Da quale dei due stacks rimuovere e quanto? (game2stacks)

Alice e Bob si alternano nell'atto di rimuovere pedine disposte in due pile. Ad ogni suo turno, ciascun giocatore deve scegliere una delle due pile e quante pedine rimuovere dalla pila scelta. Il giocatore che non riesce più a muovere (perchè il numero di pedine è ormai sceso a zero in entrambe le pile) perde la partita, mentre il suo avversario viene proclamato vincitore degli undergames 2018.12.05.

Devi scrivere una funzione che riceve in input due parametri n_1 ed n_2 che specificano il numero di pedine attualmente presenti sul primo e sul secondo stack, rispettivamente. La funzione deve individuare se esista una mossa vincente e, in caso affermativo, dire quale. In caso contrario la funzione abbandona il gioco rifiutandosi di muovere.

Trovi un template della soluzione nel file **game2stacks_template_sol.py** tra gli attachments, dovrai solo curare l'implementazione della funzione richiesta:

```
def play(n1, n2):
    return (0,0)
```

Attualmente `play` ritorna sempre $(0, 0)$, che equivale ad abbandonare la partita senza rimuovere alcuna pedina, da nessuna pila. Tuttavia questa è la risposta corretta solo in quelle situazioni in cui il giocatore non abbia una strategia vincente (ossia non possa vincere pur assumendo che l'avversario giochi ottimamente). Quando invece il giocatore sia in grado di vincere a gioco ottimo, allora vorremmo che la funzione `play` computasse una mossa vincente e ritornasse il numero di pedine da rimuovere sui due stack (in generale almeno uno dei due numeri ritornati deve essere nullo). Il tuo compito è correggere/completare l'implementazione attuale della funzione `play`. A volte si può vincere!

Dati di input

Il vostro programma riceve come suo input due numeri naturali n_1 ed n_2 che indicano il numero di pedine presenti su ciascuona delle due pile del gioco quando siete chiamati a compiere la vostra mossa.

Dati di output

Dovete ritornare in output la coppia di numeri $(0, 0)$ quando realizzate che la partita è ormai persa, in tutti gli altri casi dovete invece ritornare una codifica di una qualche mossa vincente. La mossa vincente è rappresentata da una coppia $(a, 0)$ se prescrive di rimuovere a pedine dalla prima pila; essa è invece rappresentata da una coppia $(0, b)$ se prescrive di rimuovere b pedine dalla seconda pila. A livello di output dell'intero programma (gestito dal template, vi consigliamo di non modificare il template al di fuori della funzione `play`), tali coppie ordinate vengono di fatto rese come puoi vedere nei seguenti due esempi.

Esempi di input/output

input (da stdin)	output (su stdout)
2 4	0 2

input (da stdin)	output (su stdout)
2 2	0 0

Assunzioni

- $0 \leq n_1, n_2 \leq 10^{100}$.

Subtask

- **Subtask 1 [0 punti]:** gli esempi del testo.
- **Subtask 2 [30 punti]:** $n_1 + n_2 \leq 10$.
- **Subtask 3 [30 punti]:** $n_1, n_2 \leq 100$.
- **Subtask 4 [10 punti]:** $n_1 = 4,\ n_2 \leq 10^{100}$.
- **Subtask 5 [30 punti]:** $n_1, n_2 \leq 10^{100}$.