1. (za 1.5 punktu) Zdefiniuj abstrakcyjny typ danych dla artymetyki modulo liczba pierwsza p. Reprezentaja ma zawierać zarówno wartość jak i p. Operacje dwuargumentowe mają działać wtedy gdy p dla obu argumentów jest takie same, różne p w operacji dwuargumentowej traktujemy jako błąd. Typ ma pozwalać na operacje arytmetyczne, tworzenie liczb, odczyt n i p i wypisywanie. To ma być komplet operacji, innych nie ma być. Przy tworzeniu na wiarę przyjmujemy że p jest pierwsze (sprawdzanie pierwszości byłoby zbyt czasochłonne). Odwrotność n modulo p (potrzebne do dzielenia) można obliczać przy pomocy rozszerzonego algorytmu Euklidesa. Dokładniej roszerzony algorytm Euklidesa produkuje liczby a, b i g takie że g to największy wspólny dzielnik n i p (w naszym przypadku 1 bo zakładamy że p jest liczbą pierwszą) oraz $g = an + bp$. We efekcie a daje odwrotność n modulo p.

2. (za 1.5 punktu) Zdefiniuj abstrakcyjny typ danych dla operacji na permutacjach. Typ ma pozwalać na składanie permutacji (operator mnożenia), obliczanie odwrotności, obliczanie znaku permutacji, tworzenie permutacji z wektora (konstruktor) i wypisywanie. Ponato zaimplementuj funkcję podającą największy element zmieniany przez permutację (dla permutacji identycznościowej przyjmujemy że jest to -1).

3. (za 1 punkt) Zdefinuj klasę do reprezentacji dat z zakresu lat 1901-2099 (taki zakres by rok był przestępny wtedy i tylko wtedy gdy jego numer dzieli się przez 4). Klasa ma mieć konstruktor w którym podajemy rok, miesiąc i dzień, ten konstruktor ma sygnalizować błąd dane nie są poprawne. Klasa ma mieć przedrostkowy operator `++` który daje następny dzień i operatory porównania odpowiadające porządkowi kalendarzowemu. Zdefinuj operator `<<` wypisujący w postaci rok.miesiąc.dzień, np `2025.4.24`.