

# Wstęp do informatyki

## Lista 4

Uwagi:

- Programy/funkcje stanowiące rozwiązania poniższych zadań powinny być napisane w języku C lub Python i poprzedzone prezentacją **idei rozwiązania** (na przykład przy pomocy pseudokodu). Należy również przeanalizować **złożoność** czasową i pamięciową. Staraj się, aby złożoność Twojego rozwiązania była jak **najmniejsza**!
- Jeśli to konieczne, doprecyzuj specyfikację problemu sformułowanego w zadaniu (Twój program powinien być zgodny z ostateczną specyfikacją).
- W programach rozwiązujących zadania możesz korzystać z operatorów  $+$   $-$   $/$   $\%$   $*$  oraz  $//$  (ostatni dotyczy Pythona). Nie korzystaj z innych operatorów arytmetycznych ani z dostępnych funkcji realizujących inne operacje.

1. Napisz program (lub funkcję), który

- a. [0.5] dla podanej liczby naturalnej  $n$  wyznacza

$$(-1)^n \cdot n$$

- b. [0.5] dla podanej liczby naturalnej  $n$  wyznacza

$$\sum_{i=1}^n \frac{(-1)^i}{i}$$

- c. [1] dla podanych liczb naturalnych  $n, x$  wyznacza

$$\sum_{i=1}^n ix^i$$

Oszacuj złożoność czasową swojego rozwiązania. Postaraj się, aby złożoność czasowa była jak najmniejsza.

*Uwaga:* w tym zadaniu nie musisz dbać o prawidłowy sposób „formatowania” wyniku wypisywanego na standardowym wyjściu.

2. [1] Napisz:

- program/funkcję, który oblicza najmniejszą wspólną wielokrotność dwóch liczb naturalnych;
- program/funkcję, który dla dwóch liczb naturalnych  $a, b$  wyznacza taką postać ułamka  $a/b$ , której nie można uprościć (tzn. licznik i mianownik nie mają wspólnego dzielnika większego od 1).

Postaraj się, aby złożoność czasowa Twojego rozwiązania była jak najmniejsza.

*Wskazówka:* skorzystaj z rozwiązań innych problemów poznanych w ramach wstępu do informatyki

3. [2] Napisz program/funkcję realizujący poniższą specyfikację

*Wejście:*  $n$  – dodatnia liczba całkowita,  $a_1, \dots, a_n$  – liczby naturalne

*Wyjście:*  $\text{nwd}(a_1, \dots, a_n) = \max \{ b : b \mid a_i \text{ dla każdego } i \in \{1, \dots, n\} \}$ .

W swoim rozwiązaniu możesz korzystać z funkcji wyznaczającej największy wspólny dzielnik dwóch liczb. Uzasadnij poprawność i oszacuj złożoność czasową swojego rozwiązania.

4. [1] Napisz program/funkcję realizujący poniższą specyfikację:

*Wejście:*  $n, k$  – dodatnie liczby całkowite,  $a_1, \dots, a_k$  – liczby naturalne większe od 1

*Wyjście:*

- $p$  – największa liczba naturalna taka, że  $n$  jest podzielne przez  $a_i^p$  dla jakiegoś  $i \in \{1, 2, \dots, k\}$
- liczby ze zbioru  $\{a_1, \dots, a_k\}$  których  $p$ -ta potęga jest dzielnikiem  $n$ .

Uzasadnij poprawność i oszacuj złożoność czasową swojego rozwiązania.

Przykład:

*Wejście:*  $n = 63\,000 = 2^3 \cdot 3^2 \cdot 5^3 \cdot 7$ ,  $k = 3$ ,  $a_1 = 2$ ,  $a_2 = 3$ ,  $a_3 = 5$ .

*Wyjście:* 3, 2, 5 (czyli  $p = 3$  oraz  $n$  dzieli się przez  $2^3$  i  $5^3$ ).

5. [1] Palindromem binarnym będziemy nazywać liczbę która w zapisie binarnym ma tę własność, że „czytana od końca” jest równa liczbie oryginalnej; np. 101, 110011, 101101. Napisz program/algorytm, który sprawdza czy zapis binarny podanej na wejściu liczby  $n$  jest palindromem binarnym.
6. [1] Analogicznie do palindromu binarnego zdefiniuj pojęcie palindromu  $k$ -arnego odpowiadającego reprezentacji liczby w systemie pozycyjnym o podstawie  $k$  dla liczby naturalnej  $k > 1$ . Pokaż jak zmodyfikować Twój program sprawdzający czy liczba jest palindromem binarnym tak, aby sprawdzał czy  $n$  jest palindromem  $k$ -arnym. Sprecyzuj specyfikację problemu rozwiązywanego przez Twój program/algorytm oraz podaj złożoność czasową Twojego rozwiązania.
7. [1] Mówimy, że liczba ma zapis 1-cyfrowy jeśli w jej zapisie dziesiętnym wykorzystywana jest tylko jedna cyfra, np. 77, 999, 3333; podobnie zapis 2-cyfrowy mają liczby, w których zapisie dziesiętnym występują dokładnie dwie cyfry, np. 322, 79797; itd. Napisz algorytm/program, który dla podanej na wejściu liczby naturalnej  $n$  wypisuje na wyjściu takie  $k$ , że  $n$  ma zapis  $k$ -cyfrowy.

Łatwiejsza wersja zadania:

[0,5] Napisz program, który dla podanej na wejściu liczby naturalnej  $n$  ustala i podaje czy  $n$  ma zapis 1-cyfrowy.

8. [1] Liczby naturalne  $n$  i  $m$  są *podobne* jeśli w ich zapisach dziesiętnych każda cyfra występuje taką samą liczbę razy. Na przykład 123412 i 223411 są podobne natomiast
- 123412 i 11234 nie są podobne z powodu innej liczby wystąpień cyfry 2 w ich zapisach dziesiętnych,
  - 123412 i 1234512 nie są podobne z powodu innej liczby wystąpień cyfry 5 w ich zapisach dziesiętnych.

Napisz algorytm/program, który dla podanych na wejściu liczb  $n$  i  $m$  zwraca informację o tym czy  $n$  i  $m$  są podobne.

**Zadania dodatkowe, nieobowiązkowe (nie wliczają się do puli punktów do zdobycia na ćwiczeniach, punktacja została podana tylko jako informacja o trudności zadań wg wykładowcy)**

9. [0] Pokaż, że  $\log n + \log m = O(\log(n \cdot m))$  oraz  $\log(n \cdot m) = O(\log(n + m))$ .
10. [1] Napisz program, który sprawdza czy zapis binarny podanej na wejściu liczby  $n$  jest palindromem binarnym. Złożoność pamięciowa Twojego rozwiązania powinna być  $O(1)$ , natomiast czas obliczeń  $O(k)$ , gdzie  $k$  to długość zapisu binarnego liczby podanej na wejściu.