The C Programming Language: Chapter 1

Eric Bailey

March 4, 2018 [1]

[1] Last updated October 15, 2018

Write an abstract

Contents

Hello, world!

Covers Exercises 1-1 and 1-2.

1 ⟨*hello.c* 1⟩≡
⟨*Include the standard I/O functions.* 7c⟩

```
int main()
{
    printf("Hello, world!\n");
}
```

Uses `printf` 7c.
Root chunk (not used in this document).

Fahrenheit-Celsius table

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

2a ⟨*fahrcels.c* 2a⟩≡

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

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 7c, putchar 7c, puts 7c, and strlen 7d.

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 7c 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` 7c, 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` 7c.

Copy

Covers Exercises 1-6 and 1-7.

3e ⟨*copy.c* 3e⟩≡

⟨*Include the standard I/O functions.* 7c⟩

```
int main()
{
    int c;
    ⟨For each character c until EOF 3c⟩
        ⟨Print the character. 3d⟩

    return 0;
}
```

Root chunk (not used in this document).

3c ⟨*For each character* `c` *until* `EOF` 3c⟩≡

```
while ((c = getchar()) ≠ EOF)
```

This code is used in chunks 3–5 and 7a.

3d ⟨*Print the character.* 3d⟩≡

```
putchar(c);
```

Uses `putchar` 7c.
This code is used in chunks 3e, 5b, and 6f.

Character Counting

3f ⟨*wc.c* 3f⟩≡

⟨*Include the standard I/O functions.* 7c⟩
⟨*Include the boolean type and values.* 7b⟩

This definition is continued in chunks 4–7.
Root chunk (not used in this document).

4a ⟨*wc.c* 3f⟩+≡
```
double char_count()
{
    double nc;

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

    return nc;
}
```
Defines:
char_count, never used.

Line Counting

4b ⟨*the character is a newline* 4b⟩≡
```
c == '\n'
```
This code is used in chunks 4 and 7a.

4c ⟨*wc.c* 3f⟩+≡
```
int line_count()
{
    int c, nl;

    nl = 0;
    ⟨For each character c until EOF 3c⟩
        if (⟨the character is a newline 4b⟩)
            ++nl;

    return nl;
}
```
Defines:
line_count, never used.

Exercise 1-8

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

4d ⟨*the character is a tab* 4d⟩≡
```
c == '\t'
```
This code is used in chunks 4e and 6a.

4e ⟨*the character is whitespace* 4e⟩≡
```
c == ' ' || ⟨the character is a newline 4b⟩ || ⟨the character is a tab 4d⟩
```
This code is used in chunks 4f and 7a.

4f ⟨*wc.c* 3f⟩+≡
```
bool is_whitespace(int c)
{
    return (⟨the character is whitespace 4e⟩);
}
```
Defines:
is_whitespace, used in chunk 5a.
Uses bool 7b.

5a ⟨*wc.c* 3f⟩+≡

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

    ⟨For each character c until EOF 3c⟩
        if (is_whitespace(c))
            ++ns;

    return ns;
}
```

Defines:
ws_count, never used.
Uses is_whitespace 4f.

Exercise 1-9

5b ⟨*catblanks.c* 5b⟩≡

```
⟨Include the standard I/O functions. 7c⟩
⟨Include the boolean type and values. 7b⟩


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

    ⟨For each character c until EOF 3c⟩ {
        if (!(prev_blank && c == ' '))
            ⟨Print the character. 3d⟩
        prev_blank = (c == ' ');
    }


    return 0;
}
```

Uses bool 7b.
Root chunk (not used in this document).

Exercise 1-10

Process each character c.

5d ⟨*unambiguous.c* 5c⟩+≡

```
int c;

    ⟨For each character c until EOF 3c⟩ {
```

5c ⟨*unambiguous.c* 5c⟩≡

```
⟨Include the standard I/O functions. 7c⟩


int main()
{
```

This definition is continued in chunks 5 and 6.
Root chunk (not used in this document).

Replace each tab by \t.

6a ⟨*unambiguous.c* 5c⟩+≡
```
if (⟨the character is a tab 4d⟩)
    fputs("\\t", stdout);
```
Uses fputs 7c and stdout 7c.

Replace each backspace by \b.

6c ⟨*unambiguous.c* 5c⟩+≡
```
else if (⟨the character is a backspace 6b⟩)
    fputs("\\b", stdout);
```
Uses fputs 7c and stdout 7c.

Replace each backslash by \\.

6e ⟨*unambiguous.c* 5c⟩+≡
```
else if (⟨the character is a backslash 6d⟩)
    fputs("\\\\", stdout);
```
Uses fputs 7c and stdout 7c.

Otherwise print the character unchanged.

6f ⟨*unambiguous.c* 5c⟩+≡
```
else
    ⟨Print the character. 3d⟩
```

Word Counting

6h ⟨*wc.c* 3f⟩+≡
```
#define IN  1
#define OUT 0
```
Defines:
IN, used in chunk 7a.
OUT, used in chunk 7a.

6b ⟨*the character is a backspace* 6b⟩≡
```
c == '\b'
```
This code is used in chunk 6c.

6d ⟨*the character is a backslash* 6d⟩≡
```
c == '\\'
```
This code is used in chunk 6e.

Finally, close the **while** loop and exit.

6g ⟨*unambiguous.c* 5c⟩+≡
```
    }

    return 0;
}
```

7a ⟨*wc.c* 3f⟩+≡

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

    state = OUT;
    nl = nw = nc = 0;
    ⟨For each character c until EOF 3c⟩ {
        ++nc;
        if (⟨the character is a newline 4b⟩)
            ++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 6h, OUT 6h, and printf 7c.

Exercise 1-12

Common Headers

7b ⟨*Include the boolean type and values.* 7b⟩≡

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

Defines:
bool, used in chunks 4f and 5b.
This code is used in chunks 3f and 5b.

7c ⟨*Include the standard I/O functions.* 7c⟩≡

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

Defines:
fputs, used in chunk 6.
printf, used in chunks 1–3 and 7a.
putchar, used in chunks 2c and 3d.
puts, used in chunks 2c and 3b.
stdout, used in chunk 6.
This code is used in chunks 1–3 and 5.

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

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

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

Chunks

Index