

```
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 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 // 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 name = get_string("What's your name?\n");
8     printf("hello, %s\n", name);
9 }
```

```
1 // Buggy example for printf and debug50
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 // Buggy example for help50, debug50
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int get_negative_int(void);
7
8 int main(void)
9 {
10     int i = get_negative_int();
11     printf("%i\n", i);
12 }
13
14 // Prompt user for positive integer
15 int get_negative_int(void)
16 {
17     do
18     {
19         int n = get_int("Negative Integer: ");
20     }
21     while (n < 0);
22     return n;
23 }
```

```
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 // Prints ASCII codes
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     char c1 = 'H';
8     char c2 = 'I';
9     char c3 = '!';
10    printf("%i %i %i\n", c1, c2, c3);
11 }
```



```
1 // Stores names using an array
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     // Names
10    string names[4];
11    names[0] = "EMMA";
12    names[1] = "RODRIGO";
13    names[2] = "BRIAN";
14    names[3] = "DAVID";
15
16    // Print Emma's name
17    printf("%s\n", names[0]);
18    printf("%c%c%c%c\n", names[0][0], names[0][1], names[0][2], names[0][3]);
19 }
```

```
1 // Averages three numbers
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Scores
9     int score1 = 72;
10    int score2 = 73;
11    int score3 = 33;
12
13    // Print average
14    printf("Average: %i\n", (score1 + score2 + score3) / 3);
15 }
```

```
1 // Averages three numbers using an array
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Scores
9     int scores[3];
10    scores[0] = 72;
11    scores[1] = 73;
12    scores[2] = 33;
13
14    // Print average
15    printf("Average: %i\n", (scores[0] + scores[1] + scores[2]) / 3);
16 }
```

```
1 // Averages three numbers using an array and a constant
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 const int N = 3;
7
8 int main(void)
9 {
10     // Scores
11     int scores[N];
12     scores[0] = 72;
13     scores[1] = 73;
14     scores[2] = 33;
15
16     // Print average
17     printf("Average: %i\n", (scores[0] + scores[1] + scores[2]) / N);
18 }
```

```
1 // Averages numbers using a helper function
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 float average(int length, int array[]);
7
8 int main(void)
9 {
10     // Get number of scores
11     int n = get_int("Scores: ");
12
13     // Get scores
14     int scores[n];
15     for (int i = 0; i < n; i++)
16     {
17         scores[i] = get_int("Score %i: ", i + 1);
18     }
19
20     // Print average
21     printf("Average: %.1f\n", average(n, scores));
22 }
23
24 float average(int length, int array[])
25 {
26     int sum = 0;
27     for (int i = 0; i < length; i++)
28     {
29         sum += array[i];
30     }
31     return (float) sum / (float) length;
32 }
```

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

```
1 // Prints string char by char, one per line, using strlen
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", s[i]);
14    }
15    printf("\n");
16 }
```

```
1 // Prints string char by char, one per line, using strlen, remembering string's length
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, n = strlen(s); i < n; i++)
12    {
13        printf("%c", s[i]);
14    }
15    printf("\n");
16 }
```



```
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 }
```

```
1 // Uppercases 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] - 32);
16        }
17        else
18        {
19            printf("%c", s[i]);
20        }
21    }
22    printf("\n");
23 }
```

```
1 // Uppercases 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 // Uppercases 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 }
```