Exercise 2: Drawing a Red Card from a Deck

Calculate the probability of drawing a red card from a standard deck of 52 playing cards. Provide your solution using the theory of probability and compare with the simulation.

Solution:

Simulate drawing a card from a standard deck of 52 cards 10,000 times to calculate the probability of drawing a red card.

```
import random

def simulate_drawing_red_card(trials=12000):  #TASK:  Modify the number of trials and the situation
    red_card_draws = sum([1 for _ in range(trials) if random.choice(['red', 'black']) == 'red'])
    probability = red_card_draws / trials
    return probability

print(simulate_drawing_red_card())
```

[2] ✓ 0.0s Python

```
0.5
```

Calculate the probability of randomly selecting a prime number when picking a number from 1 to 10 (inclusive). Provide your solution using the theory of probability and compare with the simulation.

Solution

Simulate picking a number from 1 to 10, 10,000 times and calculate the probability of picking a prime number.

```
import random

def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def simulate_picking_prime(trials=1000):  #TASK: Modify the number of trials and the situation
    prime_picks = sum([1 for _ in range(trials) if is_prime(random.randint(1, 10))])
    probability = prime_picks / trials
    return probability

print(simulate_picking_prime())
```

[3] ✓ 0.0s Python

```
thon.exe Untitled-1
C:\Users\ethan\AppData\Local\Programs\Python\Python313\python.exe: can't o
file 'C:\\Users\\ethan\\Untitled-1': [Errno 2] No such file or directory
PS C:\Users\ethan> & C:/Users/ethan/AppData/Local/Programs/Python/Python3
thon.exe c:/Users/ethan/Downloads/new.py
7
14
21
28
35
42
49
PS C:\Users\ethan>
```

```python
for number in range (7 , 50 , 7):
    print ( number )
```

[1] ✓ 0.0s Python

```
7
14
21
28
35
42
49
```

Exercise 1: Rolling a Die

Calculate the probability of rolling a number greater than 4 on a standard six-sided die. Provide your solution using the theory of probability and compare with the simulation.

Solution:

Simulate rolling a die 10,000 times and calculate the probability of rolling a number greater than 4.

```python
import random

def simulate_rolling_die(trials=2000):        #TASK: Modify the number of trials and the situation
    rolls_greater_than_4 = sum([1 for _ in range(trials) if random.randint(1, 6) > 4])
    probability = rolls_greater_than_4 / trials
    return probability

print(simulate_rolling_die())
```

[1] ✓ 0.0s Python

```
0.338
```