального времени:	5 с
Ограничение памяти:	64М

### Всего памяти

Описана структура данных для хранения информации об использованной памяти:

```
typedef struct list {  
void *address;  
size_t size;  
char comment[64];  
struct list *next;  
} list;
```

Требуется реализовать только одну функцию, которая анализирует данный список и возвращает сколько всего памяти используется. Адрес хранится в поле address, поле size - соответствующий размер данного блока. Если список пустой, то функция должна возвращать 0.

Прототип функции:

```
size_t totalMemoryUsage(list *head)
```

### Примеры

#### Входные данные

```
3  
140525067852320 10  
140525067852350 30  
140525067852900 100
```

#### Результат работы

```
140
```