Using Recursion to Convert Number to Other Number Bases
99_{10} \text{ is also } 1100011_2

Problem: Convert a decimal (base 10) number into other bases

<table>
<thead>
<tr>
<th>Message</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>convert(99, 2)</td>
<td>&quot;1100011&quot;</td>
</tr>
<tr>
<td>convert(99, 3)</td>
<td>&quot;10200&quot;</td>
</tr>
<tr>
<td>convert(99, 4)</td>
<td>&quot;1203&quot;</td>
</tr>
<tr>
<td>convert(99, 5)</td>
<td>&quot;344&quot;</td>
</tr>
<tr>
<td>convert(99, 6)</td>
<td>&quot;243&quot;</td>
</tr>
<tr>
<td>convert(99, 7)</td>
<td>&quot;201&quot;</td>
</tr>
<tr>
<td>convert(99, 8)</td>
<td>&quot;143&quot;</td>
</tr>
<tr>
<td>convert(99, 9)</td>
<td>&quot;120&quot;</td>
</tr>
</tbody>
</table>
Multiply Digits by Powers of Base
10, 8, 2, or whatever

Decimal numbers: multiply digits by powers of 10

\[9507_{10} = 9 \times 10^3 + 5 \times 10^2 + 0 \times 10^1 + 7 \times 10^0\]

Octal numbers  powers of 8

\[1567_8 = 1 \times 8^3 + 5 \times 8^2 + 6 \times 8^1 + 7 \times 8^0\]
\[= 512 + 320 + 48 + 7 = 887_{10}\]

Binary numbers  powers of 2

\[1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\]
\[= 8 + 4 + 0 + 1 = 13_{10}\]
Converting base 10 to base 2

1) divide number (5) by new base(2), write remainder (1)
2) divide quotient (2), write new remainder (0) to left
3) divide quotient (1), write new remainder (1) to left

\[
\begin{align*}
2 \overline{)5} & \quad \text{Remainder} = 1 \\
& \quad 1 \\
2 \overline{)2} & \quad \text{Remainder} = 0 \\
& \quad 0 \\
2 \overline{)1} & \quad \text{Remainder} = 1 \\
& \quad 1
\end{align*}
\]

Stop when the quotient is 0

\[5_{10} = 101_2\]

Place remainders in reverse order
Converting base 10 to base 8

1) divide number by new base (8), write remainder (1)
2) divide quotient (2), write new remainder (0) to left
3) divide quotient (1), write new remainder (1) to left

\[
\begin{align*}
8 & \mid 99 & \quad \text{Remainder} = 3 \\
8 & \mid 12 & \quad \text{Remainder} = 4 \\
8 & \mid 1 & \quad \text{Remainder} = 1
\end{align*}
\]

Stop when the quotient is 0 \[99_{10} = 143_8\]

Place remainders in reverse order
Possible Solutions

We could either

1. store remainders in an array and reverse it, or
2. write out the remainders in reverse order, or
3. postpone the output until we get quotient = 0
4. store result as a String (like a recursion assignment)

Iterative Algorithm

while the decimal number > 0  {
  Divide the decimal number by the new base
  Set decimal number = decimal number divided by the base
  Store the remainder to the left of any preceding remainders
}
Recursive algorithm

Base case
if decimal number being converted = 0
  • do nothing (or return "")

Recursive case
if decimal number being converted > 0
  • solve a simpler version of the problem by using the quotient as the argument to the next call
  • store the current remainder (number % base) in the correct place
public String convert(int num, int base) {
    if (num == 0)
        return "";
    else
        return convert(num / base, base) + (num % base);
}
Hexadecimal

Convert this algorithm to handle all base up through hexadecimal (base 16)

10 = A
11 = B
12 = C
13 = D
14 = E
15 = F