

## messaggi.c

```

/* messaggi *
   solutore
   Roberto Grossi 2005-02-26
   modificato 8-11-2012 da Romeo Rizzi per adattarlo all'esame 9/11/2012
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>

// #define STDSTREAM
// #define DEBUG

#define MaxN 100000
#define MaxZ 8
#define MaxB 8

char S[MaxN]; // Sequenza binaria in ingresso
char Z[MaxZ][MaxB]; // Codifica dei simboli dell'alfabeto: A->Z[1], B->Z[2],...
int LZ[MaxZ]; // Lunghezza in bit delle codifiche

int N, NZ; // lunghezza sequenza e dimensione alfabeto

void init(){
    // Versione sol Grossi: 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

    NZ = 7;
    LZ[0] = 1; Z[0][0] = 0;
    LZ[1] = 2; Z[1][0] = 0; Z[1][1] = 0;
    LZ[2] = 3; Z[2][0] = 0; Z[2][1] = 0; Z[2][2] = 1;
    LZ[3] = 3; Z[3][0] = 0; Z[3][1] = 1; Z[3][2] = 0;
    LZ[4] = 4; Z[4][0] = 0; Z[4][1] = 1; Z[4][2] = 0; Z[4][3] = 0;
    LZ[5] = 4; Z[5][0] = 0; Z[5][1] = 0; Z[5][2] = 1; Z[5][3] = 0;
    LZ[6] = 4; Z[6][0] = 0; Z[6][1] = 1; Z[6][2] = 1; Z[6][3] = 0;
}

int main(int argc, char *argv[]) {
    int i, j, r, s, xi;
    int C[MaxN+1]; // Tabella di programmazione dinamica per # messaggi decodificati
    FILE *in, *out;

#ifdef STDSTREAM
    in = stdin;
    out = stdout;
#else
    in = fopen("input.txt", "r");
    out = fopen("output.txt", "w");
#endif

    if ( in == NULL || out == NULL )
        return 1;

    if ( fscanf(in, "%d", &N) != 1 )
        return 1;

    assert( 1 < N && N < MaxN );

    /* Lettura input */

    for ( i = 0 ; i < N ; i++ ) {
        if ( fscanf(in, "%d", &x) != 1 )
            return 1;
        S[i] = x;
        assert ( S[i] == 0 || S[i] == 1 );
    }
}

```

## messaggi.c

```

#ifdef DEBUG
    printf("%d\n", N);
    for ( i = 0 ; i < N-1 ; i++ ){
        printf("%d ", S[i]);
    }
    printf("%d\n", S[N-1]);
#endif

/* Risoluzione */

init();

// C[i] = #messaggi codificati da S[0 .. i-1]
C[0]=1;

for (i=1; i <= N; i++){
    C[i] = 0;
    for (j=0; j < NZ; j++){
        if ( i >= LZ[j] ){ // Il codice del carattere non e' piu' lungo del prefisso di S
            for ( r = LZ[j]-1, s = i-1; r >= 0; r--, s-- )
                if ( Z[j][r] != S[s] ) break;
            if ( r < 0 ) // Successo
                C[i] = (C[i] + C[i - LZ[j]]) % 10; // in versione Grossi mancava mod;
        }
    }
}

#ifdef DEBUG
    printf("%d\n", N);
    // Inutile stampare C[0]
    for ( i = 1 ; i < N ; i++ ){
        printf("%d ", C[i]);
    }
    printf("%d\n", C[N]);
#endif

fprintf(out, "%d\n", C[N]); // in versione Grossi: printf("%d\n", C[N]);

return 0;
}

```