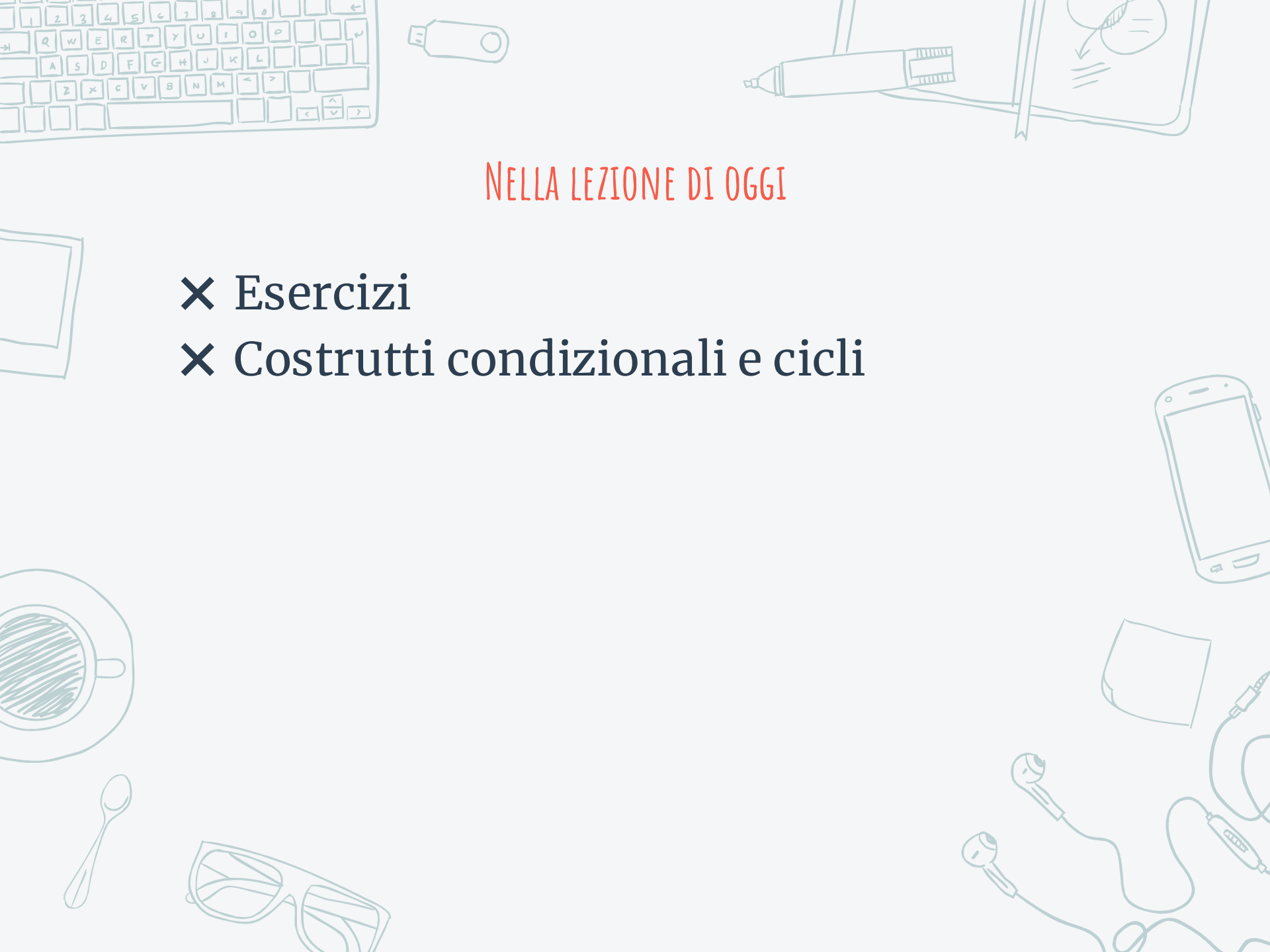




LABORATORIO DI PENSIERO COMPUTAZIONALE

Simone Zennaro – simone.zennaro@unipd.it



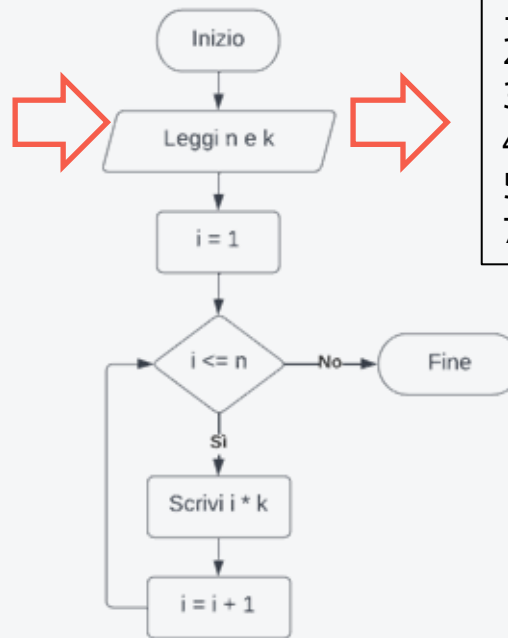
NELLA LEZIONE DI OGGI

- ✕ Esercizi
- ✕ Costrutti condizionali e cicli

ESERCIZIO 1

- ✗ Dati $n, k > 0$, calcolare i primi n multipli di k .
- ✗ La soluzione del problema è piuttosto elementare: letti in input i dati del problema, si iterano le istruzioni con cui si calcola l' i -esimo multiplo di k (per $i = 1, 2, \dots, n$) come prodotto $i \times k$ e si stampa il risultato.

```
1: leggi n e k
2: i = 1
3: se i > n vai al passo 7
4: scrivi i * k
5: i = i + 1
6: vai al passo 3
7: stop
```



```
1: leggi n e k
2: i=1
3: fintanto che i <= n
4:     scrivi i * k
5:     i = i + 1
7: stop
```

TRACCIAMO INSIEME I MULTIPLI

```
leggi n, k  
i = 1  
fintanto che  $i \leq n$ :  
    scrivi  $i \times k$   
     $i = i + 1$ 
```

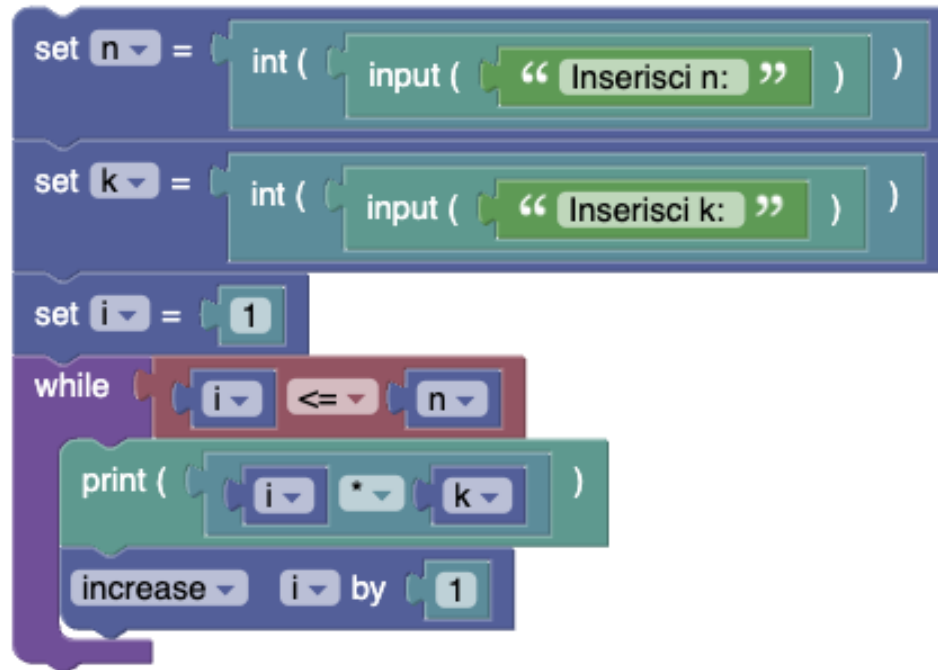
Con $n = 3$ e $k = 4$.

Ogni riga è un passaggio dal rombo: prima si controlla la condizione, poi si esegue il corpo del ciclo. Il valore nuovo si scrive accanto al vecchio, non al posto suo.

Passo	i	$i \leq n$?	Scrive
inizio	1	—	
1	1	Vero	4
2	2	Vero	8
3	3	Vero	12
4	4	Falso	esce dal ciclo

ESERCIZIO 1

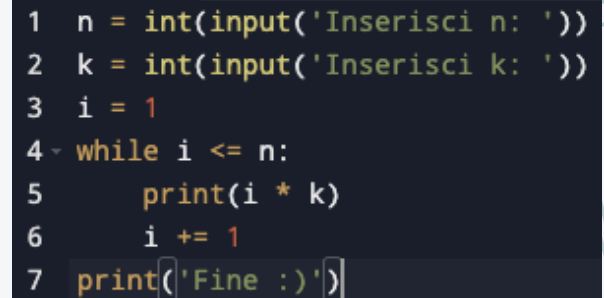
- ✗ Dati $n, k > 0$, calcolare i primi n multipli di k .
- ✗ La soluzione del problema è piuttosto elementare: letti in input i dati del problema, si iterano le istruzioni con cui si calcola l' i -esimo multiplo di k (per $i = 1, 2, \dots, n$) come prodotto $i \times k$ e si stampa il risultato.



Scratch code blocks for the exercise:

```
set n = int ( input ( " Inserisci n: " ) )
set k = int ( input ( " Inserisci k: " ) )
set i = 1
while i <= n
  print ( i * k )
  increase i by 1
```

PYTHON



```
1 n = int(input('Inserisci n: '))
2 k = int(input('Inserisci k: '))
3 i = 1
4 while i <= n:
5     print(i * k)
6     i += 1
7 print('Fine :)|')
```

ESERCIZIO 2



Dati $n, k > 0$ stabilire se n è divisibile per k .

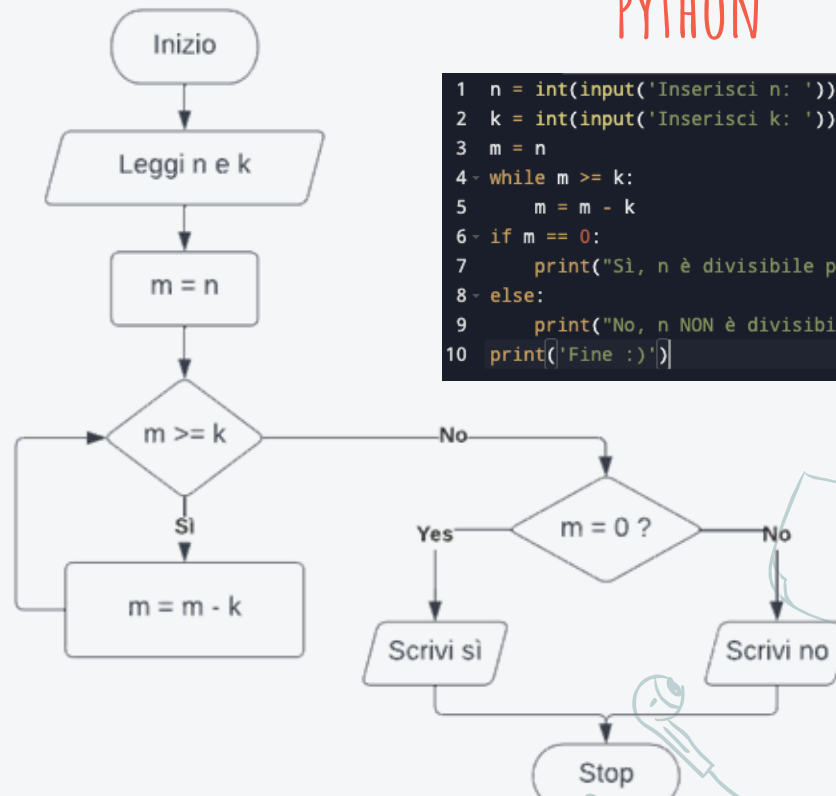
La soluzione si basa sull'applicazione di una definizione assai elementare del concetto di divisibilità tra due numeri interi: possiamo dire che n è divisibile per k se k "entra" in n esattamente d volte; in altri termini n è divisibile per k se esiste un intero d tale che $n = kd$. La verifica avviene quindi sottraendo ripetutamente k da n fino ad ottenere il valore 0 nel caso in cui n sia divisibile per k o un numero maggiore di zero e minore di k , nel caso in cui n non sia divisibile per k .

```
1: leggi n e k
2: m = n
3: se m >= k allora prosegui,
   altrimenti vai al passo 6
4: m = m - k
5: vai al passo 3
6: se m = 0 allora scrivi "Sì, n è divisibile
   per k"
7: altrimenti scrivi "No, n non è divisibile
   per k"
8: stop
```

```
1: leggi n e k
2: m = n
3: fintanto che  $m \geq k$ :
4:      $m = m - k$ 
6: se  $m = 0$  allora
7:     scrivi "Sì, n è
       divisibile per k"
8: altrimenti
9:     scrivi "No, n non è
       divisibile per k"
10: stop
```

PYTHON

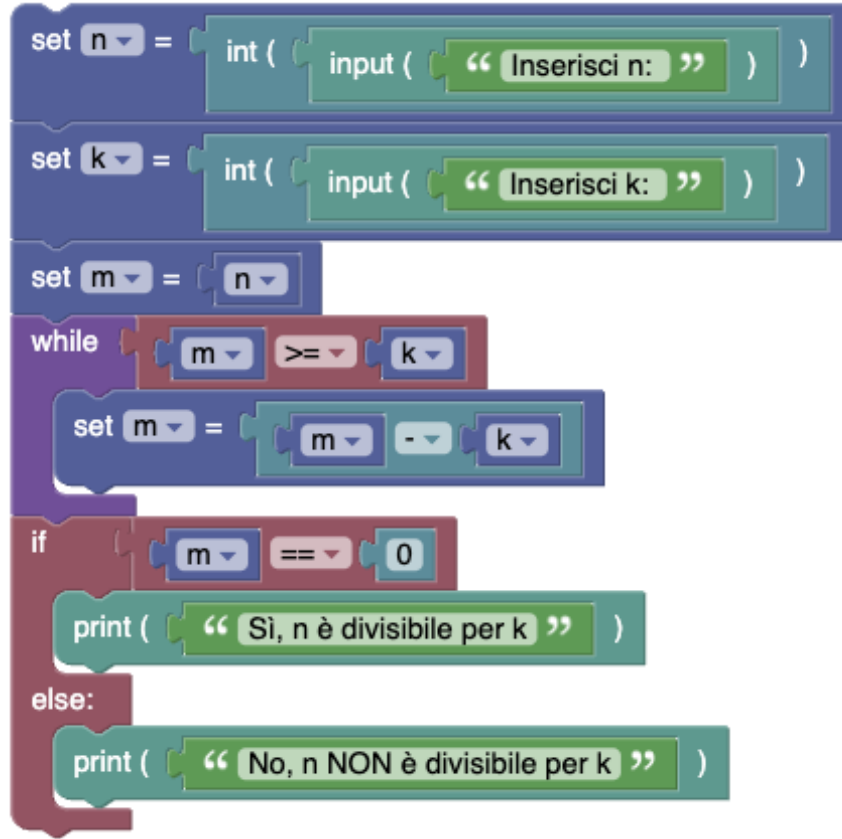
```
1 n = int(input('Inserisci n: '))
2 k = int(input('Inserisci k: '))
3 m = n
4 while m >= k:
5     m = m - k
6 if m == 0:
7     print("Sì, n è divisibile per k")
8 else:
9     print("No, n NON è divisibile per k")
10 print('Fine :)' )
```



ESERCIZIO 2

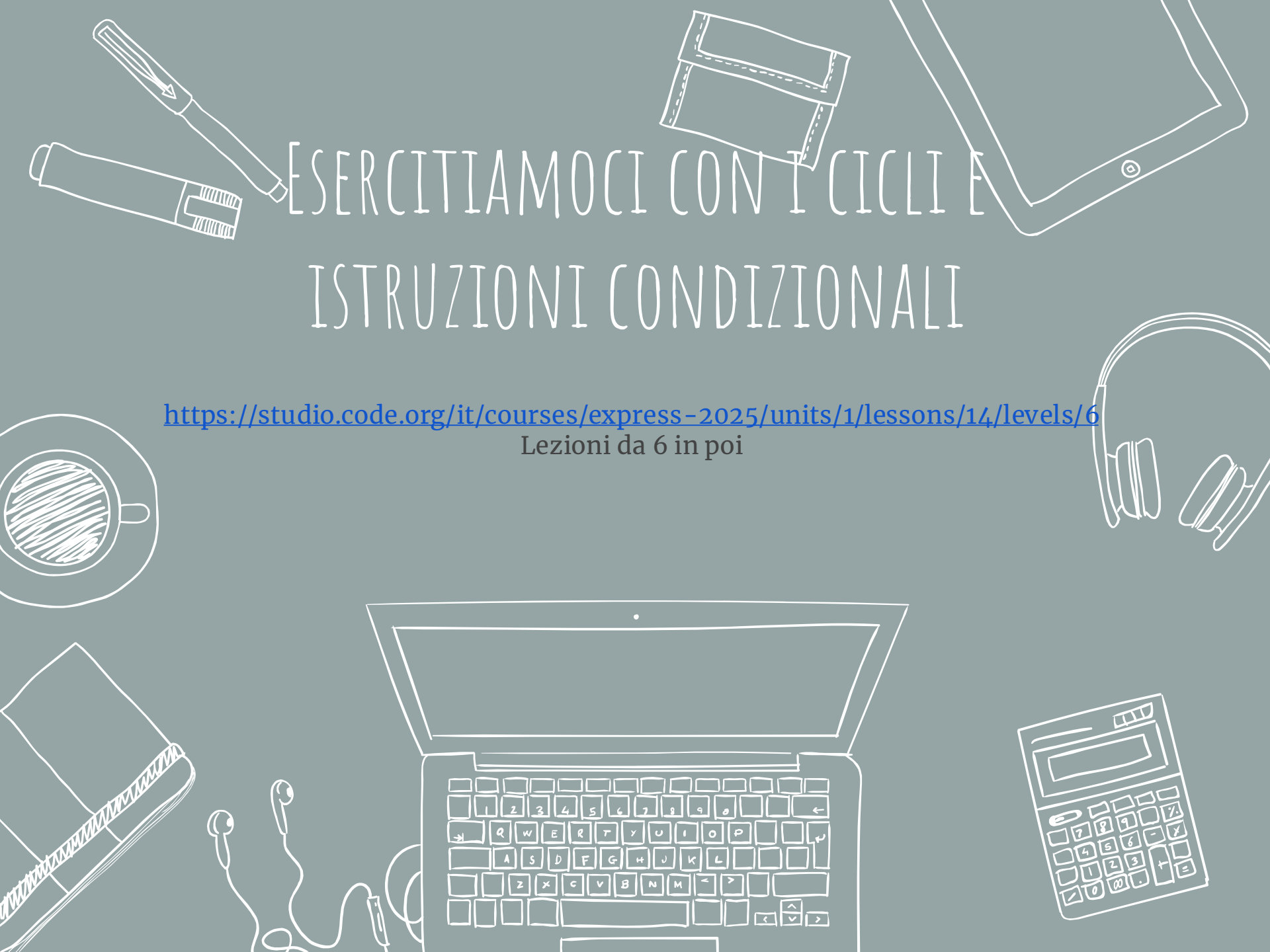
✕✕ Dati $n, k > 0$ stabilire se n è divisibile per k .

La soluzione si basa sull'applicazione di una definizione assai elementare del concetto di divisibilità tra due numeri interi: possiamo dire che n è divisibile per k se k "entra" in n esattamente d volte; in altri termini n è divisibile per k se esiste un intero d tale che $n = kd$. La verifica avviene quindi sottraendo ripetutamente k da n fino ad ottenere il valore 0 nel caso in cui n sia divisibile per k o un numero maggiore di zero e minore di k , nel caso in cui n non sia divisibile per k .



PYTHON

```
1 n = int(input('Inserisci n: '))
2 k = int(input('Inserisci k: '))
3 m = n
4 while m >= k:
5     m = m - k
6 if m == 0:
7     print("Sì, n è divisibile per k")
8 else:
9     print("No, n NON è divisibile per k")
10 print('Fine :)')
```



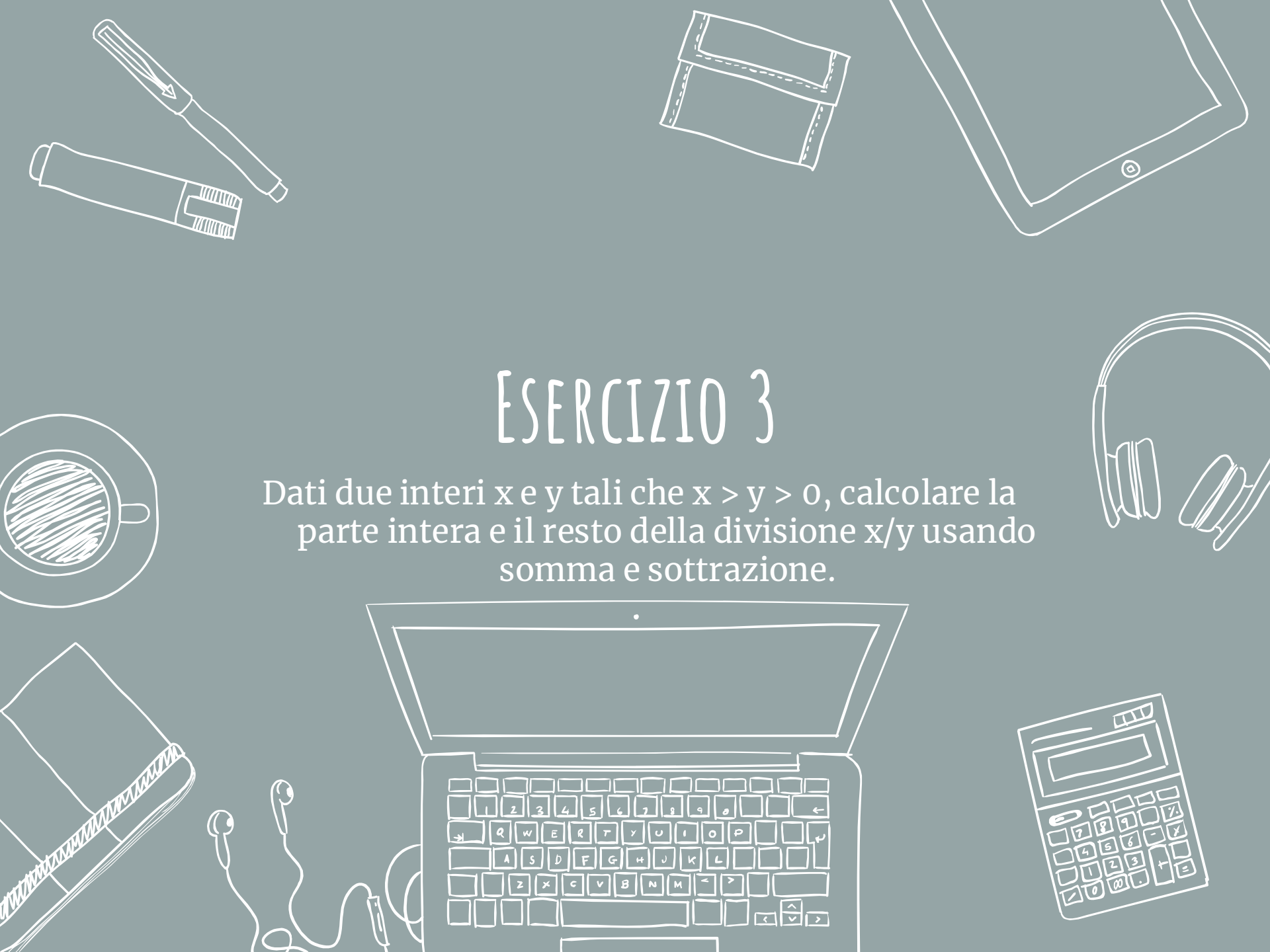
ESERCITIAMOCI CON I CICLI E ISTRUZIONI CONDIZIONALI

<https://studio.code.org/it/courses/express-2025/units/1/lessons/14/levels/6>

Lezioni da 6 in poi

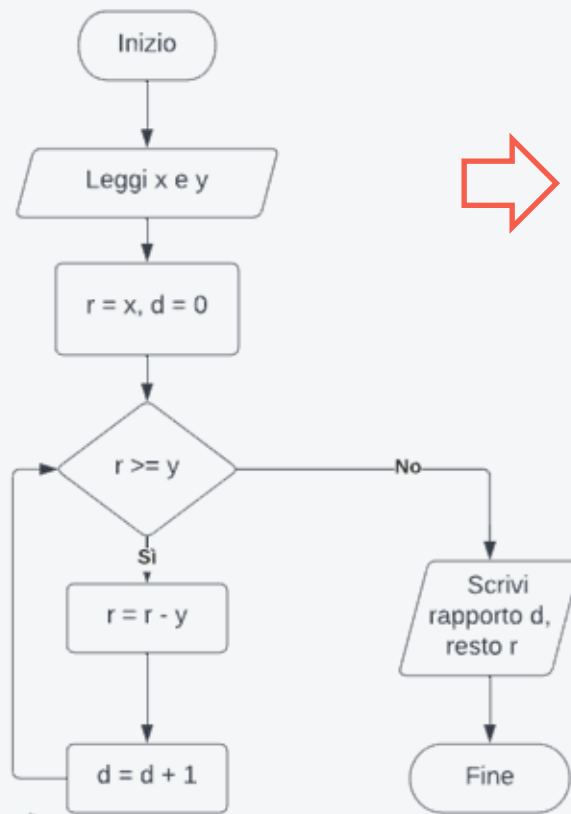
ESERCIZIO 3

Dati due interi x e y tali che $x > y > 0$, calcolare la parte intera e il resto della divisione x/y usando somma e sottrazione.



ESERCIZIO 3

- ✗ Dati due interi x e y tali che $x > y > 0$, calcolare la parte intera e il resto della divisione x/y .
- ✗ Il rapporto x/y è uguale a d con resto r ($0 \leq r < y$) se $x = y * d + r$. In termini più rudimentali possiamo dire che d , la parte intera del risultato della divisione, è dato da quante volte y “entra” in x ; mentre il resto r è dato dal valore che “avanza” dopo aver tolto per d volte y da x .



```
1: leggi x e y
2: r = x
3: d = 0
4: fintanto che r >= y:
5:     r = r - y
6:     d = d + 1
7: scrivi "x/y = d con resto r"
8: stop
```

PYTHON ?

ESERCIZIO 3



Dati due interi x e y tali che $x > y > 0$, calcolare la parte intera e il resto della divisione x/y .



Il rapporto x/y è uguale a d con resto r ($0 \leq r < y$) se $x = y * d + r$. In termini più rudimentali possiamo dire che d , la parte intera del risultato della divisione, è dato da quante volte y “entra” in x ; mentre il resto r è dato dal valore che “avanza” dopo aver tolto per d volte y da x .

PYTHON

```
set x = int ( input ( "Inserisci n: " ) )
set y = int ( input ( "Inserisci k: " ) )
set r = x
set d = 0
while r >= y:
    set r = r - y
    set d = d + 1
print ( x , " / " , y , " = " , d , " con resto " , r )
```

```
1 x = int(input("Inserisci x: "))
2 y = int(input("Inserisci y: "))
3 r = x
4 d = 0
5 while r >= y:
6     r = r - y
7     d = d + 1
8 print(f"{x}/{y} = {d} con resto {r}")
9 print("fine :)|")
```

ESERCIZIO 4

Dati due numeri interi positivi x e y calcolare il massimo comune divisore tra x e y .

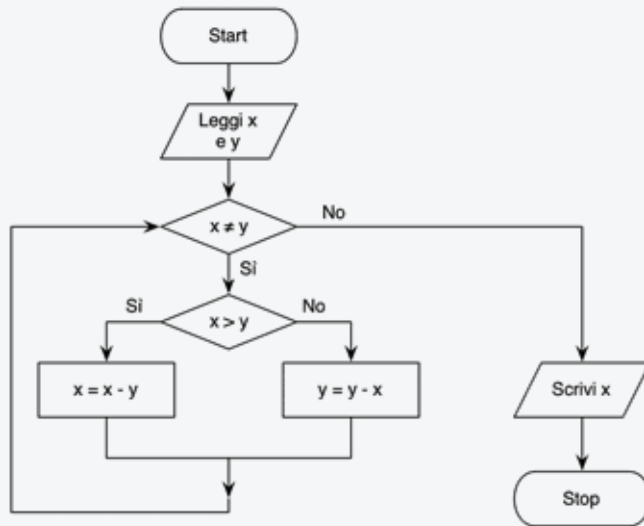


ESERCIZIO 4



Dati due numeri interi positivi x e y calcolare il massimo comune divisore tra x e y .

Per costruire un algoritmo elementare per calcolare il MCD (massimo comun divisore) tra x e y ancora una volta facciamo ricorso ad una divisione piuttosto elementare in modo da costruire la strategia risolutiva basandoci soltanto su operazioni aritmetiche di base. In particolare l'idea è quella di applicare una variante del cosiddetto "algoritmo di Euclide": si tratta di sottrarre ripetutamente x da y (se $y > x$) o y da x (se $x > y$) fino ad ottenere lo stesso valore per x e y ; tale valore è proprio il massimo comun divisore dei due numeri originali.



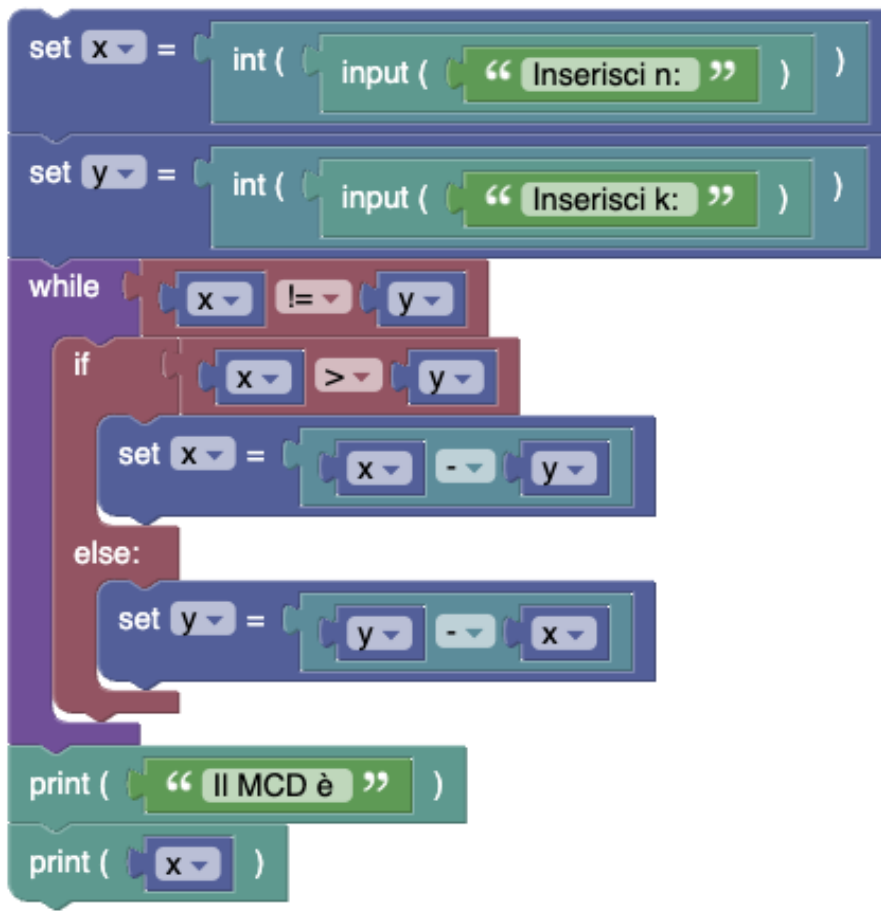
```
1: leggi x e y
4: fintanto che  $x \neq y$  ripeti
5:   se  $x > y$  allora
6:      $x = x - y$ 
7:   altrimenti
8:      $y = y - x$ 
9: scrivi "il MCD è x"
10: stop
```

ESERCIZIO 4



Dati due numeri interi positivi x e y calcolare il massimo comune divisore tra x e y .

Per costruire un algoritmo elementare per calcolare il MCD (massimo comun divisore) tra x e y ancora una volta facciamo ricorso ad una divisione piuttosto elementare in modo da costruire la strategia risolutiva basandoci soltanto su operazioni aritmetiche di base. In particolare l'idea è quella di applicare una variante del cosiddetto "algoritmo di Euclide": si tratta di sottrarre ripetutamente x da y (se $y > x$) o y da x (se $x > y$) fino ad ottenere lo stesso valore per x e y ; tale valore è proprio il massimo comun divisore dei due numeri originali.



PYTHON

```
1 x = int(input('Inserisci x: '))
2 y = int(input('Inserisci y: '))
3 while x != y:
4     if x > y:
5         x = x - y
6     else:
7         y = y - x
8 print(f"Il MCD è {x}")
9 print('Fine :)')
```