

09 Ottobre 2023

Programmazione B
Ingegneria e Scienze Informatiche - Cesena
A.A. 2023-2024

Elaborato 4

Data di sottomissione: entro le 23.59 del 15 Ottobre 2023.

Formato di sottomissione: un file compresso con nome Elaborato4.zip contenente un unico file sorgente con nome `prime.c`.

Codeboard: <https://codeboard.io/projects/131845/>

Specifiche:

- Sviluppare una libreria, `prime.c`, che implementi le funzioni dichiarate nei seguenti prototipi (da definire in un file header `prime.h`):

```
1  /* Ritorna 1 se n e' primo, 0 altrimenti. */
2  unsigned short int is_prime(unsigned short int n);
3
4  /*
5   * Ritorna l'n-esimo primo, contando a partire da 0.
6   *
7   * Se il numero e' troppo grande per essere rappresentato
8   * con un unsigned short int, ritorna 0.
9   */
10 unsigned short int nth_prime(unsigned short int n);
11
12 /* Ritorna la successione di numeri primi.
13  * La prima chiamata ritorna 2, la seconda 3, ecc.
14  *
15  * Se il parametro reset e' diverso da 0, allora la
16  * successione viene resettata e la funzione ritorna 2.
17  * Diversamente, la funzione ritorna il primo successivo
18  * a quello ritornato alla chiamata precedente.
19  *
20  * Se il primo successivo e' troppo grande per essere
21  * rappresentato con un unsigned short int, la funzione
22  * ritorna 0 e la seccessione viene resettata.
23  */
24 unsigned short int succ_prime(int reset);
25
26 /* Ritorna 1 se m e n sono coprimi, 0 altrimenti. */
27 unsigned short int co_prime(unsigned short int m,
28                             unsigned short int n);
```

- Le definizioni di funzioni devono essere commentate nei punti critici: deve essere chiaro dai commenti l'algoritmo utilizzato per l'implementazione.
- Le implementazioni devono rispettare le seguenti specifiche:
 - La funzione `is_prime()` esegue un test di primalità sul numero passato in input: ritorna 1 se il numero è primo, 0 altrimenti.
 - La funzione `nth_prime()` ritorna l'n-esimo primo. Assumiamo che si inizi a contare da 0. Quindi, `nth_prime(0)` ritorna 2, `nth_prime(3)` ritorna 7, ecc. Se l'n-esimo primo risulta essere troppo grande per poter essere rappresentato con un `unsigned short int` (rischiando quindi di causare overflow), la funzione ritorna il valore 0.
 - La funzione `succ_prime()` ritorna in successione i numeri primi. Alla prima invocazione restituisce il numero 2, alla seconda invocazione il numero 3, ecc. Può essere resettata, passando come parametro un valore diverso da 0. In tal caso, la funzione ritorna 2 (il primo primo). Come per la funzione precedente, se il primo successivo risulta essere troppo grande per poter essere rappresentato con un `unsigned short int`, la funzione ritorna il valore 0 e viene resettata (alla chiamata successiva ritornerà 2).
 - La funzione `co_prime()` ritorna 1 se i due interi passati in input sono coprimi, 0 altrimenti.

Vincoli:

- Le implementazioni devono aderire perfettamente ai prototipi e alle specifiche fornite.
- E' possibile sviluppare altre funzioni che facilitino l'implementazione delle quattro indicate. Tali funzioni non devono essere visibili all'esterno della libreria `prime.c`: la loro visibilità deve essere limitata a `prime.c` (`static`).
- Allo stesso modo, è possibile utilizzare variabili globali in `prime.c` unicamente se dichiarate `static`.

APPENDICE

Definizioni delle proprietà.

- **Numero primo.** Un intero positivo $x > 1$ è un numero primo se è divisibile unicamente per 1 e per se stesso. Esempi:
 1. 2 è un numero primo poiché non ha altri divisori oltre a 1 e se stesso.
 2. 3 è un numero primo poiché non ha altri divisori oltre a 1 e se stesso.
 3. 4 non è un numero primo poiché è divisibile per 2.
 4. 10 non è un numero primo poiché è divisibile per 2 e 5.
- **Numeri coprimi.** Due interi positivi, $x > 1$ e $y > 1$ sono coprimi se hanno unicamente 1 come divisore in comune. Esempi:
 1. 2 e 3 sono coprimi poiché hanno come unico divisore in comune 1
 2. 4 e 9 sono coprimi poiché hanno come unico divisore in comune 1
 3. 4 e 8 non sono coprimi poiché hanno 2 e 4 come divisori in comune.