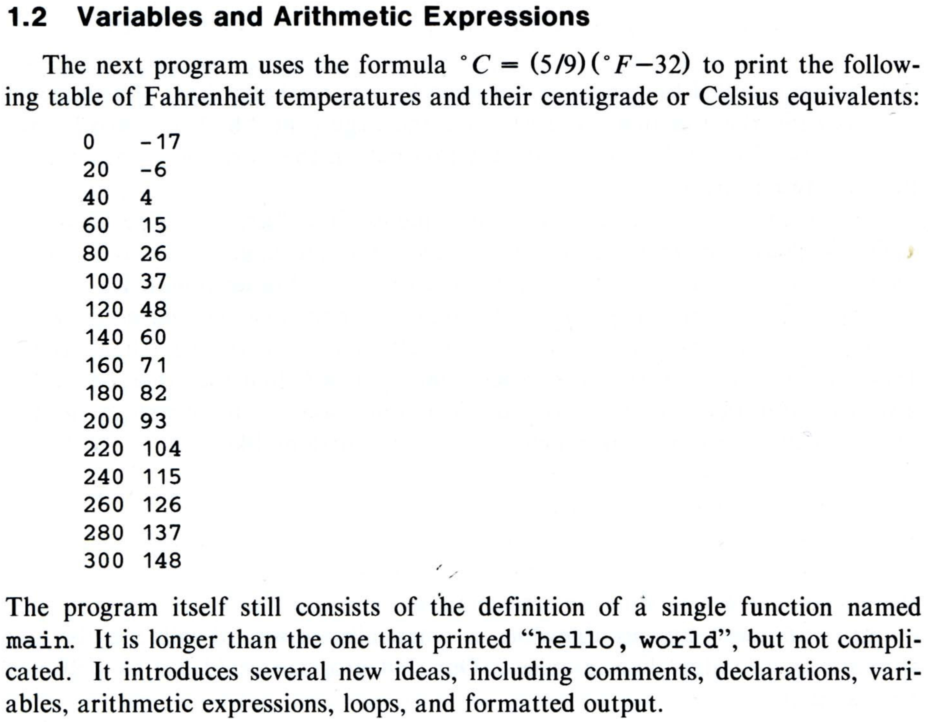
Excerpts from Kernighan and Ritchie, “The C Programming Language”, 2nd Ed.

Fahrenheit-Celsius Calculation Example Program

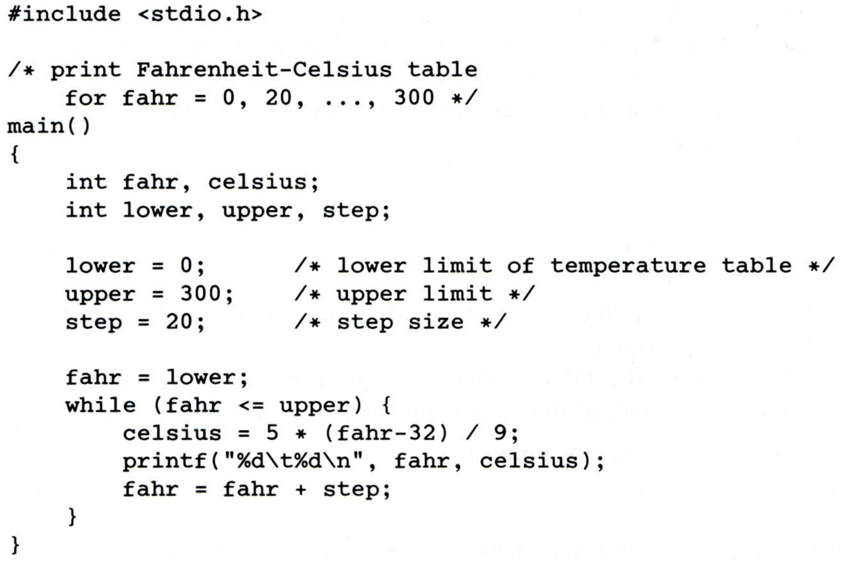


1.2 Variables and Arithmetic Expressions

The next program uses the formula $^\circ C = (5/9)(^\circ F-32)$ to print the following table of Fahrenheit temperatures and their centigrade or Celsius equivalents:

```
0    -17
20   -6
40   4
60   15
80   26
100  37
120  48
140  60
160  71
180  82
200  93
220  104
240  115
260  126
280  137
300  148
```

The program itself still consists of the definition of a single function named `main`. It is longer than the one that printed “`hello, world`”, but not complicated. It introduces several new ideas, including comments, declarations, variables, arithmetic expressions, loops, and formatted output.



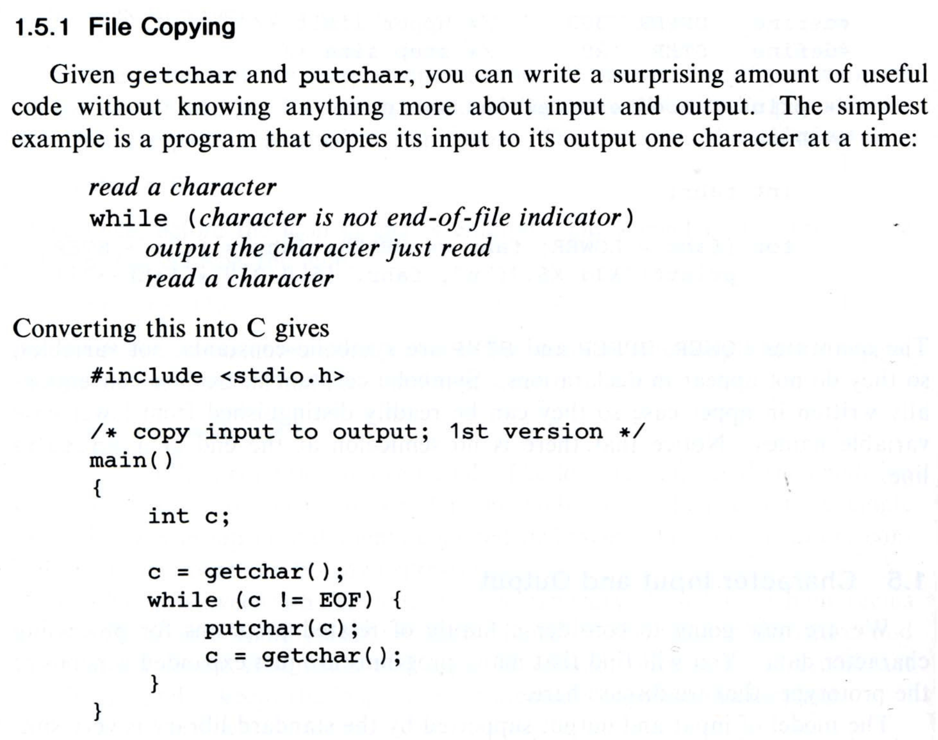
```
#include <stdio.h>

/* print Fahrenheit-Celsius table
    for fahr = 0, 20, ..., 300 */
main()
{
    int fahr, celsius;
    int lower, upper, step;

    lower = 0;      /* lower limit of temperature table */
    upper = 300;    /* upper limit */
    step = 20;      /* step size */

    fahr = lower;
    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

File Copying Example Program



1.5.1 File Copying

Given `getchar` and `putchar`, you can write a surprising amount of useful code without knowing anything more about input and output. The simplest example is a program that copies its input to its output one character at a time:

```
read a character
while (character is not end-of-file indicator)
    output the character just read
    read a character
```

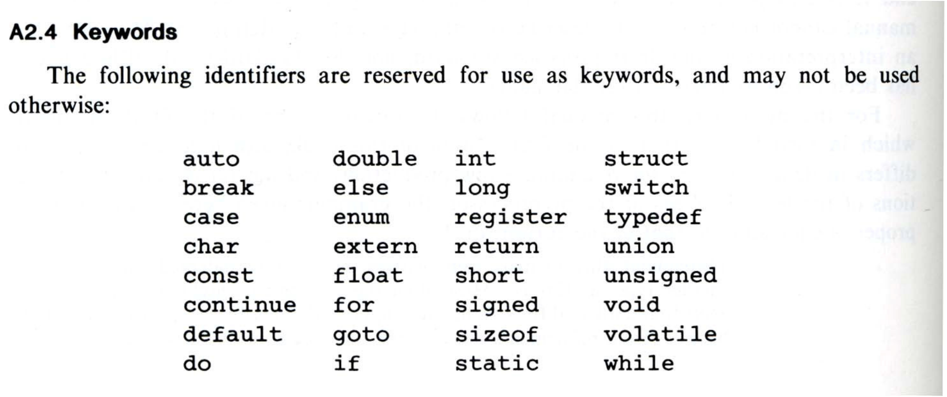
Converting this into C gives

```
#include <stdio.h>

/* copy input to output; 1st version */
main()
{
    int c;

    c = getchar();
    while (c != EOF) {
        putchar(c);
        c = getchar();
    }
}
```

'C' Language Keyword Table

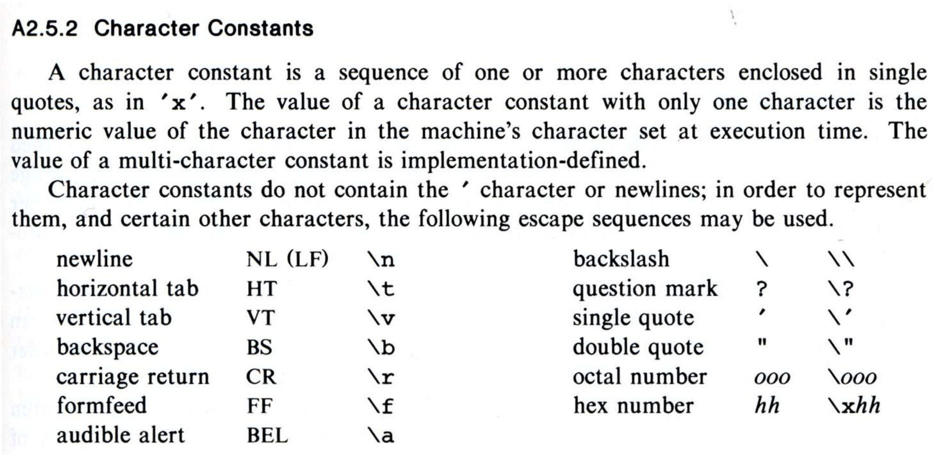


A2.4 Keywords

The following identifiers are reserved for use as keywords, and may not be used otherwise:

auto
break
case
char
const
continue
default
do
double
else
enum
extern
float
for
goto
if
int
long
register
return
short
signed
sizeof
static
struct
switch
typedef
union
unsigned
void
volatile
while

Special Character Representation



A2.5.2 Character Constants

A character constant is a sequence of one or more characters enclosed in single quotes, as in `'x'`. The value of a character constant with only one character is the numeric value of the character in the machine's character set at execution time. The value of a multi-character constant is implementation-defined.

Character constants do not contain the ' character or newlines; in order to represent them, and certain other characters, the following escape sequences may be used.

newline	NL (LF)	`\n`
horizontal tab	HT	`\t`
vertical tab	VT	`\v`
backspace	BS	`\b`
carriage return	CR	`\r`
formfeed	FF	`\f`
audible alert	BEL	`\a`
backslash	\	`\\`
question mark	?	`\?`
single quote	'	`\'`
double quote	"	`\"`
octal number	*ooo*	\ooo
hex number	*hh*	\x*hh*