CSC9V4 Practical 3
Where to begin, where to end?

Introduction

Before moving on to tackle some of the real quircks in C, it is important first to ensure familiarity with the basics. Today's goal: get past the agony of arrays so that they are second nature!

Background

The following program looks for repeat digits in a sequence of digits. Main loop is the while: it takes the last digit of the number and checks if it has already been seen (i.e., true in the boolean array), in which case it breaks (and n > 0 !). Then, it marks the position of that digit in the array as true, anyway. Go through line by line and be sure to understand in detail how the program works.

Compile and run it.

```
#include <stdbool.h>   /* C99 only */
#include <stdio.h>

int main(void)
{
  bool digit_seen[10] = {false};
  int digit;
  long n;

  printf("Enter a number: ");
  scanf("%ld", &n);    /* Not '1', but lower-case L! */

  while (n > 0) {
    digit = n % 10;
    if (digit_seen[digit])   /* This conditional can be */
      break;                 /* re-written without the  */
    digit_seen[digit] = true;/* How might this be done? */
    n /= 10;
  }

  if (n > 0)
    printf("There are repeated digit\n");
  else
    printf("No repeated digit\n");

  return 0;
}
```

1. Does the program above work for all the possible inputs?

Can you find a "numeric string", i.e. a string that is made of digits only, that is accepted by the program, contains repeated digits, but the program will return "No repeated digit" for it.

Generating Useful Output

The default program above only tells us if a digit has been repeated. Instead, rewrite it in a new program `repdigits1.c` so that it shows which digits (if any) were repeated:

```
Enter a number: 939577
The digit 7 is repeated 2 times
The digit 9 is repeated 2 times
```

HINT: the array no longer stores booleans.

2. Can you find a number, properly formatted, that produces a wrong result, i.e. the declaration of repeated digits is not consistent with the input number. Explain why this happens.

"More Power!"

Repeat digits alone tell only part of the story. Save a new version of your source in `repdigit2.c`, and modify the program so that it prints a table showing the number of times each digit appears in the number:

```
Enter a number: 12288

Digit:       0 1 2 3 4 5 6 7 8 9
Occurrences:     2           2
```

Mind the spaces: you have to proper align digits!

The First Step Towards Gamedom

Copy of your source into a new program, `repdigit3.c,` and modify the program so that the user can enter as many numbers as desired for testing and evaluation. The program should terminate when the user enters a number that is less than or equal to `0`.

Check Point.

But if wanting additional fun…

How might the program be modified to use the expression `(int)(sizeof(a)/sizeof(a[0])` or a macro with this value? Make sure to try this out!

Could you declare and initialise a chess board? Try declaring an 8 x 8 char array that includes an initializer to put the starting positions of a chess game on the board (one char per array element):

```
r n b q k b n r
p p p p p p p p
  .   .   .   .
.   .   .   .
  .   .   .   .
.   .   .   .
P P P P P P P P
R N B Q K B N R
```