

10 Ottobre 2016

Programmazione M-Z
Ingegneria e Scienze Informatiche - Cesena
A.A. 2016-2017

Elaborato 3

Data di sottomissione: entro la mezzanotte del 16 Ottobre 2016.

Formato di sottomissione: un file compresso con nome `Elaborato3.zip`, contenente un unico file sorgente con nome `Elaborato3.c`

Specifiche:

- Scrivere un programma C che, dato in input un intero positivo x , determini se:
 - x è un numero primo,
 - x è la somma di due quadrati,
 - x è un numero triangolare.
- L'intero senza segno x (i.e., di tipo `unsigned int`) deve essere letto da tastiera, tramite l'utilizzo dell'istruzione

```
scanf ("%u",&x);
```

- Dato in input un intero senza segno x , il programma deve stampare sul terminale una frase che indichi se tale numero possiede o meno le tre proprietà indicate sopra. Esempi:

1. Se il numero inserito è 2, il programma deve stampare

```
Integer 2 is prime  
Integer 2 is the sum of two squares  
Integer 2 is not triangular
```

2. Se il numero inserito è 3, il programma deve stampare

```
Integer 3 is prime  
Integer 3 is not the sum of two squares  
Integer 3 is triangular
```

3. Se il numero inserito è 4, il programma deve stampare

```
Integer 4 is not prime  
Integer 4 is not the sum of two squares  
Integer 4 is not triangular
```

4. Se il numero inserito è 45, il programma deve stampare

```
Integer 10 is not prime  
Integer 10 is the sum of two squares  
Integer 10 is triangular
```

Vincoli:

- Per il calcolo delle proprietà, non è possibile utilizzare le istruzioni di salto `break`, `continue`, `goto`.
- Per il calcolo delle proprietà è richiesto l'utilizzo di strutture di controllo iterative distinte, a scelta. Per essere precisi, non è possibile utilizzare unicamente il costrutto `for` per determinare le tre proprietà richieste. Lo stesso discorso vale per i costrutti `while` e `do-while`. Ad esempio, se si decide di utilizzare il costrutto `for` per determinare la primalità dell'intero, non sarà possibile utilizzare il costrutto `for` per determinare se l'intero è somma di due quadrati.

Suggerimenti:

- Una delle tre proprietà può essere verificata con una singola istruzione.
- Non è strettamente necessario utilizzare cicli annidati.
- E' preferibile gestire a parte i due casi *speciali* 0 e 1 (i.e. stampare direttamente le proprietà dei due numeri).

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.
- **Somma di quadrati.** Un intero x è la somma di due quadrati se esistono due interi $y > 0$ e $z > 0$, tali che

$$x = y^2 + z^2.$$

Esempi:

1. $2 = 1^2 + 1^2$ è la somma di due quadrati.
 2. 3 non è la somma di due quadrati poiché $3 < 2^2$ e tutte le combinazioni per valori minori di 2 sono: $1^2 + 1^2 \neq 3$.
 3. 4 non è la somma di due quadrati poiché $4 < 3^2$ e tutte le combinazioni per valori minori di 3 sono: $1^2 + 1^2 \neq 1^2 + 2^2 \neq 2^2 + 2^2 \neq 4$.
 4. $10 = 1^2 + 3^2$ è la somma di due quadrati.
- **Numero triangolare.** Un intero x è un numero triangolare se esiste un intero positivo n tale che

$$x = \sum_{i=1}^n i$$

oppure, equivalentemente,

$$x = \frac{n(n+1)}{2}$$

Esempi:

1. 2 non è un numero triangolare poiché $1 < \mathbf{2} < 1 + 2$.
2. $3 = 1 + 2$ è un numero triangolare.
3. 4 non è un numero triangolare poiché $1 + 2 < \mathbf{4} < 1 + 2 + 3$.
4. $10 = 1 + 2 + 3 + 4$ è un numero triangolare.