



Розглянемо звичайну колоду карт зі **значеннями** карт (в порядку зростання) 2-10, "Jack", "Queen", "King", "Ace" та чотири **мастями** "Hearts", "Clubs", "Diamonds", "Spades".

Завдання 1. Згенерувати колоду карт і записати її у список *cards*. Спосіб представлення кожної карти - рядок. Приклад: "10S", "QD".

В даному проєкті ми будемо вивчати **частоту випадання покерних комбінацій** з 5 карт і перевіримо чи справді сила комбінації залежить від ймовірності її отримання. Далі наведено список комбінацій в порядку зростання їх сили.

"One pair" - комбінація, в якій є дві карти з однаковим значенням. Приклад:
["3S", "10C", "QD", "10D", "7H"]

"Two pair" - комбінація, в якій є дві пари карт однакового значення. Приклад:
["7S", "10C", "QD", "10D", "7H"]

"Three of a kind" - комбінація, в якій є три карти однакового значення. Приклад:
["3S", "3C", "QD", "10D", "3H"]

"Straight" - комбінація, в якій значення карт розташовані в порядку зростання. Приклад:
["3S", "5C", "7D", "4D", "6H"]

"Flush" - комбінація, в якій всі 5 карт мають однакову масть. Приклад:
["3S", "JS", "QS", "10S", "7S"]

"Full house" - комбінація, в якій 2 карти мають однакове значення і 3 інші карти також мають однакове значення. Приклад:
["3S", "10C", "10D", "10H", "3H"]

"Four of a kind" - комбінація, в якій 4 карти мають однакове значення. Приклад:
["3S", "3C", "3D", "10D", "3H"]

Завдання 2. Для кожної комбінації написати функцію, яка приймає список з 5 карт і повертає *True*, якщо список є відповідною комбінацією карт. Приклад:

`three(["3S", "3C", "QD", "10D", "3H"]) → True`

Завдання 3. Дослідити частоту випадання вказаних вище комбінацій. Для цього провести 100000 еспериментів, де в кожному ми обираємо випадковим чином 5 карт (функція `random.sample(list, n)`) і рахуємо скільки разів зі 100000 ми отримали вказані комбінації. Вивести статистику на екран у наступному вигляді:

Number of experiments: 100000

One pair:

Two pair:

Three of a kind:

Straight:

Flush:

Full house:

Four of a kind: