

## Converting between any two number systems

We now know how to convert between any number system to decimal.

We now know how to convert between decimal and any other number system.

Does this mean that we can convert between any two non-decimal number systems?

Yes! We can simply convert from the starting number system to decimal and then from decimal to the new number system like so:

1. Convert from base A number system to base 10
2. Convert from base 10 number system to base B
3. Profit

We are not going to make this subsystem that easy though! Subsystems 2 and 3 only used number systems with a base less than or equal to 10. We are now going to remove that restriction and convert between number systems anywhere between base 2 and base 16.

How does this change our answer? The functions `ConvertToBase10()` and `ConvertFromBase10()` that we defined earlier both utilise mathematical operations that were designed for base 10 number systems. For example, the `parseInt()` function that converts a character to an integer only works with the decimal digits 0-9. If we ask it to convert the number 10 in hexadecimal (A) it will get confused as it thinks that A is letter. We will therefore need to build our own version of this function.

There are multiple ways that we could solve this problem, but the most straightforward would probably be a direct mapping between decimal and other number systems, similar to what we used in subsystem 1.

Because we only have to worry about number systems between base 2 and 16 which all use consistent symbols, if we build this helper function to deal with conversions between hexadecimal and decimal then we can also reuse it for all of our other number systems (as the other number systems are all [subsets](#) of hexadecimal).

This is an example of the function `parseCharToInt()`, a replacement for the function `parseInt()`. You will need to copy this code and add it to your program in subsystem 4 to give your other functions access to it, and then replace

`parseCharToInt(***YOURCODEHERE***)` with `parseInt(***YOURCODEHERE***)` in the function `ConvertToBase10()`.

```
// The ConvertToBase10 Function needs to be modified slightly to deal
// with numbers A-F using this helper function
let parseCharToInt = (number) => {
  if (number == '0') { return 0; }
  else if (number == '1') { return 1; }
  else if (number == '2') { return 2; }
  else if (number == '3') { return 3; }
  else if (number == '4') { return 4; }
  else if (number == '5') { return 5; }
  else if (number == '6') { return 6; }
  else if (number == '7') { return 7; }
  else if (number == '8') { return 8; }
  else if (number == '9') { return 9; }
  else if (number == 'A') { return 10; }
  else if (number == 'B') { return 11; }
  else if (number == 'C') { return 12; }
  else if (number == 'D') { return 13; }
  else if (number == 'E') { return 14; }
  else if (number == 'F') { return 15; }
  else { return 0; }
}
```

In addition to needing to convert from any number system to base 10, you will also need to convert from base 10 to any other number system by creating a function `parseIntToChar()`. This function should work the same way as `parseCharToInt()` but in reverse.

It is up to you to create this function and replace `remainder.toString()` with `parseIntToChar(remainder)` in your function `ConvertFromBase10()`.

Return to your learning journal and complete the research questions for the fourth and final subsystem.