



Next: [What to hand in](#): Up: [No Title](#) Previous: [Part A - Manipulating](#)

Part B - Adding binary numbers

If we were adding two binary numbers by hand, say 1001 (that's 9 in decimal, or base10 notation) and 1111 (that's 15 in decimal notation), we would write them down, one below the other, and then add them up, column by column, starting from the right and keeping track of what is carried to the next column. For example, to add 1001 and 1111, we start with

$$\begin{array}{r} 1001 \\ + 1111 \\ \hline \end{array}$$

and add up the rightmost column. 1+1 is 10 in binary (2 in decimal) so we write down 0 and carry the 1 to the next column to get

$$\begin{array}{r} 1 \\ 1001 \\ + 1111 \\ \hline 0 \end{array}$$

Now we add the second column from the right. 1+0+1 is 10 in binary (2 in decimal), so we write down the 0 and carry the 1 to the next column to get

$$\begin{array}{r} 1 \\ 1001 \\ + 1111 \\ \hline 00 \end{array}$$

We add the third column with similar results:

$$\begin{array}{r} 1 \\ 1001 \\ + 1111 \\ \hline 000 \end{array}$$

Adding the fourth column, we get 1+1+1 or 11 in binary (3 in decimal), so we write down a 1 and carry a 1 to get

$$\begin{array}{r} 1 \\ 1001 \\ + 1111 \\ \hline 1000 \end{array}$$

And we now sum the (new) fifth column to get a final answer of

$$\begin{array}{r} 1 \\ 1001 \\ + 1111 \\ \hline 11000 \end{array}$$

Note how similar this is to adding two decimal (base 10) numbers. For decimal numbers, we still add column by column starting at the right. The sum of the two digits in the column (plus the carry from the previous column) is between 0 and 19, so we write down a digit between 0 and 9 and carry a 0 or a 1.

Consider using strings of 0's and 1's to represent binary numbers, e.g. ``1001" to represent the binary number 1001 or ``1111" to represent the binary number 1111.

1. Write a Turing program that accepts 2 such ``binary strings" as input and then outputs a string representing their total. To make it easier, make sure that the two input strings are the same length. Here is what the output from a run of this program should look like:

```
Enter 2 binary numbers as strings:    "1001"  "1111"  
The sum of the binary numbers 1001 and 1111 is 11000.
```

Be sure your program works for arbitrary binary strings, not just the ones given as examples.

2. Once you have the program from part 1 working correctly, alter it so that it repeatedly prompts for and accepts two strings and outputs their total. It should stop only when the first string entered is ``quit". Create a test file of numbers so that you don't have to type them in at the keyboard each time you run your program. This also allows you to hand in your test file and the results with your program.
3. Once you have the program from part 2 working correctly, alter it so that it works even if the two input strings have different lengths. Here is what the output from a run of this program should look like:

```
Enter 2 binary numbers as strings:    "10"    "11111"  
The sum of the binary numbers 10 and 11111 is 100001.
```



Next: [What to hand in:](#) **Up:** [No Title](#) **Previous:** [Part A - Manipulating](#)

Michelle Wahl Craig
Tue Feb 20 14:55:15 EST 1996