Background

Some computer architectures store numbers in base 8 representation (**octal**), which means that they are stored in a series of **digits** which can be 0, 1, 2, 3, 4, 5, 6, or 7. A base 10 representation (**decimal**) of a number can be converted into a base 8 representation (octal) of the same number, and vice versa, by following a simple algorithm. Two examples are shown below for a **3-digit** representation.

Convert decimal (base 10) to octal (base 8)

Number = 314 (base 10)

314 / 8 = 39 remainder 2
39 / 8 = 4 remainder 7
4 / 8 = 0 remainder 4

Performed 3 times for 3 digits

→ Number = 472 (base 8)

Convert octal (base 8) to decimal (base 10)

Number = 472 (base 8)

2nd digit, 1st digit, 0th digit

→ Number = 4*8^2 + 7*8^1 + 2*8^0
= 4*64 + 7*8 + 2*1
= 256 + 56 + 2
= 314 (base 10)

Binary (base 8) Representation Summary:

8^2	8^1	8^0
4	7	2

Decimal (base 10) Representation Summary:

10^2	10^1	10^0
3	1	4

The above examples use an **unsigned** 3-digit representation. This means that only nonnegative numbers can be represented in the 3 digits. The smallest number represented in this representation is 000 (base 8), and the largest is 777 (base 8). Based on this, what is the corresponding valid range of decimal (base 10) numbers?

You can read more about octal here: https://en.wikipedia.org/wiki/Octal.

Instructions

- Complete the C25-1 assignment which introduces repetition and selection structures in C.
- Create a new program.
- Prompt the user to enter a decimal (base 10) number which can be represented in 3 digits of octal.
- If the user does not enter a number in the valid range of decimal (base 10) numbers, prompt the user again until a valid number is entered. (You do not need to be able to handle incorrect inputs other than invalid numbers.)
- Convert the entered number into octal (base 8) in 3 digits using the algorithm above.
- Display the converted octal (base 8) 3-digit representation of the number to the screen.
- Convert the new octal (base 8) representation back into a decimal (base 10) representation using the algorithm above.
- Display the converted decimal (base 10) representation of the number to the screen.

- Your code must include at least one **`while()`** or **`for()`** loop, and one **`if()`** structure.
- **Hint #1:** You might store the digits of the octal (base 8) representation in an array, or in separate variables for each digit. If you use an array, the individual elements of the array are accessed by using brackets **`[]`** with the index inside the brackets. You must also declare the array with the number of elements in brackets, such as:

```
int array[size];
```

  Note that **`size`** is not equal to the highest usable index for the array since the index counting starts at 0.
- **Hint #2:** Keep track of the order of the digits in the octal (base 8) representation and in the stored array, and make sure they are printed in the proper order to the screen.
- **Hint #3:** You can check your conversions using this online tool: https://www.binaryhexconverter.com/decimal-to-octal-converter.