

Write a Microsoft Visual Studio .NET project that combines a C++ main function with another function named `properFactors`. The C++ main function (see sample below) should search for and display all amicable pairs of integers where at least one of the integers is between 1 and 1,000,000, inclusive. The `properFactors` function should be written in a separately compiled file, `properFactors.asm`

Definition: A proper factor of a positive integer, p , is any positive integer less than p that evenly divides into p . For example, the proper factors of 12 are 1, 2, 3, 4 and 6. Notice, that p can never be a proper factor of itself.

A pair of integers, p and q , are said to be **amicable** if the sum of the proper factors of p equals q and the sum of the proper factors of q equals p . For example, the integers 220 and 284 form an amicable pair because the proper factors of 220 add up to 284 ($1+2+4+5+10+11+20+22+44+55+110=284$) and the proper factors of 284 add up to 220 ($1+2+4+72+142=220$).

The function, `properFactors`, you are assigned to write should accept a positive integer, n , and then (1) find all proper factors of n , storing them in an integer array passed by reference in the parameter list, (2) store the number of proper factors in an integer parameter passed by reference, and (3) return the sum of all of the proper factors as the function's return value.

The first parameter of the `properFactors` function should be the positive integer, n , passed by value. The second parameter should be the integer array, passed by reference, which will be assigned each of the proper factors of n (in no particular order); and the third parameter is an integer variable, passed by reference, which will be assigned the number of proper factors. Thus, the function prototype should be similar to: `int properFactors (int, int[], int&);` **A flowchart that describes this function has been provided.**

Assignment part 1: Write the `properFactors` function using C++. Run the program four times, each time recording the number of seconds necessary to find all amicable pairs in the range 1 to 1,000,000. Calculate the average time.

Assignment part 2: Convert your `properFactors` function to MASM. Again run the program four times, each time recording the number of seconds necessary to find all amicable pairs. What is the average time using your MASM function?

Sample main function:

```
#include <iostream>
#include <time.h>
using namespace std;

#define MAXIMUM 1000000
int Factors[MAXIMUM];
extern "C" int properFactors (int, int[], int&);

int main () {
    int q;
    int numFactors;
    int PairCount = 0;

    clock_t StartClock = clock();
    for (int p=1; p<=MAXIMUM; p++) {
        q = properFactors(p, Factors, numFactors);
        if (q > p) {
            if ( properFactors(q, Factors, numFactors) == p ) {
                cout << "Amicable Pair: " << p << "," << q ;
                cout << " (" << q << " has " << numFactors << " factors: ";
                cout << Factors[0];
                for (int i=1; i<numFactors; i++) { cout << "," << Factors[i]; }
                cout << ")" << endl;
                PairCount++;
            }
        }
    }

    double TotalTime = (double) (clock() - StartClock) / CLOCKS_PER_SEC;
    cout << "Number of amicable pairs found = " << PairCount << endl;
    cout << "Time = " << TotalTime << " seconds" << endl;
}
```

```
int ProperFactors (int n, int Factors[], int &numFactors)
```

