

**Задача:** разработать api для расчета количества набранных очков победителем и типа его победы (оин, марс, кокс) для коротких нард. Формат входных и выходных данных - ниже.

### Определения:

*оин* - положение шашек в конце игры, при котором проигравший успел выбросить хотя бы 1 шашку, в то время как оппонент вывел все за доску. Считается как 1 очко

*марс* - положение шашек в конце игры, при котором проигравший не успел завести в свой дом все свои шашки, в то время как оппонент вывел все за доску. Такая победа приносит 2 очка

*кокс* - ситуация, при которой проигравший не успел вывести одну или несколько своих шашек из первого квадранта или оставил шашку на баре, в то время как оппонент вывел все свои за доску

**Используемые технологии:** Python 3.9+. Остальной набор фреймворков - на ваше усмотрение (предпочтительнее - FastAPI + pydantic).

### Формат входных данных (pydantic):

```
class GameResultInput(BaseModel):
    board: Board
    start_position: Dict[UUID, conint(ge=0, le=23)] # Key - UUID is the player's
    identifier. Value - board position number starting from 0 to 23 (24 positions total)

class Board(BaseModel):
    bar_counts: Dict[UUID, int] # UUID is the player's identifier, int - checkers count for
    the given player on the bar
    points: List[BoardPoint]

class BoardPoint(BaseModel):
    number: conint(ge=0, le=23)
    checkers_count: conint(ge=0, le=15)
    occupied_by: Optional[UUID] = None # None if the point is not occupied
```

Copy

### Формат выходных данных (pydantic):

```
class GameWinType(str, Enum):
    Oin = 'oin'
    Mars = 'mars'
    Koks = 'koks'

class GameResultOutput(BaseModel):
    points: conint(ge=0)
    win_type: GameWinType
```

Copy

**Формат приема работы:** ссылка на git-репозиторий. Readme с инструкцией - обязательно