# Задача о прокате оборудования

## Вводим параметры

print("Ввод параметров для задачи о прокате оборудования:\n")  
p1 = float(input("вероятность отстаться в рабочем состоянии при нормальной эксплуатации, p1 > p2: "))  
p2 = float(input("вероятность отстаться в рабочем состоянии при жесткой эксплуатации: "))  
q1 = float(input("вероятность не починить машину за единицу времени при нормальном ремонте: "))  
q2 = float(input("вероятность не починить машину за единицу времени при быстром ремонте, q1 < q2: "))  
  
c11 = float(input("доход в рабочем состоянии при нормальной эксплуатации, c11 < c12: "))  
c12 = float(input("доход в рабочем состоянии при жесткой эксплуатации: "))  
c21 = float(input("стоимость нормального ремонта, c21 < c22: "))  
c22 = float(input("стоимость быстрого ремонта: "))  
  
# Проверка соответствия введенных параметров условию задачи  
if not (p1 > p2):  
 print("Condition p1 > p2 is not satisfied.")  
   
if not (q1 < q2):  
 print("Condition q1 < q2 is not satisfied.")  
   
if not (c11 < c12):  
 print("Condition c11 < c12 is not satisfied.")  
   
if not (c21 < c22):  
 print("Condition c21 < c22 is not satisfied.")

Ввод параметров для задачи о прокате оборудования:  
  
Condition q1 < q2 is not satisfied.

### Вычисляем стационарное распределение

def stationary\_distribution(p\_i, q\_j):  
 denominator = 2 - p\_i - q\_j  
 pi\_1 = (1 - q\_j) / denominator  
 pi\_2 = (1 - p\_i) / denominator  
 return pi\_1, pi\_2

### Вычисляем средний доход

def average\_revenue(pi\_1, pi\_2, c\_1i, c\_2j):  
 return pi\_1 \* c\_1i - pi\_2 \* c\_2j

def main():  
  
 strategies = []  
 # Стратегия f11: нормальная эксплуатация, нормальный ремонт  
 pi1, pi2 = stationary\_distribution(p1, q1)  
 avg\_revenue = average\_revenue(pi1, pi2, c11, c21)  
 strategies.append(('f11', avg\_revenue))  
  
 # Стратегия f12: нормальная эксплуатация, быстрый ремонт  
 pi1, pi2 = stationary\_distribution(p1, q2)  
 avg\_revenue = average\_revenue(pi1, pi2, c11, c22)  
 strategies.append(('f12', avg\_revenue))  
  
 # Стратегия f21: жесткая эксплуатация, нормальный ремонт  
 pi1, pi2 = stationary\_distribution(p2, q1)  
 avg\_revenue = average\_revenue(pi1, pi2, c12, c21)  
 strategies.append(('f21', avg\_revenue))  
  
 # Стратегия f22: жесткая эксплуатация, быстрый ремонт  
 pi1, pi2 = stationary\_distribution(p2, q2)  
 avg\_revenue = average\_revenue(pi1, pi2, c12, c22)  
 strategies.append(('f22', avg\_revenue))  
  
 # Какая из них самая оптимальная  
 optimal\_strategy = max(strategies, key=lambda x: x[1])  
  
 # Вывод  
 print("\nСредний доход для каждой стратегии:")  
 for strat, revenue in strategies:  
 print(f"Стратегия {strat}: приносит средний доход = {revenue:.4f}")  
  
 print(f"\nОптимальная стратегия: {optimal\_strategy[0]} приносящая {optimal\_strategy[1]:.4f}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Средний доход для каждой стратегии:  
Стратегия f11: приносит средний доход = 72.0000  
Стратегия f12: приносит средний доход = 69.0909  
Стратегия f21: приносит средний доход = 134.5455  
Стратегия f22: приносит средний доход = 132.5000  
  
Оптимальная стратегия: f21 приносящая 134.5455