CS7A – Spring ’13 - Lab 9

Examine the code below for transforming a number from binary to decimal and algorithms 1.2 and 1.4 from the text. From this, develop code for converting from ternary to decimal and back again.

```
// G. Hagopian
// Convert binary to decimal

#include <iostream>
using namespace std;

int decimal(int b);
// Returns the decimal numeral whose value equals that
// represented in the binary form b.
// PRECONDITION: each digit of b is a bit: 0 or 1
// EXAMPLE: decimal(10001011) returns 139

int main()
{
	int b;
	cout << "\nEnter a number in binary form: ";
	cin >> b;
	cout << "\nThe decimal value is " << decimal(b) << endl;
}
/*
1. Set x = 0.
2. For each b_j = 1, add 2^j to x. (Note that j is the number of bits to the right of b_j).
3. Return x.*/

int decimal(int b)
{
	int x = 0, addend = 1;

	while(b>0)
	{
		if(b%10 == 1)
			x += addend;
		addend *= 2;
		b /= 10;
	}
	return x;
}
```

1. Adapt the code above to change from a ternary (base 3) number to a decimal
2. Modify Algorithm 1.4 to convert decimal numerals to ternary:

```
int ternary(int n);
// Returns the ternary numeral that represents n.
// PRECONDITION: n >= 0
// POSTCONDITION: each digit of the integer returned is a base 3 numeral
// EXAMPLE: ternary(139) returns 12011
```

Algorithm 1.4 Conversion from Decimal to Binary

To convert the integer ***x*** into its equivalent binary numeral:

1. Assert $x > 0$.
2. Set $k = 0$.
3. If ***x*** is odd, set $b_k = 1$; otherwise set $b_k = 0$.
4. Subtract b_k from ***x.***