messaggi.cpp

```
/* FILE: messaggiRic.cpp   last change: 7-Nov-2012   last author: Romeo Rizzi
 * a recursive (exponential) solver for problem "messaggi"
 * adapted, cleaned, and corrected from Luca Foschini - ioi2005 - Fase nazionale
 */

//#define NDEBUG   // NDEBUG definita nella versione che consegno
#include <cassert>
#ifndef NDEBUG
#  include <iostream>  // uso di cin e cout non consentito in versione finale
#endif
#include <fstream>
#include <set>
#include <string>

using namespace std;

int cm_len=4;
string s; int n; // the input string and its length
set<string> cw;

int count(int p) {
  // counts the number of interpretations for the suffix of s starting with the character in positio
n 0. (For p=n is the empty string, for p=0 is the whole s).
  if ( p == n )  return 1;
  int sum = 0; string sub = "";
  for(int len = 1; len <= cm_len; len++) {
     if( p+len > n ) break;
     sub += s[p+len-1];
     if( cw.find(sub) != cw.end() )  sum = (sum + count(p+len)) % 10; // the .mod. was missing in Lu
ca's version. With this, the outputs may differ.
  }
  return sum;
}

int main() {
  // Versione sol Foschini: A = 0, B = 001, C = 010, D = 0100, E = 0010

  // Vers. att.: A = 0, B = 00, C = 001, D = 010, E = 0010 , F = 0100, G = 0110
  cw.insert("0"); cw.insert("00"); cw.insert("001"); cw.insert("010"); cw.insert("0010"); cw.insert("0
100"); cw.insert("0110");

  ifstream fin("input.txt"); assert( fin );
  fin >> n;
  s = ""; int digit;
  for(int i = 1; i <= n; i++) { fin >> digit; s += char(digit+'0'); }
  fin.close();

  ofstream fout("output.txt"); assert( fout );
  fout << count(0) << endl;
  fout.close();
  return 0;
}
```