

## Dato un numero stabilire se sia o meno primo

La ricerca dei divisori di un numero può essere fatta dividendo un numero dato per i numeri naturali da 1 a  $n$ . Per trovare i divisori di 12 serve quindi verificare quali valori da 1 a 12 usati come divisori con 12 come dividendo diano resto 0.

12:1=12 r=0	12:2=6 r=0	12:3=4 r=0	12:4=3 r=0	12:5=2 r=2	12:6=2 r=0
12:7=1 r=5	12:8=1 r=4	12:9=1 r=3	12:10=1 r=2	12:11=1 r=1	12:12=1 r=0

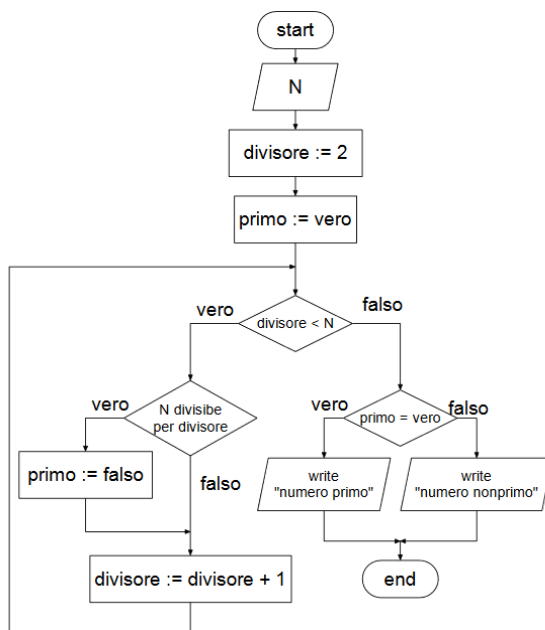
$$D(12) = \{1, 2, 3, 4, 6, 12\}$$

Dall'analisi delle operazioni si vede come considerando i divisori del 12 a un certo punto divisori e quozienti si invertono. A quel punto non sarebbe necessario proseguire.

Il limite massimo dei divisori di un numero  $n$  è dato, infatti, dal valore di  $n/2$ . Non è infatti necessario provare se tutti i numeri da 2 a  $n$  se sono divisori del numero ma è sufficiente controllare i numeri da 2 a  $n/2$  siano o meno divisori del numero dato.

E' nello stesso modo sufficiente fare la verifica da 2 a  $\sqrt{n}$ . Per ogni divisore di  $n$ , minore della sua radice quadrata, ce n'è uno maggiore di essa e i due sono strettamente collegati ed è inutile ricercare divisori di  $n$  maggiori della sua radice quadrata.

L'algoritmo per valutare se un numero è primo si può rappresentare nel modo seguente.



**start**

```

read N
divisore := 2
primo := vero
while (divisore < N)
    if (N divisibile per divisore) then
        primo := falso
    end if
    divisore := divisore + 1
end while
if (primo = vero) then
    write "numero primo"
else
    write "numero non primo"
end if
end
    
```

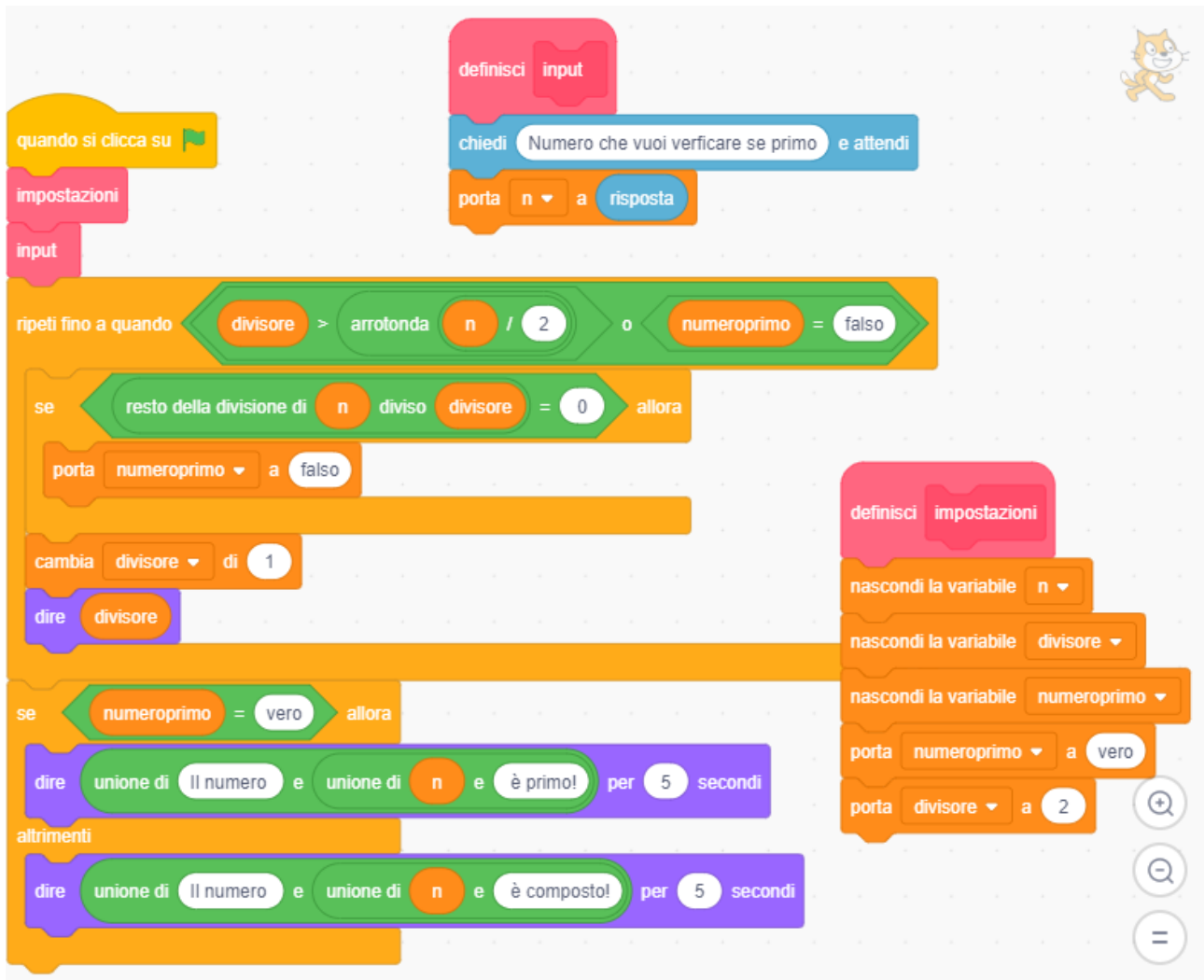
<http://www.di.unito.it/~rossano/DIDATTICA/MDAMS-0405/mdams-17.pdf>

## Codice Python

```

n=int(raw_input("Inserisci un numero naturale n maggiore di 1: "))
primo=True # ipotizziamo che N sia primo
for i in range(2,n):
    if n%i==0:
        primo=False
    if primo: # se N è primo
        print n," il numero è primo"
else:
    print n," il numero è composto"
    
```

È possibile utilizzare un linguaggio di programmazione a blocchi anche per questo scopo e realizzare una procedura con cui allenarsi o con cui trovare i divisori di un numero qualsiasi. L'algoritmo e il relativo diagramma di flusso è traducibile nella seguente codifica a blocchi.



Nella prima si usano due blocchi personalizzati, uno per le impostazioni e uno per la acquisizione (input) del numero di cui verificare la primalità.

## METTITI ALLA PROVA

1.

Per numeri grandi il sistema sta fermo a pensare senza dare alcun cenno all'utente che è al lavoro. Inserisci la visualizzazione del divisore di cui sta eseguendo il test in modo che si veda a video che i lavori proseguono.

2.

Introduci un controllo per l'inserimento di un valore 0 e 1, da considerare con accettabili. Inserisci i blocchi [Azzera cronometro] e [Cronometro] per visualizzare il tempo di esecuzione con numeri via via crescenti (100, 1000, 10000, ...) e annotali.

3.

Inserisci il controllo da 2 a  $\sqrt{n}$  e verifica che ottieni gli stessi risultati.

Modifica il codice per ottenere di un numero primo (ha solo i divisori banali) o composto (ha più di due divisori) i suoi divisori. Ricorda che 1 non è primo o composto.