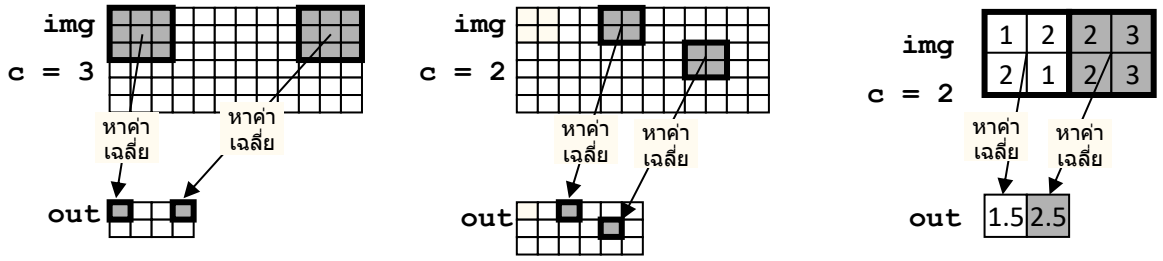
การปรับขนาดภาพ

ฟังก์ชัน **scale(img, c)** รับรูปภาพ **img** ที่เป็น numpy อาเรย์ 2 มิติ เพื่อคืน numpy อาเรย์ 2 มิติ (ให้ชื่อว่า **out**) ที่มีจำนวนแถวและจำนวนหลักลดลง **c** เท่าของ **img** (เช่น **img** มีขนาด 10×15 และ **c = 5** จะได้ **out** ที่มีขนาด 2×3) ให้ถือว่า **c** > 0 และจำนวนแถวและจำนวนหลักของ **img** หารด้วย **c** ได้ลงตัว สำหรับค่าในแต่ละช่องของ **out** นั้นหาได้จากค่าเฉลี่ยของข้อมูลในอาเรย์ย่อยขนาด **c** × **c** ของ **img** ในลักษณะที่แสดงเป็นตัวอย่างข้างล่างนี้




```
import numpy as np

def scale(img, c) :

  ???              # เขียนตรงนี้

def read_img():
  row, col = [int(e) for e in input().split()]
  img = np.ndarray((row,col))
  for i in range(row):
    img[i] = [float(e) for e in input().split()]
  return img

def show_output(out):
  for i in range(out.shape[0]):
    print(" ".join([str(e) for e in out[i]]))

img = read_img()
c = int(input())
out = scale(img, c)
show_output(out)
```

ข้อมูลนำเข้า

บรรทัดแรกเป็นจำนวนเต็ม 2 จำนวน, Row กับ Col บอกจำนวนแถวกับจำนวนคอลัมน์ ตามลำดับ
Row บรรทัดต่อมา หนึ่งบรรทัดแทนข้อมูลในเมทริกซ์หนึ่งแถว ประกอบด้วยจำนวนจริง Col จำนวน
บรรทัดสุดท้ายเป็นจำนวนเต็ม 1 จำนวน แทนค่า c ที่อธิบายข้างต้น

ข้อมูลส่งออก

แสดงเมทริกซ์ผลลัพธ์ของฟังก์ชัน scale จำนวน Row / c บรรทัด
แต่ละบรรทัดแสดงจำนวนจริง Col / c จำนวน แทนค่าในเมทริกซ์ผลลัพธ์

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
1 3 3 2 2 1	3.0 2.0 2.0
3 3 3 2 2 2 0 2 2 2 3 3	2.0
2 4 1 2 2 3 2 1 2 3 2	1.5 2.5
4 6 1 2 2 3 3 2 2 1 2 3 2 3 2 3 2 2 4 2 3 2 2 2 2 4 2	1.5 2.5 2.5 2.5 2.0 3.0