The C Programming Language: Chapter 1

Eric Bailey

March 4, 2018 [1]

Write an abstract

Contents

Hello, world!

Covers Exercises 1-1 and 1-2.

1
```
⟨hello.c 1⟩≡
  ⟨Include the standard I/O functions. 6d⟩


  int main()
  {
      printf(”Hello, world!\n”);
  }
```
Uses `printf` 6d.
Root chunk (not used in this document).

[1] Last updated June 20, 2018

Fahrenheit-Celsius table

Covers Exercises 1-3, 1-4, and 1-5.

2a ⟨*fahrcels.c* 2a⟩≡

```
⟨Include the standard I/O functions. 6d⟩
⟨Include the standard string functions. 7⟩
```

This definition is continued in chunks 2 and 3.
Root chunk (not used in this document).

Declare some useful constants.

2b ⟨*fahrcels.c* 2a⟩+≡

```
#define LOWER 0
#define UPPER 300
#define STEP  20
```

Defines:
`LOWER`, used in chunk 3a.
`STEP`, used in chunk 3a.
`UPPER`, used in chunk 3a.

Exercise 1-3

2c ⟨*fahrcels.c* 2a⟩+≡

```
void print_header(char lhs[], char rhs[])
{
    printf(”| %s | %s |\n”, lhs, rhs);
    putchar(’|’);
    for (int i = -2; i < (int)strlen(lhs); ++i)
        putchar(’-’);
    putchar(’+’);
    for (int i = -2; i < (int)strlen(rhs); ++i)
        putchar(’-’);
    puts(”|”);
}
```

Defines:
`print_header`, used in chunks 2d and 3a.
Uses `printf` 6d, `putchar` 6d, `puts` 6d, and `strlen` 7.

Exercise 1-4

2d ⟨*fahrcels.c* 2a⟩+≡

```
void celsfahr()
{
    print_header(”Celsius”, ”Fahrenheit”);
    for (int celsius = 0; celsius ≤ 300; celsius += 20)
        printf(”| %7d | %10.0f |\n”, celsius, 32.0 + (9.0/5.0) * celsius);
}
```

Defines:
`celsfahr`, used in chunk 3b.
Uses `printf` 6d and `print_header` 2c.

Exercise 1-5

3a ⟨*fahrcels.c* 2a⟩+≡

```
void fahrcels()
{
    print_header(”Fahrenheit”, ”Celsius”);
    for (int fahr = UPPER; fahr ≥ LOWER; fahr -= STEP)
        printf(”| %10d | %7.1f |\n”, fahr, (5.0/9.0) * (fahr-32.0));
}
```

Defines:
`fahrcels`, used in chunk 3b.
Uses `LOWER` 2b, `STEP` 2b, `UPPER` 2b, `printf` 6d, and `print_header` 2c.

The `main` *function*

3b ⟨*fahrcels.c* 2a⟩+≡

```
int main()
{
    fahrcels();
    puts(”\n”);
    celsfahr();

    return 0;
}
```

Uses `celsfahr` 2d, `fahrcels` 3a, and `puts` 6d.

Copy

Covers Exercises 1-6 and 1-7.

3c ⟨*copy.c* 3c⟩≡

```
⟨Include the standard I/O functions. 6d⟩


int main()
{
    int c;
    while ((c = getchar()) ≠ EOF)
        putchar(c);

    return 0;
}
```

Uses `putchar` 6d.
Root chunk (not used in this document).

Character Counting

4a ⟨*wc.c* 4a⟩≡

```
⟨Include the standard I/O functions. 6d⟩
⟨Include the boolean type and values. 6c⟩
```

This definition is continued in chunks 4–6.
Root chunk (not used in this document).

4b ⟨*wc.c* 4a⟩+≡

```
double char_count()
{
    double nc;

    for (nc = 0; getchar() != EOF; ++nc)
        ;

    return nc;
}
```

Defines:
`char_count`, never used.

Line Counting

4c ⟨*the character is a newline* 4c⟩≡

```
c == '\n'
```

This code is used in chunks 4 and 6b.

4d ⟨*wc.c* 4a⟩+≡

```
int line_count()
{
    int c, nl;

    nl = 0;
    while ((c = getchar()) != EOF)
        if (⟨the character is a newline 4c⟩)
            ++nl;

    return nl;
}
```

Defines:
`line_count`, never used.

Exercise 1-8

For our purposes, whitespace is a space, tab, or newline.

4e ⟨*the character is whitespace* 4e⟩≡

```
c == ' ' || ⟨the character is a newline 4c⟩ || c == '\t'
```

This code is used in chunks 5a and 6b.

5a ⟨*wc.c* 4a⟩+≡

```
bool is_whitespace(int c)
{
    return (⟨the character is whitespace 4e⟩);
}
```

Defines:
  is_whitespace, used in chunk 5b.
Uses bool 6c.

5b ⟨*wc.c* 4a⟩+≡

```
double ws_count()
{
    double ns = 0;
    int c = 0;

    while ((c = getchar()) != EOF)
        if (is_whitespace(c))
            ++ns;

    return ns;
}
```

Defines:
  ws_count, never used.
Uses is_whitespace 5a.

Exercise 1-9

5c ⟨*catblanks.c* 5c⟩≡

```
⟨Include the standard I/O functions. 6d⟩
⟨Include the boolean type and values. 6c⟩


int main()
{
    int c;
    bool prev_blank = false;

    while ((c = getchar()) != EOF) {
        if (!(prev_blank && c == ' '))
            putchar(c);
        prev_blank = (c == ' ');
    }


    return 0;
}
```

Uses bool 6c and putchar 6d.
Root chunk (not used in this document).

Exercise 1-10

Word Counting

6a ⟨*wc.c* 4a⟩+≡

```
#define IN  1
#define OUT 0
```

Defines:
`IN`, used in chunk 6b.
`OUT`, used in chunk 6b.

6b ⟨*wc.c* 4a⟩+≡

```
int main()
{
    int c, nl, nw, nc, state;

    state = OUT;
    nl = nw = nc = 0;
    while ((c = getchar()) != EOF) {
        ++nc;
        if (⟨the character is a newline 4c⟩)
            ++nl;
        if (⟨the character is whitespace 4e⟩)
            state = OUT;
        else if (state == OUT) {
          state = IN;
          ++nw;
        }
    }

    printf(”%7d%8d%8d\n”, nl, nw, nc);

    return 0;
}
```

Uses `IN` 6a, `OUT` 6a, and `printf` 6d.

Exercise 1-11

Exercise 1-12

Common Headers

6c ⟨*Include the boolean type and values.* 6c⟩≡

```
#include <stdbool.h>
```

Defines:
`bool`, used in chunk 5.
This code is used in chunks 4a and 5c.

6d ⟨*Include the standard I/O functions.* 6d⟩≡

```
#include <stdio.h>
```

Defines:
`printf`, used in chunks 1–3 and 6b.
`putchar`, used in chunks 2c, 3c, and 5c.
`puts`, used in chunks 2c and 3b.
This code is used in chunks 1–5.

7 ⟨*Include the standard string functions.* 7⟩≡

```
#include <string.h>
```

Defines:
`strlen`, used in chunk 2c.
This code is used in chunk 2a.

Chunks

Index