

## CS 3: Introduction to Software Design

### Bitops Exercises

```
bitset.c
1 typedef struct bitset {
2     uint8_t set[(USHRT_MAX / CHAR_BIT) + (USHRT_MAX & 1)];
3 } bitset_t;
4
5 bitset_t *bitset_init() {
6     bitset_t *new = malloc(sizeof(bitset_t));
7     memset(new->set, 0, sizeof(bitset_t));
8     return new;
9 }
10
11 void bitset_free(bitset_t *haystack) {
12     free(haystack);
13 }
14
15 void bitset_add(bitset_t *haystack, uint16_t needle) {
16     size_t element_number = needle / CHAR_BIT;
17     size_t bit_number = needle % CHAR_BIT;
18
19 }
20
21 bool bitset_contains(bitset_t *haystack, uint16_t needle) {
22     size_t element_number = needle / CHAR_BIT;
23     size_t bit_number = needle % CHAR_BIT;
24
25 }
26
27 void bitset_remove(bitset_t *haystack, uint16_t needle) {
28     size_t element_number = needle / CHAR_BIT;
29     size_t bit_number = needle % CHAR_BIT;
30
31 }
32
33 size_t bitset_size(bitset_t *haystack) {
34     size_t result = 0;
35
36     // Fill in this entire function.
37
38
39     return result;
40 }
```