

Esercizio guidato

Settimana 4
25/10/2022

Si ringrazia il Dott. Giacomo Baruzzo per il materiale

if – else – elif

if **condizione1**

faccio cose

elif **condizione2**

faccio altre cose

else **tutte le condizioni rimanenti**

faccio altre cose ancora

if – else – elif

if **condizione1**

faccio cose

elif **condizione2**

faccio altre cose

else **tutte le condizioni rimanenti**

faccio altre cose ancora

elif e else NON SONO INTERSCAMBIABILI!

if – else – elif

```
from random import randint
a=randint(0,3)
if a == 0:
    ...
elif a == 1:
    ...
else:
    ...
```

Possibili valori di a:

0	1	2	3
----------	----------	----------	----------

if – else – elif

```
from random import randint  
a=randint(0,3)
```

```
if a == 0:
```

```
...
```

```
elif a == 1:
```

Possibili valori di a:

0	1	2	3
---	---	---	---

Importare un modulo in Python

`from random import randint` → importo una o più funzioni → `a=randint(0,3)`

`from random import *` → importo tutte le funzioni → `a=randint(0,3)`

`import random` → importo tutte le funzioni → `a=random.randint(0,3)`

`import random as ran` → importo tutte le funzioni
rinominando il modulo → `a=ran.randint(0,3)`

if – else – elif

```
from random import randint
a=randint(0,3)
if a == 0:
    ...
elif a == 1:
    ...
else:
    ...
```

Possibili valori di a:

0	1	2	3
---	---	---	---

if definisce una condizione specifica

if – else – elif

```
from random import randint
a=randint(0,3)
if a == 0:
    ...
elif a == 1:
    ...
else:
    ...
```

Possibili valori di a:

0	1	2	3
---	---	---	---

elif definisce una condizione **alternativa** all'if, ma **specifica**

if – else – elif

```
from random import randint
a=randint(0,3)
if a == 0:
    ...
elif a == 1:
    ...
else:
    ...
```

Possibili valori di a:

0	1	2	3
---	---	---	---

else definisce **tutte** le condizioni **alternative** non comprese nell'if e/o nell'elif

if – else – elif

```
from random import randint
a=randint(0,3)
if a == 0:
    ...
else:
    ...
```

Possibili valori di a:

0	1	2	3
---	---	---	---

Supponiamo di voler valutare solo se:

- $a==0$

OPPURE

- $a>0$

In questo caso **non** serve usare un **elif** per ogni valore diverso da 0, perchè else valuta già tutti i valori non considerati dall'if!

Tavola Pitagorica (potenze)

- La tavola delle potenze è una matrice in cui alla colonna j -esima della riga i -esima è riportata la potenza j della base i .

$$m(i,j) = i^j$$

- Ogni riga della tabella deve contenere le potenze consecutive di una stessa base, con base crescente dall'alto in basso (come in una normale tavola pitagorica).

		esponente				
		1	2	3	4	5
base	1	1	1	1	1	1
	2	2	4	8	16	32
	3	3	9	27	81	243
	4	4	16	64	256	1024
	5	5	25	125	625	3125

Esempio di tavola con base massima=5 ed esponente massimo=5:

- in posizione (2,5) si ha $2^5=32$
- in posizione (3,4) si ha $3^4=81$

Esercizio

Scrivere il programma **tableOfPowers.py** che visualizzi una tavola pitagorica dopo aver chiesto all'utente i **valori massimi della base e dell'esponente**, che devono essere numeri interi positivi.

Ogni riga della tabella deve contenere le potenze consecutive di una stessa base, con base crescente dall'alto in basso (come in una normale tavola pitagorica).

Porre particolare cura nell'impaginazione della tabella, che deve rispettare queste regole:

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra", cioè la cifra che rappresenta le unità di un valore deve trovarsi nella stessa colonna della cifra che rappresenta le unità degli altri valori.
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Esercizio

Scrivere il programma **tableOfPowers.py** che visualizzi una tavola pitagorica dopo aver chiesto all'utente i **valori massimi della base e dell'esponente**, che devono essere numeri interi positivi.

Ogni riga della tabella deve contenere le potenze consecutive di una stessa base, con base crescente dall'alto in basso (come in una normale tavola pitagorica).

Porre particolare cura nell'impaginazione della tabella, che deve rispettare queste regole:

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra", cioè la cifra che rappresenta le unità di un valore deve trovarsi nella stessa colonna della cifra che rappresenta le unità degli altri valori.
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Algoritmo (soluzione 1)

$$m(x,y) = x^y$$

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo (maxY) dell'esponente y da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una nuova riga della tabella

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una nuova riga della tabella

Dall'algoritmo al codice

```
from sys import exit

print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
```

```
if maxX < 1 or maxY < 1 :
    exit("Errore")
```

```
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da

Modulo sys:

Questo modulo fornisce l'accesso ad alcune variabili utilizzate o mantenute dall'interprete e a funzioni che interagiscono fortemente con l'interprete.

Funzione exit:

Solleva un'eccezione SystemExit, segnalando l'intenzione di uscire dall'interprete.

<https://docs.python.org/3/library/sys.html>

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una nuova riga della tabella

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una nuova riga della tabella

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una

N.B.: Colonne separate da uno spazio!

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo

Per impostazione predefinita, la funzione print di Python termina con un newline.
Il passaggio di uno spazio bianco al parametro end (end=" ") indica che il carattere finale deve essere identificato da uno spazio bianco e non da una newline.

Dall'algoritmo al codice

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

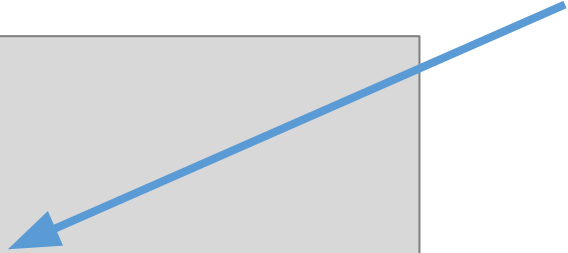
- Leggo il valore massimo (maxX) della base x da input
- Leggo il valore massimo dell'esponente (maxY) da input
- Per ciascun valore della base, da 1 a maxX:
 - Per ciascun valore dell'esponente, da 1 a maxY:
 - Calcolo la base corrente elevata all'esponente corrente
 - Sistemo le potenze consecutive in colonne consecutive e le stampo
 - Esauriti gli esponenti per la base corrente, inizio una nuova riga della tabella

Output della soluzione 1

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
1  1  1  1  1
2  4  8  16 32
3  9  27 81 243
4  16 64 256 1024
5  25 125 625 3125
```

- La tabella non è ben incolonnata!
- In fase di costruzione abbiamo separato le colonne solo con uno spazio

```
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```



Esercizio: consigli di stile

Scrivere il programma **tableOfPowers.py** che visualizzi una tavola pitagorica dopo aver chiesto all'utente i valori massimi della base e dell'esponente, che devono essere numeri interi positivi.

Ogni riga della tabella deve contenere le potenze consecutive di una stessa base, con base crescente dall'alto in basso (come in una normale tavola pitagorica).

Porre particolare cura nell'impaginazione della tabella, che deve rispettare queste regole:

- In ciascuna riga, due valori consecutivi devono essere separati da **almeno uno spazio**.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra", cioè la cifra che rappresenta le unità di un valore deve trovarsi nella stessa colonna della cifra che rappresenta le unità degli altri valori.
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Soluzione 1

```
from sys import exit

print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente?"))
if maxX < 1 or maxY < 1 :
    exit("Errore")

x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(x ** y, " ", end="")
        y += 1
    print() # vado a capo
    x += 1
```

Soluzione 2

```
from sys import exit

print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente?"))
maxValue = maxX ** maxY
columnLength = 1 + len(str(maxValue))
formatString = "%" + str(columnLength) + "s"

x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(formatString % x ** y, end="")
        y += 1
    print() # vado a capo
    x += 1
```

Soluzione 2: le novità

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
```

```
maxValue = maxX ** maxY
```

```
columnLength = 1 + len(str(maxValue))
```

1

```
formatString = "%" + str(columnLength) + "s"
```

```
x = 1
```

```
while x <= maxX :
```

```
    y = 1
```

```
    while y <= maxY :
```

```
        print(formatString % x ** y, end="")
```

```
        y += 1
```

```
    print() # vado a capo
```

```
    x += 1
```

- 1 Calcolo larghezza massima della singola colonna, pari al numero di cifre della potenza più grande

Soluzione 2: le novità

```
from sys import exit

print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))

maxValue = maxX ** maxY
columnLength = 1 + len(str(maxValue))
formatString = "%" + str(columnLength) + "s" ②

x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(formatString % x ** y, end="")
        y += 1
    print() # vado a capo
    x += 1
```

- ① Calcolo larghezza massima della singola colonna, pari al numero di cifre della potenza più grande
- ② Operatore di formato per le stringhe con un campo di tipo %cs, dove c è la larghezza del campo
 - il valore di c non è noto al programmatore, dipende dalla lunghezza della stringa
 - c va calcolata dal programma
 - nel nostro esempio c è pari alla larghezza massima della colonna

Soluzione 2: le novità

```
from sys import exit

print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))

maxValue = maxX ** maxY
columnLength = 1 + len(str(maxValue))
formatString = "%" + str(columnLength) + "s"

x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        print(formatString % x ** y, end="")
        y += 1
    print() # vado a capo
    x += 1
```

- 1 Calcolo larghezza massima della singola colonna, pari al numero di cifre della potenza più grande
- 2 Operatore di formato per le stringhe con un campo di tipo %cs, dove c è la larghezza del campo
 - il valore di c non è noto al programmatore, dipende dalla lunghezza della stringa
 - c va calcolata dal programma
 - nel nostro esempio c è pari alla larghezza massima della colonna
- 3 Ogni cella della tabella sarà della stessa larghezza e conterrà una potenza

Output della soluzione 2

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
  1    1    1    1    1
  2    4    8   16   32
  3    9   27   81  243
  4   16   64  256 1024
  5   25  125  625 3125
```

Ora le colonne sono allineate ma:

1. c'è spazio a sinistra della prima colonna
2. il fatto che le colonne siano tutte della stessa larghezza può generare una tabella troppo "sparsa" se il valore in basso a destra è troppo grande...

Possibile soluzione del problema 2: calcolare separatamente la larghezza di ciascuna colonna!

Versione 3: le novità

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = 1 + len(str(maxValue))
        formatString = "%" + str(columnLength) + "s"
        print(formatString % x ** y, end="")
        y += 1
    print()
    x += 1
```

Calcolo il valore massimo
(potenza più grande) di ogni
colonna, non dell'intera matrice
come in precedenza

Versione 3: le novità

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = 1 + len(str(maxValue))
        formatString = "%" + str(columnLength) + "s"
        print(formatString % x ** y, end="")
        y += 1
    print()
    x += 1
```

NOTA: il calcolo della lunghezza di ogni colonna viene RIPETUTO su ogni riga
--> inefficiente

Output della versione 3

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
 1  1  1  1  1
 2  4  8 16 32
 3  9 27 81 243
 4 16 64 256 1024
 5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi sono separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori sono "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Output della versione 3

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
 1  1  1  1  1
 2  4  8 16 32
 3  9 27 81 243
 4 16 64 256 1024
 5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi sono separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori sono "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Possiamo fare meglio: togliamo lo spazio nella prima colonna

Versione 4: le novità

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        formatString = "%" + str(columnLength) + "s"
        print(formatString % x ** y, end="")
        y += 1
    print() # vado a capo
    x += 1
```

1 Calcolo la larghezza di ogni colonna,
senza aggiungere lo spazio

Versione 4: le novità

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        formatString = "%" + str(columnLength) + "s"
        print(formatString % x ** y, end="")
        y += 1
    print() # vado a capo
    x += 1
```

- 1 Calcolo la larghezza di ogni colonna, senza aggiungere lo spazio
- 2 Se NON sto elaborando la prima colonna, allora aggiungo lo spazio

Output della versione 4

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
1  1  1  1  1
2  4  8 16 32
3  9 27 81 243
4 16 64 256 1024
5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Output della versione 4

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
1  1  1  1  1
2  4  8 16 32
3  9 27 81 243
4 16 64 256 1024
5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.



Soluzione alternativa

Anziché usare l'operatore di formato per le stringhe, stampare gli spazi che servono

Soluzione alternativa

Anziché usare l'operatore di formato per le stringhe, stampare gli spazi che servono

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        value = x ** y
        spaces = columnLength - len(str(value))
        i = 0
        while i < spaces :
            print(" ", end="")
            i += 1
        print(value, end="")
        y += 1
    print()
    x += 1
```

1

Calcolo la potenza da stampare

Soluzione alternativa

Anziché usare l'operatore di formato per le stringhe, stampare gli spazi che servono

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        value = x ** y
        spaces = columnLength - len(str(value))
        i = 0
        while i < spaces :
            print(" ", end="")
            i += 1
        print(value, end="")
        y += 1
    print()
    x += 1
```

- 1 Calcolo la potenza da stampare
- 2 Calcolo il numero di spazi necessari:
lung colonna – lung numero

Soluzione alternativa

Anziché usare l'operatore di formato per le stringhe, stampare gli spazi che servono

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        value = x ** y
        spaces = columnLength - len(str(value))
        i = 0
        while i < spaces :
            print(" ", end="")
            i += 1
        print(value, end="")
        y += 1
    print()
    x += 1
```

- 1 Calcolo la potenza da stampare
- 2 Calcolo il numero di spazi necessari:
lung colonna – lung numero
- 3 Stampo gli spazi con un ciclo

Soluzione alternativa

Anziché usare l'operatore di formato per le stringhe, stampare gli spazi che servono

```
from sys import exit
print("Tavola Pitagorica fino a x elevato alla y")
maxX = int(input("Valore massimo della base? "))
maxY = int(input("Valore massimo dell'esponente? "))
if maxX < 1 or maxY < 1 :
    exit("Errore")
x = 1
while x <= maxX :
    y = 1
    while y <= maxY :
        maxValue = maxX ** y
        columnLength = len(str(maxValue))
        if y != 1 :
            columnLength += 1
        value = x ** y
        spaces = columnLength - len(str(value))
        i = 0
        while i < spaces :
            print(" ", end="")
            i += 1
        print(value, end="")
        y += 1
    print()
    x += 1
```

- 1 Calcolo la potenza da stampare
- 2 Calcolo il numero di spazi necessari:
lung colonna – lung numero
- 3 Stampo gli spazi con un ciclo
- 4 Stampo il valore della potenza

Output della soluzione alternativa

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
1  1  1  1  1
2  4  8 16 32
3  9 27 81 243
4 16 64 256 1024
5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

Output della soluzione alternativa

```
Tavola Pitagorica fino a x elevato alla y
Valore massimo della base? 5
Valore massimo dell'esponente? 5
1  1  1  1  1
2  4  8 16 32
3  9 27 81 243
4 16 64 256 1024
5 25 125 625 3125
```

- In ciascuna riga, due valori consecutivi devono essere separati da almeno uno spazio.
- In ciascuna colonna, tutti i valori devono essere "allineati a destra"
- La lunghezza delle righe deve essere quella minima che risulta compatibile con le regole precedenti.

