Matematica con Python[1]

Mauro Saita

e-mail: maurosaita@tiscalinet.it

LEZIONE N. 2

Euclide.

Indice

[1] Nome File: python_lezione_02.tex

1 Iterazioni

1.1 Il ciclo ‘for’ e il comando ‘range’

La sintassi del comando for è la seguente

```
for i in <sequenza>:
   <istruzioni>
```

Per specificare una certa sequenza di numeri si usa spesso il comando **range**. Esso restituisce un numero finito di termini consecutivi di una progressione aritmetica e può essere usato con 1, 2, o 3 parametri. Ecco come funziona:

range(5)	0,1,2,3,4
range(2,8)	$2, 3, 4, 5, 6, 7$
range(3,12,2)	$3, 5, 7, 9, 11$

Esempi di utilizzo del ciclo for.

```
for i in range(2,5):
      print(i)

#output:
2
3
4
```

```
for i in range(5):
      print(i,2*i,i**2)

#output:
0 0 0
1 2 1
2 4 4
3 6 9
4 8 16
```

I cicli for possono essere annidati

```
>>> for i in range(0,5):
        for j in range(0,6):
                print(i+j,end=’ ’)
        print()
```

```
#output:
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
```

1.2 Il ciclo 'while'

La sintassi del ciclo "while" è la seguente:

```
while <Condizione>:
    <istruzioni>
```

```
i=0
while (i<5):
      print(i)
      i=i+1

#output:
0
1
2
3
4
```

1.3 Il comando 'break' e il comando 'continue'

Un modo per uscire da un ciclo 'for' o 'while' è utilizzare il comando break. Per esempio, se si vuole trovare il primo numero maggiore di 100 che è divisibile per 13 si può utilizzare un ciclo while nel modo seguente:

```
x=101
while (x>101):
    if (x%13==0):
        print('Il primo numero maggiore di 100, divisibile per 13 è: '+ str(x))
        break

    x=x+1
```

```
#output:
Il primo numero maggiore di 100, divisibile per 13 è: 104
```

oppure si può utilizzare un ciclo for:

```
for x in range(100,114):
        if (x%13==0):
        print('Il primo numero maggiore di 100 divisibile per 13 è: '+ str(x))
        break
```

Il comando continue interrompe l'esecuzione corrente di un ciclo e passa all'iterazione successiva. Per esempio se si vogliono trovare i numeri pari compresi tra 0 e 10 si può fare così:

```
x=0
while x in range(0,11):
        if (x%2>0):
                x=x+1
                continue
        print(x)
        x=x+1

#output:
0
2
4
6
8
10
```

Oppure, più semplicemente, usando il ciclo for:

```
x=0
for x in range(0,11):
        if (x%2>0):
                continue
        print(x)
```

1.4 Uscita dal programma: sceglie l'utente.

Tutti gli esempi fin qui presentati permettono di mandare in esecuzione programmi per una sola iterazione di input e output. Per esempio il programma che stabilisce se un numero è pari o dispari, chiede di inserire un numero intero (input) e poi scrive a video "numero" o "numero dispari" (output), quindi il programma termina. Se si vuole far decidere dall'utente se continuare a utilizzare il programma oppure no, si può utilizzare un ciclo while infinito e il comando break. Il seguente esempio spiega come

```
while True:
     print("Questo è un ciclo infinito")
     uscita = input("Si desidera uscire dal programma? (s) per sì: ")
     if (uscita==('s' or 'S')):
         break
```

2 Il modulo math

La libreria math permette di eseguire le funzioni riportate in tabella. Per utilizzare questa libreria in un programma bisogna 'importarla' utilizzando l'istruzione:

```
>>>import math
```

In tabella sono riportate le funzioni più importanti contenute in questa libreria; per utilizzarne una qualunque bisogna antemporre al nome della funzione il prefisso math. Per esempio, per calcolare la radice quadrata di 36 si scrive

```
>>> import math
>>> math.sqrt(36)
6.0
```

	Funzione	**Output**
1.	degree(x)	converte l'angolo x da radianti a gradi
2.	radians(x)	converte l'angolo x da gradi a gradi
3.	fmod(x, y)	resto della divisione tra x e y
4.	fabs(x)	valore assoluto di x
4.	sqrt(x)	radice quadrata di x
5.	ceil(x)	approssimazione per eccesso di x in formato float
6.	floor(x)	approssimazione per difetto di x in formato float
7.	hypot(x, y)	$x^2 + y^2$
8.	pow(x, y)	x^y
9.	ldexp(x, i)	$x \cdot 2^i$
10.	frexp(x)	mantissa e esponente di x
11.	modf(x)	parte frazionaria e parte intera di x
12.	exp(x)	e(numero di Nepero) elevato alla x
13.	log(x[, base])	logaritmo in base base di x
14.	log10(x)	logaritmo in base 10 di x
15.	cos(x)	coseno di x misurato in radianti
16.	sin(x)	seno di x misurato in radianti
17.	tan(x)	tangente di x misurato in radianti
18.	acos(x)	arco coseno di x misurato in radianti
19.	asin(x)	arco seno di x misurato in radianti
20.	atan(x)	arco tangente di x misurato in radianti
21.	atan2(y,x)	angolo nella rappresentazione polare di un punto
		$(x, y) \neq (0, 0)$, cioè arctangente di $\frac{y}{x}$ misurato in radianti
22.	cosh(x)	coseno iperbolico di x
23.	sinh(x)	seno iperbolico di x
24.	tanh(x)	tangente iperbolica di x
25.	fsum(v)	somma gli elementi di v
26.	e	Numero di Nepero
27.	pi	Pi greco

L'elenco completo delle funzioni della libreria math si ottiene, da Idle, con il comando dir:

```
>>> import math
>>> dir(math)
```

3 Esercizi

Esercizio 3.1. *Scrivere un programma che, dopo aver acquisito da tastiera un intero positivo n, stampa i numeri da 1 a n e ne calcola la loro somma:* $1 + 2 + 3 + \cdots + n$.

Esercizio 3.2. *Scrivere un programma che, dopo aver acquisito da tastiera un numero dispari $2n - 1$, stampa i primi n numeri dispari e ne calcola la loro somma:* $1 + 3 + 5 \cdots + (2n - 1)$.

Esercizio 3.3 (**Divisori di un numero.**). *Scrivere un programma che stampa tutti i divisori di un intero inserito da tastiera.*

4 Massimo Comun Divisore

Siano a e b due interi. Il massimo comun divisore di due interi a e b, non entrambi nulli, è *il maggiore tra i divisori comuni di a e b.* Si denota con

$$MCD(a, b)$$

In altre parole il $MCD(a, b)$ è un numero d caratterizzato dalle due seguenti proprietà:

- *d è un divisore di a e b.*
- *Se d' è un divisore comune di a e b allora $d' \leq d$.*

Alcuni esempi: $MCD(12, 8) = 4$; $MCD(-12, 8) = 4$; $MCD(15, 0) = 15$. $MCD(a, 0) = |a|$, per ogni $a \neq 0$; $MCD(8, 9) = 1$. Due numeri, come 8 e 9, il cui massimo comun divisore è 1, si dicono *primi tra loro*.

Una proprietà importante del massimo comun divisore che verrà utilizzata più avanti è la seguente:

Il massimo comun divisore di due numeri a e b si può sempre scrivere come combinazione lineare, a coefficienti interi, di a e b e cioè :

$$MCD(a, b) = ha + kb$$

per opportuni interi h e k.

Per esempio il $MCD(12, 8)$ si può scrivere nel seguente modo:

$$MCD(12, 8) = +1 \cdot 12 - 1 \cdot 8.$$

Se a e b sono primi fra loro, allora esistono due interi h e k per i quali

$$ha + kb = 1. \tag{4.1}$$

5 Algoritmo di Euclide per il calcolo del massimo comun divisore di due numeri.

Per trovare il massimo comun divisore di due interi 'piccoli' a e b, si può , in primo luogo, fattorizzare i due numeri in fattori primi e poi calcolare il massimo comun divisore moltiplicando tra loro *i fattori primi comuni ad a e b presi una sola volta con il minimo esponente*. Per esempio se $a = 24$ e $b = 80$, fattorizzare i due numeri è molto semplice: si ottiene: $a = 2^3 \cdot 3$ e $180 = 2^2 \cdot 3^2 \cdot 5$. Quindi il $MCD(24, 80) = 2^2 \cdot 3 = 12$. Tuttavia se a e b sono molto grandi la fattorizzazione in numeri primi richiede tempi di elaborazione lunghissimi.

Un algoritmo molto efficiente per trovare il massimo comun divisore *senza* effettuare alcuna fattorizzazione in numeri primi fu scoperto da Euclide *Elementi, Libro* 7*,* 300 *a.C.*

Algoritmo di Euclide. Siano a e b due interi positivi di cui si vuole determinare il loro massimo comun divisore.

In primo luogo, esistono e sono unici due numeri q e r per i quali:

$$a = bq + r \qquad e \qquad 0 \leq r < b$$

Inoltre,

$$MCD(a, b) = MCD(b, r) \tag{5.1}$$

DIMOSTRAZIONE:

Se un intero d divide sia a che b, allora divide anche la differenza $a - qb = r$; quindi d è anche un divisore comune di b e r.

Viceversa, se d divide sia b che r, allora divide anche la somma $qb+r=a$; quindi d è anche un divisore comune di a e b.

Dunque i divisori comuni di a e b sono esattamente i divisori comuni di b e r, e pertanto $MCD(a,b) = MCD(b,r)$. □

Se si vuole determinare il $MCD(a,b)$ nel caso in cui $a=2406$ e $b=654$, mediante divisioni successive si può procedere come indicato nella te tabella:

a	b	$a=bq+r$	q	r
2406	654	$2406 = 3 \times 654 + 444$	3	444
654	444	$654 = 1 \times 444 + 210$	1	210
444	210	$444 = 2 \times 210 + 24$	2	24
210	24	$210 = 8 \times 24 + 18$	8	18
24	18	$24 = 1 \times 18 + 6$	1	6
18	6	$18 = 3 \times 6 + 0$	3	0

Allora il $MCD(2406,654)=6$. Infatti, applicando ripetutamente l'uguaglianza 5.1, si ha:

$$\begin{aligned}
MCD(2406,654) &= MCD(654,444)\\
MCD(444,210) &= MCD(210,24)\\
MCD(24,18) &= MCD(18,6)\\
MCD(6,0) &= 6
\end{aligned}$$

Quanto all'ultima uguaglianza, si ricordi che per ogni intero positivo a, $MCD(a,0)=a$.

Quindi *l'ultimo resto non nullo è il massimo comun divisore* di a e b.

In termini generali, l'algoritmo di Euclide per la ricerca del massimo comun divisore di due interi a, b, consiste nell'effettuare le divisioni successive:

$$\begin{array}{rcll}
a &=& q_1 b + r_1 & (0 \le r_1 < b)\\
b &=& q_2 r_1 + r_2 & (0 \le r_2 < r_1)\\
r_1 &=& q_3 r_2 + r_3 & (0 \le r_3 < r_2)\\
.. & .. & .... & ....\\
r_{n-2} &=& q_n r_{n-1} + r_n & (0 \le r_n < r_{n-1})\\
r_{n-1} &=& q_{n+1} r_n + 0 &
\end{array}$$

Si osservi che i successivi resti che si ottengono sono non negativi e decrescono strettamente:

$$r_1 > r_2 > r_3 > \cdots > r_{n-1} > \cdots$$

Quindi dopo un numero finito di passi si arriva a un resto nullo. Applicando ripetutamente l'uguaglianza 5.1, si ha:

$$\begin{array}{rcl} MCD(a,b) & = & MCD(b,r_1) \\ MCD(b,r_1) & = & MCD(r_1,r_2) \\ MCD(r_1,r_2) & = & MCD(r_2,r_3) \\ ... & .. & ... \\ MCD(r_{n-2},r_{n-1}) & = & MCD(r_{n-1},r_n) \\ MCD(r_{n-1},r_n) & = & MCD(r_n,0) = r_n. \end{array}$$

Quindi *l'ultimo resto r_n non nullo è il massimo comun divisore* cercato.

5.1 Diagramma di flusso e pseudocodifica

Per costruire un diagramma di flusso bisogna rispettare le seguenti regole:

1. Ogni diagramma di flusso ha un unico punto di inizio ed un unico punto di fine: entrambe queste istruzioni ('Inizio' e 'Fine') sono rappresentate con ellissi.
2. Le operazioni di input e di output ('Leggi' e 'Scrivi') sono rappresentate con parallelogrammi.
3. Le condizioni (if ... else ...) sono espresse all'interno di rombi.
4. Le iterazioni (ciclo for, ciclo while) si descrivono mediante frecce.
5. I box del diagramma di flusso sono collegati fra loro da frecce. In ogni box del diagramma di flusso può entrare una o più frecce (fà eccezione il box 'Inizio' nel quale non entra alcuna freccia).
6. Da ogni box esce una ed una sola freccia a eccezione del box 'Fine' da cui non esce alcuna freccia, e del box contenente una condizione (rombo) da cui escono esattamente due frecce.

Un diagramma di flusso per l'algoritmo di Euclide è il seguente:

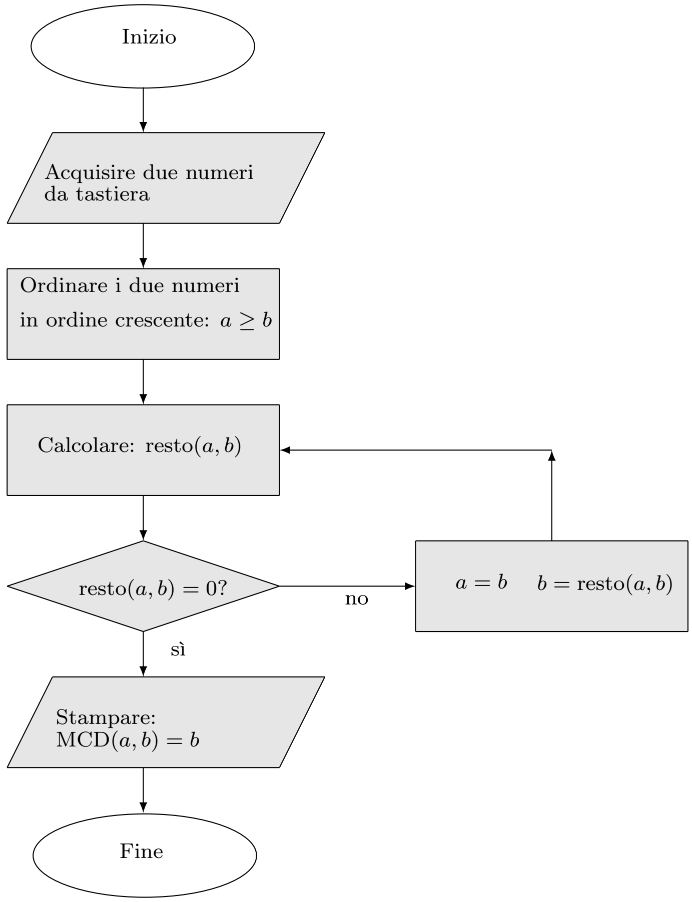

Figura 1: Diagramma di flusso.

La relativa pseudocodifica è questa:

Leggi due numeri: x,y

a=x

b=y

Se a<b **allora**

  scambiare il valore dei due numeri: a=y , b=x

Fine-Se

r=resto(a,b)

Ripeti finché r <> 0

  a=b

  b=r

  r=resto(a,b)

Fine-Ripeti

Scrivi: "Il Massimo Comun Divisore di (x,y) è b"

6 Esercizi

Esercizio 6.1.

(a) *Scrivere la pseudocodifica del programma che calcola il massimo comun divisore con l'algoritmo di Euclide.*

(b) *Scrivere il programma definito in* (a).

È facile constatare, dopo aver eseguito qualche test, la straordinaria efficienza di questo algoritmo. A titolo di curiosità si segnala il seguente importante risultato

Teorema 6.2. (Teorema di Lamé[a]) *Siano a e b due interi positivi con $a \geq b$. Allora il numero di divisioni eseguite dall'algoritmo di Euclide per trovare il $MCD(a, b)$ è minore o uguale a cinque volte il numero di cifre di b.*

[a] Gabriel Lamé (1795-1870) è considerato da Gauss il più importante matematico francese del suo tempo.

Esercizio 6.3. *Scrivere il programma che calcola il massimo comun divisore di 3 numeri.*

Esercizio 6.4. *Siano a e b due interi. Dopo aver verificato che vale l'uguaglianza:*

$$a \cdot b = MCD(a, b) \cdot mcm(a, b)$$

realizzare un programma che determini il minimo comune multiplo *di due numeri interi assegnati.*

Esercizio 6.5 (Il gioco di Collatz). *Scrivere un programma che acquisisce da tastiera un intero $n > 1$ e stampa la successione di numeri che si ottiene con il seguente algoritmo:*
se n è pari il numero successivo è $\dfrac{n}{2}$
se n è dispari il numero successivo è $3 \cdot n + 1$ *e così via.*
Il programma si arresta quando l'ultimo numero scritto è uguale a 1.

Definizione 6.6 (Numeri perfetti). *Un intero positivo n si dice* perfetto *se esso è la somma di tutti i suoi divisori (minori di n)*

Il primi due numeri perfetti sono $6 = 1 + 2 + 3$ e $28 = 1 + 2 + 4 + 7 + 14$.

Esercizio 6.7. *Scrivere un programma che acquisisce un numero da tastiera e stabilisce se è un numero perfetto oppure no.*

Esercizio 6.8. *Scrivere un programma in grado di determinare tutti i numeri perfetti compresi tra due numeri assegnati.*