# Pốc cơ(Poker)

## Objectives:

* Luyện tập về cách sử dụng functions (modularity-use functions properly)
* Chọn tên function, tên biến một cách phù hợp
* Luyện tập testing and debugging
* Intro về hướng đi của OOP

## Overview

Tính xác suất xảy ra của các bộ lá bài trong phiên bản Poker đã được đơn giản hoá.

## What is Poker? – The following documentation will be written in English to better obey the correctness of terminologies, naming of functions, variables.

Poker is a card game in which players bet on whether their hands are better than their opponents' hands, without knowing for certain what cards their opponents hold. There are many variations of Poker, but in all of them players have or make 5-card hands, and depending on the cards in the hand, the hand can be worth more or less (or the same) as another 5-card hand.

See cardplayer.com for a list of possible hand kinds and their relative rankings.

The rankings are based on how likely it is to obtain a hand of that type. A royal flush, for example, is worth more than a hand with four-of-a-kind because a royal flush is less likely to be dealt than four- of-a-kind. Similarly, four-of-a-kind is worth more than two pairs because it is more common to be dealt two different pairs than four cards of equal rank. Thus, a high value hand has fewer hands that can beat it. That's why having four-of-a-kind is a pretty good hand (but not a sure thing)!

**A simplification**

In this project, we will group the different hand types into these four categories:

* Flush (includes normal, royal, and straight flushes): tất cả lá bài cùng chất,
* Two pair (includes two pair, four-of-a-kind, and full house): 2 cặp 2 lá cùng rank
* Pair (includes pair and three-of-a-kind): 2 lá cùng ranks
* High card (includes high card and straight). Kiểu vớ vẩn
* Ace has the highest rank and Two has the lowest.

Note that these categories are disjoint. A hand with a straight flush will only be categorized as a flush – it will not also count in the High card category.

🡪 Code được desk và hand, các hàm để kiểm tra hand, chạy 100 nghìn lần bào nhiêu % là các loại hand

Bdien 1 la bài bằng tuple

Card=(rank,suit)

Mhbbnnmm

Random.shuffle

## Your Mission

The goal of your program is to output the following table(Output phải y hệt)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # of hands | pairs | % | 2 | pairs | % | flushes | % | high | card | % |
| 10,000 | xx | xx.xx |  | xx | xx.xx | xx | xx.xx |  | xx | xx.xx |
| 20,000 | xx | xx.xx |  | xx | xx.xx | xx | xx.xx |  | xx | xx.xx |
| 30,000 | xxx | xx.xx |  | xxx | xx.xx | xxx | xx.xx |  | xxx | xx.xx |
| 40,000 | xxx | xx.xx |  | xxx | xx.xx | xxx | xx.xx |  | xxx | xx.xx |
| 50,000 | xxxx | xx.xx |  | xxxx | xx.xx | xxxx | xx.xx |  | xxxx | xx.xx |
| 60,000 | xxxxx | xx.xx | xxxxx | | xx.xx | xxxxx | xx.xx | xxxxx | | xx.xx |
| 70,000 | xxxxx | xx.xx | xxxxx | | xx.xx | xxxxx | xx.xx | xxxxx | | xx.xx |
| 80,000 | xxxxx | xx.xx | xxxxx | | xx.xx | xxxxx | xx.xx | xxxxx | | xx.xx |
| 90,000 | xxxxx | xx.xx | xxxxx | | xx.xx | xxxxx | xx.xx | xxxxx | | xx.xx |
| 100,000 | xxxxx | xx.xx | xxxxx | | xx.xx | xxxxx | xx.xx | xxxxx | | xx.xx |

(x chỉ là giá trị mô phỏng) To accomplish this, you need to:

1. model a deck of cards that can be shuffled(tráo) and from which cards can be dealt
2. shuffle the deck(tráo bộ bài)
3. repeatedly create poker hands by dealing cards from that deck. Once the deck is used up, gather up the cards, reshuffle, and keep dealing.
4. Run 100000 times. Each time through 10000, print!

## Formatting

Your output should match the sample output above. That means:

* If a percentage is less than 10%, you need a leading zero (e.g. 02.53 instead of 2.53).
* The raw counts should be **right-aligned** and should not have any leading zeroes. Ignore the number of x's for the raw counts in the picture above. It can have more digits, not just xx (Just an example).
* The total number of hands analyzed should be displayed with a comma(e.g. 10,000 instead of 10000) but the other columns should not have any commas.
* Width không nhất thiết phải match

## Requirements

You are required to model a deck of cards with a list of elements, where each element represents a card. But what each element actually \*is\* will be up to you – string, tuple, integer

* or whatever. For example, you could make each card a string like "5D" for the five of diamonds. But what will you do about the ten of diamonds? "10D"? "TD"? And what about the Jack, Queen, King, and Ace? Remember, what seems the easiest right now may not be so easy to work with later. **Be willing to refactor**. And if you do, write a comment at the top of that file explaining why you refactored the way you did. And if you never do, then write a comment explaining what functions were easy to write because of the implementation you chose and what functions would've been easier to write had you chosen something else (and what that alternative implementation would've been). This self-reflection is important so spend some time thinking about this.

You are required to have the following functions with the following behaviors. Exactly how you implement these is up to you, including what the *signature* (that is, the function name and parameter list) and return types should be:

* + a function to create a standard deck of 52 cards
  + a function to shuffle a deck of cards
  + a function to deal from a deck of cards

## FAQ

## Làm thế nào để tạo một Poker Hand?

Một hand cũng là một list of cards, giống như deck nhưng nhỏ hơn. Một hand có 5 lá, 1 deck có 52 lá.

## Làm thế nào để tìm ra kiểu lá bài(Kind of Hand)?

That depends on your implementation, but you'll definitely need a way to get a card's suit separately from its rank. For flushes, you want to see if all the suits in the hand are the same. For the other hand types, you care about ranks.

## Có nhất thiết phải cần multiple .py files?

Nên nhớ, file nào có cùng một logic thì để một file. Main để riêng, các file còn lại cũng phải để riêng, xong rồi hẵng import

## Cần viết file test không?

Có. Test as you go. Đừng viết code xong hết rồi mới test!!. Đấy gọi là crappy coding, crappy coder. “Code a little, test a little”. Test each function before writing the next one(think of real cases that can happen first, then CODE). Use the if name == " main " trick so you can keep your tests around without them causing interference.