
```
1 // Printing a command-line argument
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(int argc, string argv[])
7 {
8     if (argc == 2)
9     {
10         printf("hello, %s\n", argv[1]);
11     }
12     else
13     {
14         printf("hello, world\n");
15     }
16 }
```

```
1 // Printing command-line arguments
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(int argc, string argv[])
7 {
8     for (int i = 0; i < argc; i++)
9     {
10         printf("%s\n", argv[i]);
11     }
12 }
```

```
1 // Printing characters in an array of strings
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(int argc, string argv[])
8 {
9     for (int i = 0; i < argc; i++)
10    {
11        for (int j = 0, n = strlen(argv[i]); j < n; j++)
12        {
13            printf("%c\n", argv[i][j]);
14        }
15        printf("\n");
16    }
17 }
```

```
1 // Explicitly casts chars to ints
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     string s = get_string("String: ");
10    for (int i = 0; i < strlen(s); i++)
11    {
12        int c = (int) s[i];
13        printf("%c %i\n", s[i], c);
14    }
15 }
```

```
1 // Implicitly casts chars to ints
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     string s = get_string("String: ");
10    for (int i = 0; i < strlen(s); i++)
11    {
12        printf("%c %i\n", s[i], s[i]);
13    }
14 }
```

```
1 // Iterative binary search
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 // Names in a phone book
8 string book[] = {
9     "Chen",
10    "Kernighan",
11    "Leitner",
12    "Lewis",
13    "Malan",
14    "Muller",
15    "Seltzer",
16    "Shieber",
17    "Smith"
18 };
19
20 int main(void)
21 {
22     // Prompt user for name
23     string name = get_string("Name: ");
24
25     // Search for name
26     int left = 0, right = sizeof(book) / sizeof(string) - 1;
27     while (left <= right)
28     {
29         // Look at middle
30         int middle = (left + right) / 2;
31         if (strcmp(name, book[middle]) == 0)
32         {
33             printf("Calling %s\n", name);
34             return 0;
35         }
36
37         // Search left half
38         else if (strcmp(name, book[middle]) < 0)
39         {
40             right = middle - 1;
41         }
42
43         // Search right half
44         else if (strcmp(name, book[middle]) > 0)
```

```
45     {
46         left = middle + 1;
47     }
48 }
49 printf("Quitting\n");
50 return 1;
51 }
```

```
1 // Recursive binary search
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 // Names in a phone book
8 string book[] = {
9     "Chen",
10    "Kernighan",
11    "Leitner",
12    "Lewis",
13    "Malan",
14    "Muller",
15    "Seltzer",
16    "Shieber",
17    "Smith"
18 };
19
20 bool search(string name, string names[], int left, int right);
21
22 int main(void)
23 {
24     // Prompt user for name
25     string name = get_string("Name: ");
26
27     // Search for name
28     if (search(name, book, 0, sizeof(book) / sizeof(string) - 1))
29     {
30         printf("Calling %s\n", name);
31     }
32     else
33     {
34         printf("Quitting\n");
35     }
36 }
37
38 // Searches names for name
39 bool search(string name, string names[], int left, int right)
40 {
41     // No more names to search
42     if (left > right)
43     {
44         return false;
```



```
45     }
46
47     // Look at middle
48     int middle = (left + right) / 2;
49     if (strcmp(name, names[middle]) == 0)
50     {
51         return true;
52     }
53
54     // Search left half
55     else if (strcmp(name, names[middle]) < 0)
56     {
57         return search(name, names, left, middle - 1);
58     }
59
60     // Search right half
61     else if (strcmp(name, names[middle]) > 0)
62     {
63         return search(name, names, middle + 1, right);
64     }
65
66     return false;
67 }
```

```
1 // Buggy example for help50
2
3 int main(void)
4 {
5     printf("hello, world\n")
6 }
```

```
1 // Buggy example for help50
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     string s = get_string("Name: ");
8     printf("hello, %s\n", s);
9 }
```

```
1 // Buggy example for printf
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     for (int i = 0; i <= 10; i++)
8     {
9         printf("#\n");
10    }
11 }
```

```
1 // Capitalizes a string
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     string s = get_string("Before: ");
10    printf("After: ");
11    for (int i = 0, n = strlen(s); i < n; i++)
12    {
13        if (s[i] >= 'a' && s[i] <= 'z')
14        {
15            printf("%c", s[i] - ('a' - 'A'));
16        }
17        else
18        {
19            printf("%c", s[i]);
20        }
21    }
22    printf("\n");
23 }
```

```
1 // Capitalizes string using ctype library (and an unnecessary condition)
2
3 #include <cs50.h>
4 #include <ctype.h>
5 #include <stdio.h>
6 #include <string.h>
7
8 int main(void)
9 {
10     string s = get_string("Before: ");
11     printf("After: ");
12     for (int i = 0, n = strlen(s); i < n; i++)
13     {
14         if (islower(s[i]))
15         {
16             printf("%c", toupper(s[i]));
17         }
18         else
19         {
20             printf("%c", s[i]);
21         }
22     }
23     printf("\n");
24 }
```

```
1 // Capitalizes string using ctype library
2
3 #include <cs50.h>
4 #include <ctype.h>
5 #include <stdio.h>
6 #include <string.h>
7
8 int main(void)
9 {
10     string s = get_string("Before: ");
11     printf("After: ");
12     for (int i = 0, n = strlen(s); i < n; i++)
13     {
14         printf("%c", toupper(s[i]));
15     }
16     printf("\n");
17 }
```

```
1 // Returns explicit value from main
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(int argc, string argv[])
7 {
8     if (argc != 2)
9     {
10         printf("missing command-line argument\n");
11         return 1;
12     }
13     printf("hello, %s\n", argv[1]);
14     return 0;
15 }
```



```
1 // Linear search
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 // Names in a phone book
8 string book[] = {
9     "Chen",
10    "Kernighan",
11    "Leitner",
12    "Lewis",
13    "Malan",
14    "Muller",
15    "Seltzer",
16    "Shieber",
17    "Smith"
18 };
19
20 int main(void)
21 {
22     // Prompt user for name
23     string name = get_string("Name: ");
24
25     // Search for name
26     for (int i = 0; i < sizeof(book) / sizeof(string); i++)
27     {
28         if (strcmp(name, book[i]) == 0)
29         {
30             printf("Calling %s\n", name);
31             return 0;
32         }
33     }
34     printf("Quitting\n");
35 }
```

```
1 // Generates a bar chart of three scores
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Get scores from user
9     int score1 = get_int("Score 1: ");
10    int score2 = get_int("Score 2: ");
11    int score3 = get_int("Score 3: ");
12
13    // Generate first bar
14    printf("Score 1: ");
15    for (int i = 0; i < score1; i++)
16    {
17        printf("#");
18    }
19    printf("\n");
20
21    // Generate second bar
22    printf("Score 2: ");
23    for (int i = 0; i < score2; i++)
24    {
25        printf("#");
26    }
27    printf("\n");
28
29    // Generate third bar
30    printf("Score 3: ");
31    for (int i = 0; i < score3; i++)
32    {
33        printf("#");
34    }
35    printf("\n");
36 }
```

```
1 // Generates a bar chart of three scores
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 void chart(int score);
7
8 int main(void)
9 {
10     // Get scores from user
11     int score1 = get_int("Score 1: ");
12     int score2 = get_int("Score 2: ");
13     int score3 = get_int("Score 3: ");
14
15     // Chart first score
16     printf("Score 1: ");
17     chart(score1);
18
19     // Chart second score
20     printf("Score 2: ");
21     chart(score2);
22
23     // Chart third score
24     printf("Score 3: ");
25     chart(score3);
26 }
27
28 // Generate bar
29 void chart(int score)
30 {
31     // Output one hash per point
32     for (int i = 0; i < score; i++)
33     {
34         printf("#");
35     }
36     printf("\n");
37 }
```

```
1 // Generates a bar chart of three scores using an array
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 void chart(int score);
7
8 int main(void)
9 {
10     // Get scores from user
11     int scores[3];
12     for (int i = 0; i < 3; i++)
13     {
14         scores[i] = get_int("Score %i: ", i + 1);
15     }
16
17     // Chart scores
18     for (int i = 0; i < 3; i++)
19     {
20         printf("Score %i: ", i + 1);
21         chart(scores[i]);
22     }
23 }
24
25 // Generate bar
26 void chart(int score)
27 {
28     // Output one hash per point
29     for (int i = 0; i < score; i++)
30     {
31         printf("#");
32     }
33     printf("\n");
34 }
```

```
1 // Generates a bar chart of three scores by using an array and using a constant
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 const int COUNT = 3;
7
8 void chart(int score);
9
10 int main(void)
11 {
12     // Get scores from user
13     int scores[COUNT];
14     for (int i = 0; i < COUNT; i++)
15     {
16         scores[i] = get_int("Score %i: ", i + 1);
17     }
18
19     // Chart scores
20     for (int i = 0; i < COUNT; i++)
21     {
22         printf("Score %i: ", i + 1);
23         chart(scores[i]);
24     }
25 }
26
27 // Generate bar
28 void chart(int score)
29 {
30     // Output one hash per point
31     for (int i = 0; i < score; i++)
32     {
33         printf("#");
34     }
35     printf("\n");
36 }
```

```
1 // Generates a bar chart of three scores by passing an array, using a constant
2
3 #include <cs50.h>
4 #include <math.h>
5 #include <stdio.h>
6
7 const int COUNT = 3;
8
9 void chart(int count, int scores[]);
10
11 int main(void)
12 {
13     // Get scores from user
14     int scores[COUNT];
15     for (int i = 0; i < COUNT; i++)
16     {
17         scores[i] = get_int("Score %i: ", i + 1);
18     }
19
20     // Chart scores
21     chart(COUNT, scores);
22 }
23
24 // Generate bars
25 void chart(int count, int scores[])
26 {
27     // Output one hash per point
28     for (int i = 0; i < count; i++)
29     {
30         for (int j = 0; j < scores[i]; j++)
31         {
32             printf("#");
33         }
34         printf("\n");
35     }
36 }
```

```
1 // Generates a narrower bar chart of three scores by passing an array, using a constant
2
3 #include <cs50.h>
4 #include <math.h>
5 #include <stdio.h>
6
7 const int COUNT = 3;
8
9 void chart(int count, int scores[]);
10
11 int main(void)
12 {
13     // Get scores from user
14     int scores[COUNT];
15     for (int i = 0; i < COUNT; i++)
16     {
17         scores[i] = get_int("Score %i: ", i + 1);
18     }
19
20     // Chart scores
21     chart(COUNT, scores);
22 }
23
24 // Generate bars
25 void chart(int count, int scores[])
26 {
27     // Output one tenth as many hashes
28     for (int i = 0; i < count; i++)
29     {
30         // Calculate width
31         int width = (int) round((float) scores[i] / 10);
32
33         // Generate bar
34         printf("Score %i: ", i);
35         for (int j = 0; j < width; j++)
36         {
37             printf("#");
38         }
39         printf("\n");
40     }
41 }
```

```
1 // Prints string char by char
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     string s = get_string("Input: ");
10    printf("Output: ");
11    for (int i = 0; i < strlen(s); i++)
12    {
13        printf("%c\n", s[i]);
14    }
15 }
```

```
1 // Prints string char by char, one per line
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     string s = get_string("Input: ");
10    printf("Output:\n");
11    for (int i = 0, n = strlen(s); i < n; i++)
12    {
13        printf("%c\n", s[i]);
14    }
15 }
```

```
1 // Determines the length of a string
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt for user's name
9     string s = get_string("Name: ");
10
11     // Count number of characters up until '\0' (aka NUL)
12     int n = 0;
13     while (s[n] != '\0')
14     {
15         n++;
16     }
17     printf("%i\n", n);
18 }
```

```
1 // Poorly styled example for style50
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("hello, world\n");
8 }
```

```
1 // Poorly styled example for style50
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("hello, world\n");
8 }
```