Translating Binary to Text

Introduction:
We’ve all seen binary code. We’ve come to think of them as a bunch of ones and zeroes in long strings…
010010101010101001101011

But these ones and zeroes can also represent decimal numbers. First off, I will show you how to read these numbers as the decimal numbers we’re used to in our daily life. Then, I will show you how to use those numbers and your keypad to translate them into text. Note that your computer doesn’t use the decimal system, so technically, when it converts binary to text, it doesn’t go through the process I will show you. This is just a divertive way of explaining you how the binary system works.

The Binary System:
Here’s a simple example of binary:
10101

Let’s think of the example above as empty slots:
_ _ _ _ _

First off, you read binary from right-to-left. It’s just the way it’s designed. The first slot from the right represents a value of one, the second from the right a value of two, the third from the right a value of four, the fourth from the right a value of eight, the fifth from the right a value of sixteen, and the cycle continues by multiples of 2. This will never change.

By putting a 1 or a 0 in those slots you are either saying you want to corresponding value that’s attached to that slot or you don’t. A 1 means yes, and a 0 means no. For example, putting a zero in the first slot from the right, but a 1 in the second slot from the right means you want a two, but not a one:
_ _ _ 1 0

As such, the number above equals to a decimal value of two.

As an example, let’s say you want to represent eight in binary form. Well, thinking about the slots, you want the first slot to be 0 because you don’t want a one, you want the second slot to also be 0 because you don’t want a two, you want the third slot to also to be 0 because you don’t want a four, but you want the fifth slot to be 1 because you want a value of eight. As such, eight in binary form is:
1 0 0 0 (or simply 1000 without those underlines)

Now it is important to note that the amount of zeroes that precede the first value of one from the left is unimportant. So for example:
1 0 0 0 is the same as 0 0 0 1 0 0 0 (1000 = 000100)

To get it cleared up, here’s another example:
0 1 is the same as 1

Exercises: What do the following equal in decimal terms?
a) 100
b] 000100
c) 100000
d) 0010

Answers:
a) 4
b] 4
c) 32
d) 2

If you got the answers above right, then you pretty much understand the basics of binary.
Let’s now understand how to get the corresponding decimal values to the numbers which are not multiples of 2.

To get the total value of a binary number, add the values corresponding to each slot. So, for example, three in binary would be:
11

The above corresponds to three because if you add the total values of all the slots, that is to say a one from the slot to the right, and a two from the second slot to the right, then it equals three.

As another example, let’s say you want to represent 5 in binary terms. Then you would need a value of one to be added to a value of four, and you would not want a value of two:
101 [Reading from the right: 1(one) + 0(two) + 1(four) = five]

Here’s an additional example:
001011 [Reading from the right: 1(one) + 1(two) + 0(four) + 1(eight) + 0(sixteen) + 0(thirty-two) = eleven)

Exercises: What do the following equal in decimal terms?
a) 11011
b] 110
c) 010101
d) 10110

Answers:
a) 27
b] 6
c) 21
d) 22

If you got the above questions correct [without cheating], then you essentially understand the binary system.
Understanding the binary system was the hard part. What follows is pretty easy.

3. Converting Binary to ASCII (Text)
ASCII is essentially the letters, numbers and symbols that are stored in our computers through the use of fonts. When the keyboard relays the buttons you pressed, it sends in a code which is then converted to the

ASCII equivalent of “k” or “5” or whatever key you pressed.

Here’s an example of a message “hidden” in binary text:
0100100001100101011011000110110001101111

Now there are only so many letters, numbers and symbols stored for ASCII. Having sets of 8 digits for their binary equivalent is more than enough to represent all of these letters and the like. As such, all strings that represent text like in the above are separated into bits of 8 for simplicity:
01001000 01100101 01101100 01101100 01101111

Okay, so our example message was separated into 8 digit strings. The decimal value for each of these strings in the example was calculated for you.
01001000 = 72
01100101 = 101
01101100 = 108
01101100 = 108
01101111 = 111

The result was 72,101,108,108,111. Now, there is something called the ASCII table. It essentially corresponds to the binary numbers from yore to the equivalent letters/symbols/numbers. But since we found the decimal values of these binary strings, we can use a major shortcut.

By pressing ALT + [The Number], you will get the ASCII equivalent of that number. For example, by pressing the ALT key and at then (while keeping it down) the numbers 72 in any text editor, you will get the corresponding “H” to show up.

Let’s do so for the entire example message:
72 = H
101 = e
108 = l
108 = l
111 = o

So the entire “hidden” message translates to “Hello”.

Exercise: Decode the following message
01000011011011110110111001100111011100100110000101110100011101010110110001100001
011101000
1101001011011110110111001110011 00100001

Hint: The first step on your way to decoding the message (separated into bytes for you)
01000011 01101111 01101110 01100111 01110010 01100001 01110100 01110101 01101100 01100001 01110100 01101001
01101111 01101110 01110011 00100001