

1. Zmodyfikuj funkcję obliczającą największy wspólny dzielnik z przykładu do zajęć 1, tak by działała dla wielomianów z zadania 2 z listy 11 (0.5 punktu). Użyj algorytm potęgowania binarnego by obliczyć $P^n \bmod Q$ dla dużego n (co najmniej `__uint128_t`) oraz wielomianów P i Q (uwaga: każdą operację z osobna należy robić modulo Q , obliczanie P^n przed obliczaniem reszty z dzielenia byłoby zbyt kosztowne) (0.5 punktu).

2. Zdefiniuj klasy dla symboli i skanera. Symbole to znaki operacji arytmetycznych, nawiasy, liczby, nazwy zmiennych i specjalny symbol końca. Skaner inicjujemy łańcuchem znaków. Skaner ma mieć metodę `daj_symbol` która zwraca następny symbol z łańcucha znaków. Po dojściu do końca łańcucha skaner zwraca symbol końca. Ten opis pozostawia niektóre decyzje do wyboru, opisz w komentarzu swoje wybory.

3. (za 2 punkty) Zdefiniuj typ realizujący arytmetykę zmiennopozycyjną wielokrotnej precyzji. Liczba zmiennopozycyjna to liczba wymierna postaci $b^e m$ gdzie m to liczba całkowita o rozmiarze (ilości cyfr lub bitów) ograniczonym przez wymaganą precyzję zaś b to podstawa (typowo 2 lub 10). Arytmetykę wykonujemy zgodnie z regułami dla arytmetyki liczb wymiernych a następnie zaokrąglamy wynik tak by w postaci wyżej uzyskać m o rozmiarze nie przekraczającym zadanej wartości (precyzji). Precyzja (w cyfrach lub bitach) ma być polem klasy z atrybutem `static`. Zaokrąglenie wystarczy zrealizować przez odrzucenie mniej znaczących bitów lub cyfr. Wykładnik e wystarczy pamiętać w liczbie np. typu `int64_t`. m należy reprezentować przy pomocy klasy realizującej arytmetykę liczb całkowitych wielokrotnej precyzji (można użyć własną klasę, można użyć klasę z istniejącej biblioteki np. GMP).

Komentarz: Dzielenia można zrealizować przy pomocy dzielenia całkowitego (co wymaga tej operacji w klasie realizującej arytmetykę liczb całkowitych) lub przy pomocy wariantu metody Newtona. Dokładniej, jeśli $z_n = \frac{1+\varepsilon}{x}$ i zdefiniujemy $z_{n+1} = 2z_n - xz_n^2$ to $z_{n+1} = \frac{1-\varepsilon^2}{x}$. A więc jeśli wystartujemy z przybliżenia z $|\varepsilon| < \frac{1}{2}$ to kolejne przybliżenia podwajają dokładność mierzoną ilością cyfr. Pierwsze przybliżenie można wyliczyć używając arytmetykę dla typów całkowitych C++.