

CORSO DI FONDAMENTI DI INFORMATICA

ESERCITAZIONE 5

PROF. GABRIELE DI STEFANO - DR. FRANCESCO GALLO

Blocco 0 - III Piano - francesco.gallo@univaq.it

Web: <http://people.disim.univaq.it/~francesco.gallo/fondamenti.html>

Buzz.py

```
n = int(input("Inserire un numero intero: "))

i = 1
j = ""
trovato = False
while i < n + 1:
    trovato = False
    if i % 7 == 0:
        print("Buzz - {}".format(i))
    else:
        j = str(i)
        k = 0
        while k < len(j):
            if j[k] == '7':
                print("Buzz - {}".format(i))
                k = len(j)
                trovato = True
            else:
                k = k + 1

        if trovato == False:
            print(i)

    i = i + 1
```


BuzzBis.py

```
n = int(input("Inserire un numero intero: "))

i = 1
j = ""
trovato = False
while i < n + 1:
    trovato = False
    if i % 7 == 0:
        print("Buzz - {}".format(i))
        i = i + 1
    else:
        j = i
        while j != 0:
            if (j % 10 == 7):
                print("Buzz - {}".format(i))
                j = 0
                trovato = True
            else:
                j = j / 10
        if trovato == False:
            print(i)

    i = i + 1
```

turtleEsercizio7tris.py

```
import turtle
import math

n = int(input("Inserire il numero dei lati del poligono: "))
raggio = int(input("Inserire la lunghezza del raggio della circonferenza: "))

osvaldo = turtle.Turtle()
osvaldo.shape("turtle")

osvaldo.pensize(3)
osvaldo.pencolor('red')
osvaldo.speed(1)

alfa = math.radians(360 / n)
lato = 2 * raggio * math.sin(alfa/2)

l = 0

while l < n:
    osvaldo.forward(lato)
    osvaldo.left(360/n)
    l = l + 1

osvaldo.pencolor('white')
osvaldo.goto(lato/2, -(raggio - (raggio * math.cos(math.radians(180)/n))))

osvaldo.pencolor('blue')
osvaldo.circle(raggio)
```

Operatori di scorrimento

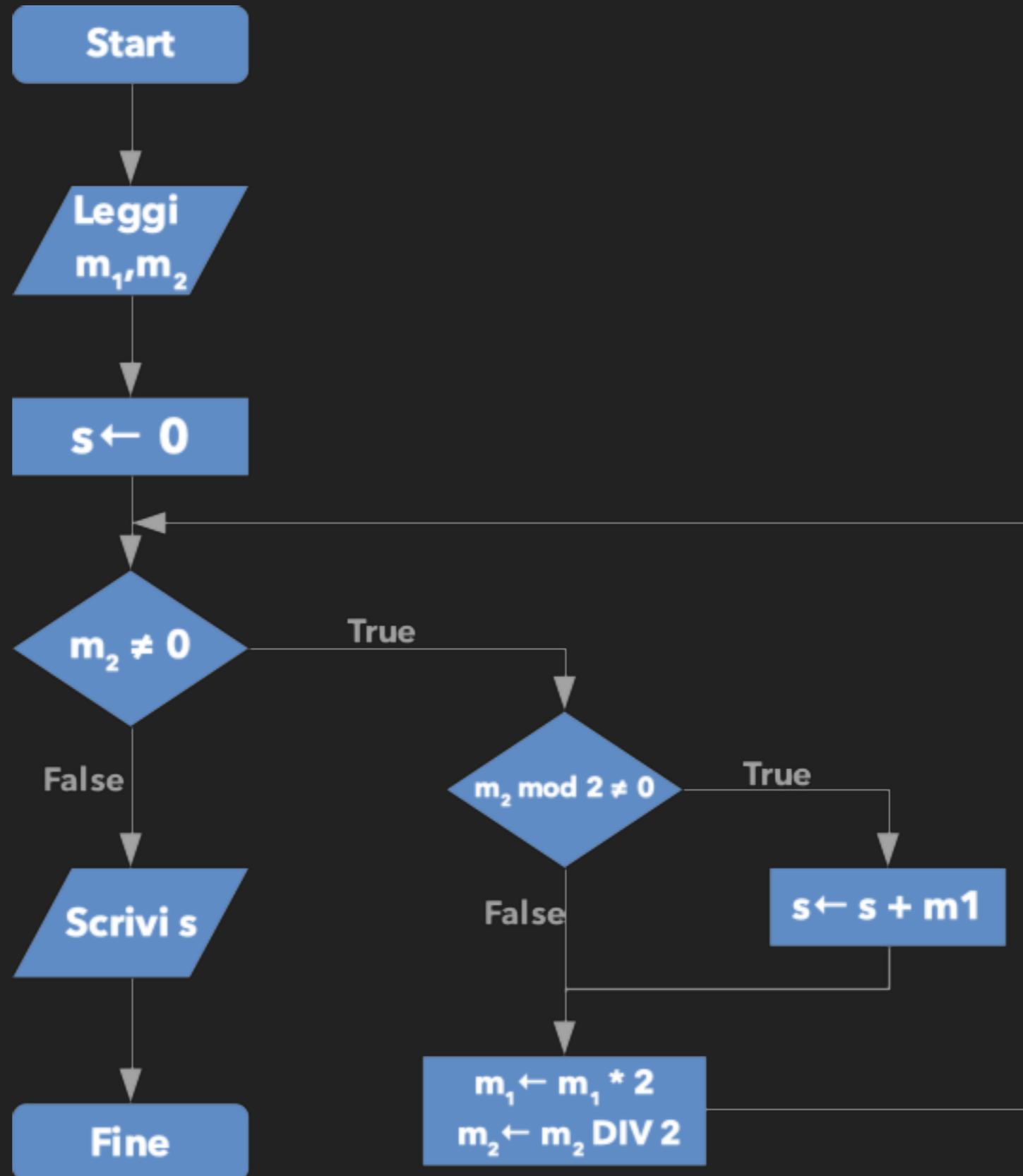
Il Python mette a disposizione degli operatori che lavorano su numeri di tipo intero manipolando il dato a livello di singolo bit. Non si applicano ad operandi floating point o a dati di tipo strutturato.

>> SHIFT (SCORRIMENTO) a DESTRA

<< SHIFT (SCORRIMENTO) a SINISTRA

```
a = 61  
print(a >> 1) #Stampa valore  
print(a << 1) #Stampa valore
```

russianAlgorithm.py



russianAlgorithm.py

```
m1 = int(input("Inserire primo valore: "))
m2 = int(input("Inserire secondo valore: "))

s = 0

while m2 != 0:
    if m2 % 2 != 0:
        s += m1
    m1 *= 2
    m2 = m2 // 2

print("s: {}".format(s))
```

russianAlgorithm.py

```
m1 = int(input("Inserire primo valore: "))
m2 = int(input("Inserire secondo valore: "))

s = 0

while m2 != 0:
    if m2 % 2 != 0:
        s += m1
    m1 = m1 << 1
    m2 = m2 >> 1

print("s: {}".format(s))
```

Operatori di scorrimento

Il Python mette a disposizione degli operatori che lavorano su numeri di tipo intero manipolando il dato a livello di singolo bit. Non si applicano ad operandi floating point o a dati di tipo strutturato.

>> SHIFT (SCORRIMENTO) a DESTRA

<< SHIFT (SCORRIMENTO) a SINISTRA

```
a = 61
```

```
print(a >> 1) #30
```

```
print(a << 1)
```

Si noti come l'operatore corrisponda ad una divisione intera per multipli di 2.

Shiftare a destra di 1 significa dividere per 2, shiftare di 2 è dividere per 4.

Operatori di scorrimento

Il Python mette a disposizione degli operatori che lavorano su numeri di tipo intero manipolando il dato a livello di singolo bit. Non si applicano ad operandi floating point o a dati di tipo strutturato.

>> SHIFT (SCORRIMENTO) a DESTRA

<< SHIFT (SCORRIMENTO) a SINISTRA

```
a = 61
print(a >> 1)
print(a << 1) #122
```

Si noti come l'operatore corrisponda ad una divisione intera per multipli di 2.

Shiftare a destra di 1 significa dividere per 2, shiftare di 2 è dividere per 4.

Notare come questo operatore corrisponda al risultato di una moltiplicazione intera per un multiplo di due finché a sinistra non esce qualche bit ad 1.