

```
1 // Implements a list of numbers using an array of fixed length
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt for number of numbers
9     int capacity;
10    do
11    {
12        capacity = get_int("Capacity: ");
13    }
14    while (capacity < 1);
15
16    // Memory for numbers
17    int numbers[capacity];
18
19    // Prompt for numbers
20    int size = 0;
21    while (size < capacity)
22    {
23        // Prompt for number
24        int number = get_int("Number: ");
25
26        // Add to list
27        numbers[size] = number;
28        size++;
29    }
30
31    // Print numbers
32    for (int i = 0; i < size; i++)
33    {
34        printf("%i\n", numbers[i]);
35    }
36 }
```

```
1 // Implements a list of numbers using an array of dynamic length
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Memory for numbers
9     int *numbers = NULL;
10    int capacity = 0;
11
12    // Prompt for numbers (until EOF)
13    int size = 0;
14    while (true)
15    {
16        // Prompt for number
17        int number = get_int("Number: ");
18
19        // Check for EOF
20        if (number == INT_MAX)
21        {
22            break;
23        }
24
25        // Check whether enough space for number
26        if (size == capacity)
27        {
28            // Allocate space for number
29            int *tmp = realloc(numbers, sizeof(int) * (size + 1));
30            if (!tmp)
31            {
32                if (numbers)
33                {
34                    free(numbers);
35                }
36                return 1;
37            }
38            numbers = tmp;
39            capacity++;
40        }
41
42        // Add number to list
43        numbers[size] = number;
44        size++;
```

```
45     }
46
47     // Print numbers
48     printf("\n");
49     for (int i = 0; i < size; i++)
50     {
51         printf("%i\n", numbers[i]);
52     }
53
54     // Free memory
55     if (numbers)
56     {
57         free(numbers);
58     }
59 }
```

```
1 // Implements a list of numbers using a linked list
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 typedef struct node
7 {
8     int number;
9     struct node *next;
10 }
11 node;
12
13 int main(void)
14 {
15     // Memory for numbers
16     node *numbers = NULL;
17
18     // Prompt for numbers (until EOF)
19     while (true)
20     {
21         // Prompt for number
22         int number = get_int("number: ");
23
24         // Check for EOF
25         if (number == INT_MAX)
26         {
27             break;
28         }
29
30         // Allocate space for number
31         node *n = malloc(sizeof(node));
32         if (!n)
33         {
34             return 1;
35         }
36
37         // Add number to list
38         n->number = number;
39         n->next = NULL;
40         if (numbers)
41         {
42             for (node *ptr = numbers; ptr != NULL; ptr = ptr->next)
43             {
44                 if (!ptr->next)
```

```
45         {
46             ptr->next = n;
47             break;
48         }
49     }
50 }
51 else
52 {
53     numbers = n;
54 }
55 }
56
57 // Print numbers
58 printf("\n");
59 for (node *ptr = numbers; ptr != NULL; ptr = ptr->next)
60 {
61     printf("%i\n", ptr->number);
62 }
63
64 // Free memory
65 node *ptr = numbers;
66 while (ptr != NULL)
67 {
68     node *next = ptr->next;
69     free(ptr);
70     ptr = next;
71 }
72 }
```

```
1 // http://valgrind.org/docs/manual/quick-start.html#quick-start.prepare
2
3 #include <stdlib.h>
4
5 void f(void)
6 {
7     int *x = malloc(10 * sizeof(int));
8     x[10] = 0;
9 }
10
11 int main(void)
12 {
13     f();
14     return 0;
15 }
```

```
1 // Demonstrates structs
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 #include "struct.h"
8
9 int main(void)
10 {
11     // Allocate space for students
12     int enrollment = get_int("Enrollment: ");
13     student students[enrollment];
14
15     // Prompt for students' names and dorms
16     for (int i = 0; i < enrollment; i++)
17     {
18         students[i].name = get_string("Name: ");
19         students[i].dorm = get_string("Dorm: ");
20     }
21
22     // Print students' names and dorms
23     for (int i = 0; i < enrollment; i++)
24     {
25         printf("%s is in %s.\n", students[i].name, students[i].dorm);
26     }
27 }
```

```
1 // Demonstrates file I/O
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7
8 #include "struct.h"
9
10 int main(void)
11 {
12     // Allocate memory for students
13     int enrollment = get_int("Enrollment: ");
14     student students[enrollment];
15
16     // Prompt for students' names and dorms
17     for (int i = 0; i < enrollment; i++)
18     {
19         students[i].name = get_string("Name: ");
20         students[i].dorm = get_string("Dorm: ");
21     }
22
23     // Save students to disk
24     FILE *file = fopen("students.csv", "w");
25     if (file)
26     {
27         for (int i = 0; i < enrollment; i++)
28         {
29             fprintf(file, "%s,%s\n", students[i].name, students[i].dorm);
30         }
31         fclose(file);
32     }
33 }
```

```
1 // Represents a student
2
3 typedef struct
4 {
5     char *name;
6     char *dorm;
7 }
8 student;
```