

Analysis

Node ** ->

32	-> Node(size 32)
16	-> Node(size 16)
8	-> Node(size 8)
4	-> Node(size 4)
2	->Node(size 2)

```
(1) unsigned int init_allocator(unsigned int _basic_block_size,  
unsigned int _length);
```

I use a pointer to the pointer to the Node which is like an array. Every different elements in the array is a pointer to the Node. If the _length can separate to different values of power of two, and then we allocate a new memory for this size of node.

```
(2) int release_allocator();
```

I use free to free every memory I allocate.

```
(3) Addr my_malloc(unsigned int _length) {};
```

First I rounded the _length to the nearest power of two value, and then find the proper node to separate memory for this. In this process, we need to split the larger size to smaller size using buddy system.

```
(4) int my_free(Addr _a) {};
```

Adversely using buddy system to merge freed buddy nodes.