1. Which of the following snippets of Python code are equivalent to

\[
draws = \text{np.random.randint}(0, 2, \text{size}=\text{steps})
\]

where \( \text{steps} \) is an integer \( \geq 0 \) (note \text{np.random.randint} uses exclusive end). All NumPy vectors should be translated to Python lists.

A. \( \text{draws} = \text{randint}(0, 1) \)

B. \( \text{draws} = \text{steps} * \text{randint}(0, 1) \)

C. \( \text{draws} = []
\)
   for \( i \) in range(2):
   \text{draws.append}(\text{randint}(0, \text{steps}))

D. \( \text{draws} = []
\)
   for \( i \) in range(\text{steps}):
   \text{draws.append}(\text{randint}(0, 1))

2. Which of the following snippets of Python code are equivalent to

\[
draws = \text{np.random.randint}(0, 2, \text{size}=\text{steps})
\]

where \( \text{steps} \) is an integer \( \geq 0 \) (note \text{np.random.randint} uses exclusive end). All NumPy vectors should be translated to Python lists.

A. \( \text{draws} = \text{randint}(0, 1) \)

B. \( \text{draws} = \text{steps} * \text{randint}(0, 1) \)

C. \( \text{draws} = []
\)
   for \( i \) in range(2):
   \text{draws.append}(\text{randint}(0, \text{steps}))

D. \( \text{draws} = []
\)
   for \( i \) in range(\text{steps}):
   \text{draws.append}(\text{randint}(0, 1))

Answer: D
The NumPy code creates a vector of length size of random integers that are either 0 or 1. So we want to create a list of length size of similar random integers.
3. Which of the following snippets of Python code are equivalent to:

draws = np.random.randint(0, 2, size=steps)
delta = np.where(draws == 1, 1, -1)
walk = np.cumsum(delta)

where steps is an integer >= 0 (note np.random.randint uses exclusive end). All NumPy vectors should be translated to Python lists.

A  pos = 0
    walk = []
    for i in range(steps):
        if random.randint(0, 1) == 1:
            pos += 1
        else:
            pos -= 1
    walk.append(pos)

B  walk = []
    for i in range(steps):
        if random.randint(0, 1) == 1:
            walk.append(1)
        else:
            walk.append(-1)

C  walk = []
    for i in range(steps):
        walk.append(random.randint(0, 1))

D  pos = 0
    for i in range(steps):
        if random.randint(0, 1) == 1:
            pos += 1
        else:
            pos -= 1
4. Which of the following snippets of Python code are equivalent to:

draws = np.random.randint(0, 2, size=steps)
delta = np.where(draws == 1, 1, -1)
walk = np.cumsum(delta)

where steps is an integer >= 0 (note np.random.randint uses exclusive end). All NumPy vectors should be translated to Python lists.

A
pos = 0
walk = []
for i in range(steps):
    if random.randint(0, 1) == 1:
        pos += 1
    else:
        pos -= 1
    walk.append(pos)

B
walk = []
for i in range(steps):
    if random.randint(0, 1) == 1:
        walk.append(1)
    else:
        walk.append(-1)

C
walk = []
for i in range(steps):
    walk.append(randint(0, 1))

D
pos = 0
for i in range(steps):
    if random.randint(0, 1) == 1:
        pos += 1
    else:
        pos -= 1

Answer: A
draws is a vector of random numbers, while delta is a translation of those numbers into +1 or -1. The cumsum method computes the cumulative sum, i.e. for [a, b, c], it computes [a, a+b, a+b+c]. Answer A implements the latter via the pos accumulator.