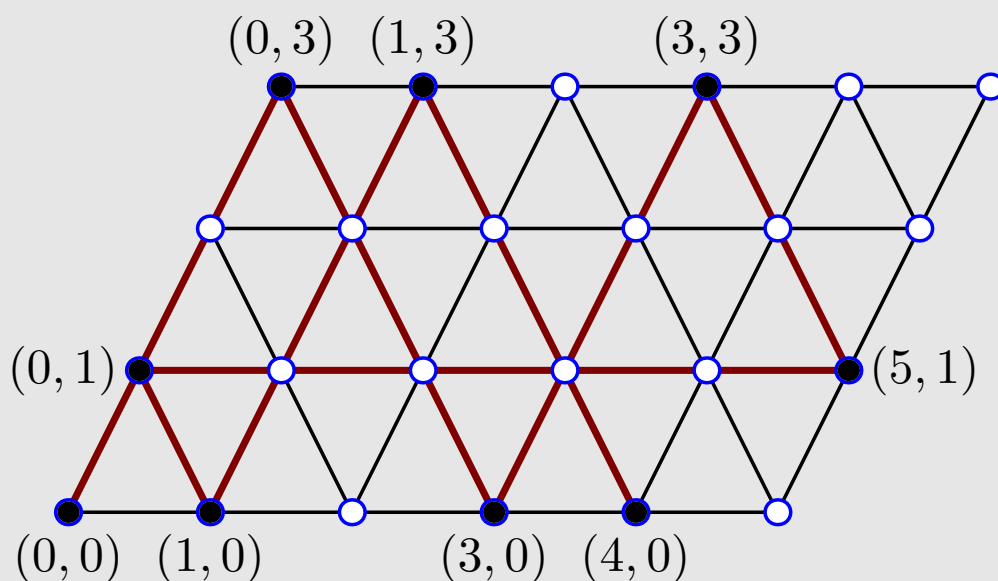




Umri pokončno

V filmu *Umri pokončno 3* morata Bruce Willis in Samuel Jackson, prvi v vlogi newyorškega policaja, drugi v vlogi črnega aktivista, reševati uganke, ki jima jih nastavlja Jeremy Irons, ki igra vlogo negativca Petra. Če ne bi razrešila uganke pravočasno, bi povzročila eksplozijo bombe. Prizorišče, kjer morata razrešiti eno od uganek, je vodnjak, ob katerem se nahaja kovček in dve posodi, ena drži 5 galon, druga pa 3 galone. Ko odpreta kovček, se zavrti posnetek s Petrovim glasom, ki pravi, da je v kovčku bomba in da imata 30 sekund časa, da jo deaktivirata, tako da postavita posodo s štirimi galonami vode na tehtnico, ki se nahaja v kovčku. Ker nimata dovolj časa, si pomagata le s posodama, ki sta na razpolago. V naglici nalogo razrešita in deaktivirata bombo.

Glej <https://www.youtube.com/watch?v=00ef3MHYEC0>.



Rešitev Petrove naloge

Če napolnita veliko posodo in vodo prelijeta v manjšo, bosta v veliki posodi ostali dve galoni vode. Odlijemo vodo iz manjše posode in prelijemo vodo iz večje posode v manjšo. V manjši sta dve galoni vode. Če napolnita veliko posodo in z njo dopolnita manjšo posodo, ostane v veliki štiri galone vode. To ni edina rešitev, glej sliko. Problem lahko tudi nima rešitve.

Definicija problema

Dani sta prostornini dveh posod in količina vode, podana v galonah, ki jo moramo odmeriti. Označimo prostornino prve posode V_1 , prostornina druge posode je V_2 , količina vode, ki jo moramo odmeriti, pa je $V \leq \max(V_1, V_2)$. Vaša naloga je, da pri danih treh podatkih V_1 , V_2 in V ugotovite, ali je naloga rešljiva ali ne.

Vhodni podatki

S pomočjo ukaza `input` preberete 3 podatke. V_1 , V_2 in V . Števila so zapisana v eni vrstici, ločena so s presledki.

Izhodni podatki

Izpišite eno od poti do rešitve, če je naloga rešljiva, in `False`, če naloga nima rešitve. Glej sliko.

Primer

Vhod

3 5 4

Izhod

[[0, 3], [3, 0], [3, 3], [5, 1], [0, 1], [1, 0], [1, 3], [4, 0]]

Rešitev

```
#!/usr/bin/env python3
def gcd(m, n):
    if m < n:
        m, n = n, m
    while n > 0:
        m, n = n, m % n
    return m

def go(V1, V2, V):
    state = [0, 0]
    target = [V, 0]
    if V1 < V2:
        V1, V2 = V2, V1
    while state != target:
        if state[0] == V1:
            state[0] = 0
        else:
            state[-1] = V2
        print(state)
        z = min(V1 - state[0], state[-1])
        state[0] += z
        state[1] -= z
        print(state)

if __name__ == "__main__":
    V1, v2, V = [int(x) for x in input('Vpisi: V1 V2 V --> ').split()]
    m = gcd(V1, V2)
    if V % m:
        print('False')
    else:
        go(V1, V2, V)
```