

## ฟังก์ชันออก

จงเขียนฟังก์ชันต่าง ๆ ข้างล่างนี้ ให้ทำงานตามหน้าที่ที่เขียนใน comment

```
def is_odd(n):
    # คืน (True/False) ว่า n เป็นจำนวนคี่หรือไม่

def has_odds(x):
    # คืน (True/False) ว่า x เป็นลิสต์ที่มีข้อมูลบางตัวเป็นจำนวนคี่

def all_odds(x):
    # คืน (True/False) ว่า x เป็นลิสต์ที่มีข้อมูลทุกตัวเป็นจำนวนคี่

def no_odds(x):
    # คืน (True/False) ว่า x เป็นลิสต์ที่ไม่มีข้อมูลที่เป็นจำนวนคี่เลย

def get_odds(x):
    # คืนลิสต์ที่มีจำนวนคี่ที่มีเก็บในลิสต์ x (ลำดับก่อนหลังให้เป็นไปตามลำดับเดียวกับใน x)
    # เช่น x = [1,2,3,5,0] จะได้ผลคือ [1,3,5]

def zip_odds(a, b):
    # คืนลิสต์ที่สร้างจากการนำจำนวนคี่ใน a และ b มาสลับกันเก็บในลิสต์ผลลัพธ์ (เริ่มจากใน a ก่อน)
    # เช่น a = [0,8,1,2,4,6,5,7,9,2,3] กับ b = [4,19,11,12,10,17] จะได้คือ
    # [1,19,5,11,7,17,9,3]

exec(input().strip()) # ต้องมีคำสั่งนี้ ตรงนี้ ดอนส่งให้ Grader ตรวจ
```

## ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ใช้ทดสอบการทำงานของฟังก์ชัน

## ข้อมูลส่งออก

ผลที่ได้จากการสั่งทำงานคำสั่งที่ได้รับ

## ตัวอย่าง

| input (จากแป้นพิมพ์)                                     | output (ทางจอภาพ)      |
|--|------------------------|
| <code>print(is_odd(31))</code>                           | True                   |
| <code>print(has_odds([0,2,3,4,8]))</code>                | True                   |
| <code>print(all_odds([1,3,11,17]))</code>                | True                   |
| <code>print(no_odds([2,4,8]))</code>                     | True                   |
| <code>print(get_odds([1,3,11,2,17]))</code>              | [1, 3, 11, 17]         |
| <code>print(zip_odds([1,3,11,2,17], [2,4,97,99]))</code> | [1, 97, 3, 99, 11, 17] |
| <code>print(zip_odds([2,4,97,99], [1,3,11,2,17]))</code> | [97, 1, 99, 3, 11, 17] |