

```
1.  /**
2.   * ages.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Ages people by a year.
8.   *
9.   * Demonstrates arrays.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // determine number of people
18.     int n;
19.     do
20.     {
21.         printf("Number of people in room: ");
22.         n = GetInt();
23.     }
24.     while (n < 1);
25.
26.     // declare array in which to store everyone's age
27.     int ages[n];
28.
29.     // get everyone's age
30.     for (int i = 0; i < n; i++)
31.     {
32.         printf("Age of person #%i: ", i + 1);
33.         ages[i] = GetInt();
34.     }
35.
36.     // report everyone's age a year hence
37.     printf("Time passes...\n");
38.     for (int i = 0; i < n; i++)
39.     {
40.         printf("A year from now, person #%i will be %i years old.\n", i + 1, ages[i] + 1);
41.     }
42. }
```

```
1.  /**
2.   * argv-0.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Prints program's first command-line argument; assumes it's present.
8.   *
9.   * Demonstrates use of argv.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(int argc, string argv[])
16. {
17.     printf("%s\n", argv[1]);
18. }
```

```
1.  /**
2.   * argv-1.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Prints command-line arguments, one per line.
8.   *
9.   * Demonstrates use of argv.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(int argc, string argv[])
16. {
17.     // print arguments
18.     for (int i = 0; i < argc; i++)
19.     {
20.         printf("%s\n", argv[i]);
21.     }
22. }
```

```
1.  /**
2.   * argv-2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Prints command-line arguments, one character per line.
8.   *
9.   * Demonstrates argv as a two-dimensional array.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14. #include <string.h>
15.
16. int main(int argc, string argv[])
17. {
18.     // print arguments
19.     for (int i = 0; i < argc; i++)
20.     {
21.         for (int j = 0, n = strlen(argv[i]); j < n; j++)
22.         {
23.             printf("%c\n", argv[i][j]);
24.         }
25.         printf("\n");
26.     }
27. }
```

```
1.  /**
2.   * argv-3.c
3.   *
4.   * Rob Bowden
5.   * rob@cs.harvard.edu
6.   *
7.   * Prints command-line arguments, one character per line.
8.   *
9.   * Uses own version of strlen function.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int my_strlen(string s)
16. {
17.     int length = 0;
18.     while(s[length] != '\0')
19.     {
20.         length++;
21.     }
22.     return length;
23. }
24.
25. int main(int argc, string argv[])
26. {
27.     // print arguments
28.     for (int i = 0; i < argc; i++)
29.     {
30.         for (int j = 0, n = my_strlen(s); j < n; j++)
31.         {
32.             printf("%c\n", argv[i][j]);
33.         }
34.         printf("\n");
35.     }
36. }
```

```
1.  /**
2.   * argv-4.c
3.   *
4.   * Rob Bowden
5.   * rob@cs.harvard.edu
6.   *
7.   * Prints command-line arguments, one character per line.
8.   *
9.   * Checks for '\0' instead of using strlen.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(int argc, string argv[])
16. {
17.     // print arguments
18.     for (int i = 0; i < argc; i++)
19.     {
20.         for (int j = 0; argv[i][j] != '\0'; j++)
21.         {
22.             printf("%c\n", argv[i][j]);
23.         }
24.         printf("\n");
25.     }
26. }
```

```
1.  /**
2.   * debug.c
3.   *
4.   * Rob Bowden
5.   * rob@cs.harvard.edu
6.   *
7.   * Intentionally buggy program that (in theory) never terminates.
8.   *
9.   * Meant to be debugged with GDB!
10.  */
11.
12.
13. #include <stdio.h>
14. #include <cs50.h>
15.
16. void foo(int i)
17. {
18.     while (i != 0)
19.     {
20.         i = i - 3;
21.     }
22.     printf("%i\n", i);
23. }
24.
25. int main(void)
26. {
27.     printf("Enter an integer: ");
28.     int i = GetInt();
29.
30.     while (i > 10)
31.     {
32.         i--;
33.     }
34.
35.     foo(i);
36. }
```