Conversion of a binary number (base 2) to its decimal equivalent

The aim of this exercise is twofold :
a) to revise your understanding of the binary numbering system
and
b) to practice manipulation of a string, allowing you to process separate characters

Processing:

The program is to ask for an 8 digit binary number containing only ones and zeroes.
Be sure to input his data as a string, otherwise you will find any leading zeroes ignored.

Validate the input, by checking for valid length, and the existence of only ones and zeros.

Use the construct

```
REPEAT
        Input data
        Validate
UNTIL errflag = 0
```

If the input is incorrect, display an error message indicating the error at the bottom of the screen, before asking for re-entry of the binary number.

Once a valid 8-bit binary string has been entered, convert it to its decimal equivalent (but do not yet display the answer).

Ask the user to enter the number they think is the correct decimal equivalent, and compare it to your answer

If their answer is correct, say so, and ask if they wish to try another example.
If their answer is incorrect, display the correct answer with the full working, as shown below.

Output

```
Wrong - the correct answer is .....
.
1 * (2**8) = .....
0 * (2**7) = .....
0 * (2**6) = .....
.
        .
.
        .
.
        .
.
        .
.
0 * (2**0) = .....

sum         = .....
```

Hints:

Start at the eighth digit (the rightmost digit), and extract one character
Multiply this digit by the correct power of 2 (2**0), and keep a sum.
Repeat this process for each digit, moving along from right to left, and multiply by the correct power of 2, adding into the final sum as you go.

Make sure you design your solution FIRST, so that your logic is clear in your mind before beginning to code.