

CS 7A - Spring 2020 - Engineering Repetends. Due 2/26/20

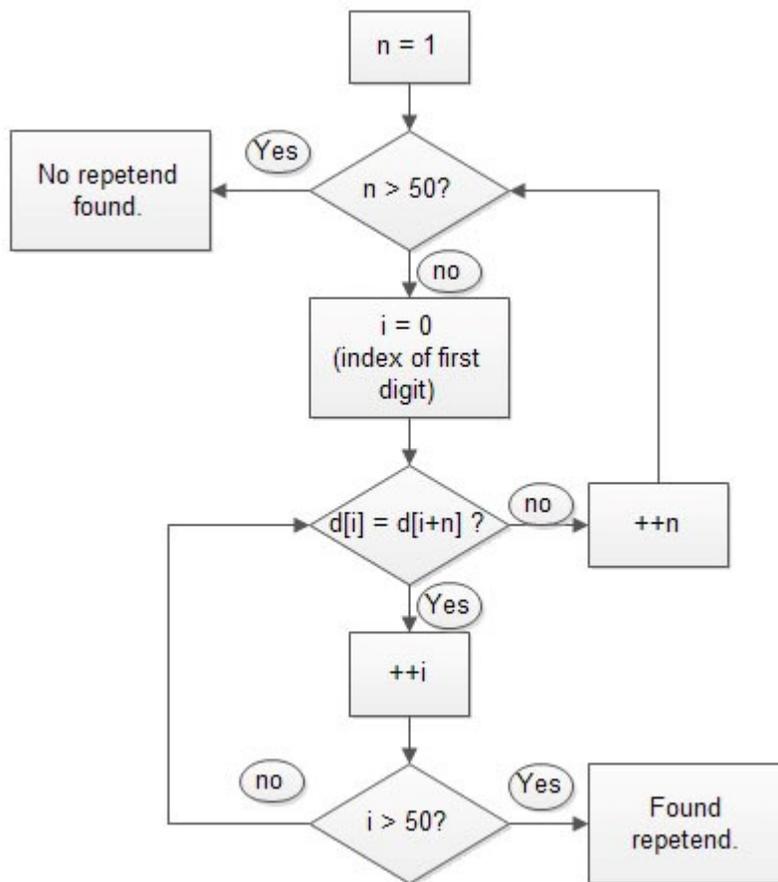
When you perform long division with integers to compute the decimal form of a rational number, there are two types of results: terminating decimals and repeating decimals. For repeating decimals the digits that repeat are called the **repetend**. For example, $\frac{1}{3} = 0.\overline{3}$ has a repetend of 3 and $\frac{12}{13} = 0.\overline{923076}$ has a repetend of 923076. Note that the length of the repetend of $\frac{p}{q}$ must be less than $q - 1$ and is often equal to that. $\frac{1}{7} = 0.\overline{142857}$, for instance.

Your goal in this project is to find the repetend for any divisor up to `MAX_INT`. Here is some started code to get you going:

```
1 // Code to search for a repetend in the decimal form of a rational number.
2 // precondition: a quotient and a divisor, both positive integers.
3 // postcondition: a vector containing the sequence of decimal digits in
4 // the quotient, the length of the repetend and a sequence of repetend digits.
5 /*
6 Enter the dividend and divisor of the rational number whose repetend we seek:
7 1 17
8
9 Here are the digits of the quotient computed:
10 005882352941176470588235294117647058823529411764705882
11 352941176470588235294117647058823529411764705882352941
12 1764705882352941176470588235294117647058823529
13 4117647058823529411764705882352941176470588235
14 The repetend is of length 16
15 The digits of the repetend are
16 4117647058823529, though not necessarily in that order. */
17
18 #include <iostream>
19 using namespace std;
20
21 int repetend(vector<short>&);
22
23 int main() {
24     int dividend, divisor, quotient, remainder;
25     vector<short> decimalDigs, rep;
26     cout << "\nEnter the dividend and divisor of the rational number";
27         << "whose repetend we seek:\n";
28     cin >> dividend >> divisor;
29     for(int i = 0; i < 200; ++i) {
30         decimalDigs.push_back(dividend/divisor);
31         remainder = dividend % divisor;
32         dividend = 10 * remainder;
33     }
34     cout << "\nHere are the digits of the quotient computed: ";
35     for(int i = 0; i < decimalDigs.size(); ++i) cout << decimalDigs[i];
36     cout << "The repetend is of length " << repetend(rep)
37         << "\nThe digits of the repetend are\n";
38     //Write a loop to print out the repetend here
39     return 0;
40 }
41 // define the repetend() function here
```

Notice that the number of decimals computed here is 200, but you could easily make that much larger! More properly, it can be made a global constant. Add code where you see the comment at the end, “//define the repetend()function here” to do just that (or rewrite from scratch).

A flow chart for the algorithm is shown at right. In my implementation of this I used nested while loops and a “break,” but you may use other control structures, as you like.



Questions:

1. Email your code to me with the title “(initials here)_repetend.cpp” and, in the body of your email, explain how you used the code to find longest repetend with a divisor less than 10000.



'Ditto,' said Tweedledum. 'Ditto, ditto!' cried Tweedledee