Part-B

Code-

**def** undoom\_dice(Die\_A, Die\_B):

*# This function will return a pair of dice that have the same distribution of sums as the original dice.*

**def** all\_pairs(dice):

*# Return all dice pairs (New\_Die\_A, New\_Die\_B) where New\_Die\_A <= New\_Die\_B (i.e., all pairs except (New\_Die\_B, New\_Die\_A)).*

pairs **=** [(A, B) **for** A **in** dice **for** B **in** dice **if** A **<=** B]

**return** pairs

**def** distribution\_of\_sums(Die\_A, Die\_B):

*# All possible sums when the dice are thrown together.*

*# this function returns a sorted list of the distribution of sums.*

A, B **=** Die\_A, Die\_B

sums **=** sorted(a **+** b **for** a **in** A **for** b **in** B)

**return** sums

**def** all\_dice():

*# All possible combinations for the dice*

allComb **=** [[1, b, c, d, e, f] **for** b **in** range(2, 9) **for** c **in** range(b, 9) **for** d **in** range(c, 9) **for** e **in** range(d, 9) **for** f **in** range(e, 9)]

**return** allComb

standard\_pair **=** Die\_A, Die\_B

standard\_sums **=** distribution\_of\_sums(Die\_A, Die\_B)

new\_pair **=** [(A,B) **for** (A,B) **in** all\_pairs(all\_dice()) **if** distribution\_of\_sums(A,B) **==** standard\_sums **and** (A,B) **!=** standard\_pair]

**return** new\_pair

INPUT

Die\_A, Die\_B = [1,2,3,4,5,6], [1,2,3,4,5,6]

New\_Die\_A, New\_Die\_B = undoom\_dice(Die\_A, Die\_B)[0]

print("New\_Die\_A: ", New\_Die\_A)

print("New\_Die\_B: ", New\_Die\_B)

OUTPUT

New\_Die\_A, New\_Die\_B = undoom\_dice(Die\_A, Die\_B)[0]

>

> print("New\_Die\_A: ", New\_Die\_A)

New\_Die\_A: [1, 2, 2, 3, 3, 4]

>

> print("New\_Die\_B: ", New\_Die\_B)

New\_Die\_B: [1, 3, 4, 5, 6, 8]