

Matematica con Python¹

Mauro Saita

e-mail: maurosaita@tiscalinet.it

LEZIONE N.3



Lunghi elenchi di numeri primi.

Indice

1	Funzioni	2
2	Esercizi	3
3	Numeri primi e teorema fondamentale dell'aritmetica	3
4	Test di primalità	4

¹Nome File: python_lezione_03.tex

1 Funzioni

Una funzione è un programma a cui è stato dato un nome in modo che possa essere chiamato, cioè eseguito, tramite quel nome. La sintassi per definire una funzione è

```
def nome_funzione(parametri):
    <istruzioni>
```

Per esempio, la seguente funzione, denominata 'divide', stabilisce se b è un divisore di a . L'output della funzione è prodotto tramite il costrutto *return*; in questo caso, se b divide a la funzione restituisce il valore 'True' altrimenti 'False'.

```
def divide(b, a):
    if a % b == 0:
        return True
    else:
        return False
```

Le funzioni sono sottoprogrammi che possono essere chiamati tutte le volte che lo si ritiene necessario. Il seguente programma acquisisce due interi da tastiera, controlla se il primo è un divisore del secondo e, in caso affermativo stampa: ' b è un divisore di a ' altrimenti stampa: ' b è NON è un divisore di a '.

```
def divide(b, a):
    if a % b == 0:
        return True
    else:
        return False

# MAIN DEL PROGRAMMA
b = input('Digitare un intero positivo: b = ')
a = input('Digitare un altro intero positivo: a= ')
b = float(b)
a = float(a)
if divide(b, a)==True:
    print(int(b), 'è un divisore di ', int(a))
else:
    print(int(b), ' NON è un divisore di ', int(a))
```

La funzione denominata hms trasforma un tempo espresso in secondi in un tempo espresso in ore, minuti, secondi.

```
def hms(nsec):
    hh = nsec/3600
```

```
nsec = nsec % 3600
mm = nsec/60
ss = nsec % 60
print hh, mm, ss
```

2 Esercizi

Esercizio 2.1.

- (a) Scrivere la funzione 'max2' che restituisce il massimo tra 2 numeri.
- (b) Scrivere il programma che, usando la funzione 'max2' legge 2 numeri e stampa il maggiore.

Esercizio 2.2. Scrivere le funzioni 'min3' e 'max3' che restituiscono rispettivamente il minimo e il massimo tra 3 numeri

Esercizio 2.3.

- (a) Scrivere la funzione 'ordina2' che restituisce due numeri in ordine crescente.
- (b) Scrivere la funzione 'ordina3' che restituisce tre numeri in ordine crescente.

Esercizio 2.4. Scrivere una funzione che trova il più piccolo divisore di un intero $n \neq 0, \pm 1$.

3 Numeri primi e teorema fondamentale dell'aritmetica

Se a e b sono due interi con $b > 0$ esistono sempre, e sono unici, il quoziente q e il resto r della divisione di a per b . Se il resto di tale divisione è zero ($r = 0$) si ha: $a = bq$ e b risulta essere un *divisore* di a . Naturalmente ogni intero positivo ha *almeno* due divisori: 1 e il numero stesso. Il numero 360 ha in tutto 24 divisori mentre 359 ne ha esattamente due (1 e 359). Si chiamano *numeri primi* gli interi positivi (maggiore di 1) che hanno esattamente due divisori; questi numeri rivestono un ruolo molto importante in diversi ambiti della matematica.

Definizione 3.1. Un intero $n > 1$ si dice primo se i suoi unici divisori positivi sono 1 e p ; si dice composto se non è primo.

Un primo fatto che evidenzia l'importanza dei numeri primi è questo: ogni intero $n > 1$ si può sempre esprimere come prodotto di numeri primi; per esempio $2860 = 2^2 \cdot 5 \cdot 11 \cdot 13$.

Ogni intero n maggiore di 1, o è primo, e in tal caso la sua fattorizzazione coincide con il numero stesso, oppure è composto; in questo secondo caso $n = qd$, con q e d diversi da 1 e n . Se il numeri q e d non sono primi si possono a loro volta scrivere come prodotto di due interi (diversi da 1 e dal numero) . . . Iterando più volte questo ragionamento si scrive n come prodotto di primi

$$n = p_1 \cdots p_h \quad (3.1)$$

gli interi p_i sono primi non necessariamente distinti.

Teorema 3.2. (Teorema fondamentale dell'aritmetica) *Ogni intero $n > 1$ può essere scritto come prodotto*

$$n = p_1 \cdots p_h$$

dove gli interi $p_1 \cdots p_h$ sono primi e $h \geq 1$. Questa espressione è unica, a meno dell'ordine dei fattori primi.

4 Test di primalità

Un test di primalità consiste in un metodo per stabilire se un dato numero, diciamo n , è primo oppure no. Un semplice algoritmo per risolvere questo problema è il seguente:

Algoritmo elementare per un test di primalità.

Sia n un intero positivo, con $n > 1$.

Si divida n per: $2, 3, 4, \dots (n - 1)$.

- Se nessuna divisione dà resto 0 allora n è primo.
- Se almeno una divisione dà resto 0 allora n è composto.

Esercizio 4.1.

(a) *Scrivere diagramma di flusso pseudocodifica dell'algoritmo riportato sopra.*

(b) *Tradurre la pseudocodifica in un programma (acquisito un intero $n > 1$ da tastiera, stabilire se quel numero è primo oppure no).*

*Denominare il programma: “**primalita01.py**”*

P.S. Il numero 1789, anno della Rivoluzione Francese, è primo.