Zadanie 1. Przypomnijmy zadanie o więźniach z ostatnich ćwiczeń nie w pracowni. Mieliśmy 10 więźniów (ponumerowanych od 1 do 10) oraz 10 skrzyń. Skrzynie były ustawione w rzędzie (tak że można mówić o pierwszej skrzyni, drugiej skrzyni...) i znajdowały się w nich numery więźniów, aczkolwiek w losowym porządku. Każdy z więźniów miał za zadanie znaleźć skrzynię ze swoim numerem, jednak w tym celu mógł sprawdzić zawartość tylko 5 skrzyń. Były też tu założenia w stylu brak komunikacji między więźniami, brak możliwości zmiany zawartości skrzyń czy ich przestawiania, niemniej mam nadzieję, że pamiętacie problem.

Tak czy siak, zadanie polega na sprawdzeniu kilku strategii, które mogli przyjąć więźniowie, tj. policzyć dla każdej z tych strategii liczbę takich permutacji, że przy danej strategii wszyscy więźniowie znajdowali swoje numery.

Przykładowe strategie:

- ⋆ więzień o numerze i sprawdza pięć kolejnych skrzyń, począwszy od skrzyni i-tej
- ⋆ więzień o numerze i sprawdza co drugą skrzynię, począwszy od skrzyni i-tej (zamiast co drugą może być co trzecią, czwartą... a może być nawet co i-tą, czyli pierwszy więzień sprawdza skrzynie po kolei, drugi więzień – co drugą, trzeci – co trzecią, itd.)
- ⋆ więzień o numerze i zagląda najpierw do skrzyni o numerze i i jeśli był w niej numer j, to sprawdza następnie j-tą skrzynię, a jak znajdzie w niej numer k, to... itd. (to na dobrą sprawę policzyliśmy na zajęciach, ale tutaj możemy się przekonać „ręcznie", że rzeczywiście ta strategia jest naprawdę dobra)
- ⋆ więzień o numerze i sprawdza pięć kolejnych skrzyń począwszy od i, przy czym jeśli i jest parzyste, to sprawdza skrzynie na prawo od i-tej, a jeśli i jest nieparzyste, to sprawdza skrzynie na lewo od i-tej

Polecam sprawdzić wszystkie strategie i porównać je ze strategią losowego wybierania, która daje średnio $\frac{1}{2^{10}} \cdot 10! = 3543,75$ dobrych permutacji (na $10! = 3628800$ wszystkich możliwych).

W zadaniu przyda się biblioteka *itertools*, która pozwoli nam na wygodne przejrzenie wszystkich permutacji. Wygodnie jest również założyć, że więźniowie są ponumerowani od 0 do 9, tak samo jak skrzynie (czyli jest skrzynia zerowa, pierwsza, ..., aż do dziewiątej). Przykład znajdziecie pod tym linkiem.

Zadanie 2. Okazuje się, że gdybyśmy chcieli sprawdzić, co się dzieje w przypadku większej grupy więźniów, np. 100 więźniów/100 skrzyń/50 sprawdzeń, a nawet dla 12 więźniów/12 skrzyń/6 sprawdzeń, to się okaże, że zliczenie tego wszystkiego okaże się w krótkim czasie na naszym sprzęcie niewykonalne. Możliwym obejściem problemu byłoby znalezienie *przybliżonej* liczby dobrych permutacji. W tym celu można sprawdzić, jak działa dana strategia więźniów na próbie np. N = 10 000 *losowych* permutacji. Jeśli wśród tych N permutacji naliczyliśmy jakieś K dobrych, to można oczekiwać, że $\frac{K}{N}$ będzie przybliżeniem prawdopodobieństwa, że losowo wybrana permutacja będzie dobra, natomiast $\frac{K}{N} \cdot n!$ będzie przybliżoną liczbą dobrych permutacji (przy czym małe n to liczba więźniów/skrzyń). Główny problem polega na tym, *jak duże* należy przyjąć N, aby wyszedł wynik z dobrym przybliżeniem? To już trzeba ustalić doświadczalnie.

Zadanie tak czy siak polega na sprawdzeniu powyższych strategii dla większej liczby więźniów (12, 50, 100, może nawet 200?). Warto też sprawdzić, jakie otrzymamy wyniki dla 10 więźniów

(które policzyliśmy dokładnie w Zadaniu 1.), żeby zobaczyć, na ile dokładne wyniki daje ta metoda.

W tym zadaniu przyda się z kolei biblioteka *random*, która pozwoli nam na generowanie losowych permutacji. Przykład znajdziecie pod tym linkiem.