

# Algorytmy i struktury danych

## Laboratorium 1

mgr inż. Bartłomiej Kizielewicz, mgr inż. Andrii Shekhovtsov

Listopad 2022

### 1 Zasady oceniania

Oceny będą wystawiane na podstawie sprawozdań wykonywanych z każdej instrukcji. Sprawozdanie ma zawierać imię i nazwisko studenta wraz z rozwiązaniami poszczególnych zadań przeznaczonych do wykonania w domu. Sprawozdania proszę przysyłać w formacie **pdf** za pośrednictwem platformy **moodle**.

Ocena za sprawozdanie równa się liczbie poprawnie wykonanych zadań. W niektórych przypadkach mogą być uwzględnione rozwiązania częściowe.

**UWAGA:** Termin oddania zadania jest ustawiony w systemie moodle. W przypadku nie oddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Zadania oddawane później niż miesiąc po terminie ustawionym na moodle są oddawane i rozliczane w trybie indywidualnym na zajęciach lub po umówieniu się z prowadzącym.

**UWAGA:** W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie `.zip` lub `.pdf` tam, gdzie był wymagany plik tekstowy z rozszerzeniem `.py`.

### 2 Zadania do wykonania na zajęciach

1. Narysuj schemat blokowy algorytmu który pobiera od użytkownika liczbę i sprawdza czy ta liczba jest parzysta. Napisz pseudokod tego algorytmu.
2. Narysuj schemat blokowy algorytmu wyliczającego poniższą sumę (liczbę  $n$  musi podać użytkownik):

$$wynik = \sum_{i=1}^n i^2.$$

3. Napisz pseudokod algorytmu który sprawdzi czy podane poprzez użytkownika 3 liczby mogą być bokami trójkąta (*Możemy zbudować trójkąt z 3 odcinków jeżeli każde 2 z nich są większe od trzeciego*).
4. Narysuj schemat blokowy algorytmu który znajdzie największą oraz najmniejszą liczby podanego przez użytkownika ciągu (listy). Dla uproszczenia "wczytujemy" wartości za pomocą jednego bloczka wprowadzenia danych wejściowych.
5. Napisz pseudokod algorytmu obliczający sumę cyfr w podanej przez użytkownika liczbie  $n$ . Przykład poniżej:

$$n = 1234; \quad wynik = 1 + 2 + 3 + 4.$$

6. Napisz pseudokod który zliczy ile liczb naturalnych jest podzielne przez 3 w przedziale od 4 do 29 włącznie.
7. Narysuj schemat blokowy algorytmu rozwiązywania równania kwadratowego w postaci  $ax^2 + bx + c = 0$ . Wartości  $a, b, c$  są podawane przez użytkownika. Proszę uwzględnić sytuację w których delta jest dodatnia, ujemna i zerem. Zakładamy że zawsze są wprowadzone poprawne współczynniki ( $a$  zawsze jest różne od zera).

### 3 Zadania do wykonania w domu

1. Napisz pseudokod algorytmu który pobiera od użytkownika liczbę i sprawdza czy ta liczba jest podzielna przez 3 lub 5.
2. Narysuj schemat blokowy algorytmu który obliczy sumę pierwszych  $n$  wyrazów ciągu arytmetycznego zaczynającego się od wartości  $a_0$  i różnicą  $r$ . Parametry ciągu  $a_0$ ,  $r$  i  $n$  musi wprowadzić użytkownik. Nie należy używać wzoru na sumę ciągu, lecz zapisać algorytm w postaci pętli.
3. Narysuj schemat blokowy algorytmu który znajdzie sumę i iloczyn podanego przez użytkownika ciągu (listy). Dla uproszczenia "wczytujemy" wartości za pomocą jednego bloczka wprowadzenia danych wejściowych.
4. Narysuj schemat blokowy algorytmu rozwiązującego równanie liniowe  $ax + b = 0$ , liczby  $a$  i  $b$  podaje użytkownik. Podaj również ten algorytm w postaci pseudokodu. Proszę pamiętać o uwzględnieniu trzech przypadków (jedno rozwiązanie, nieskończenie wiele rozwiązań, brak rozwiązań).
5. Napisz pseudokod algorytmu który sprawdzi czy podana poprzez użytkownika liczba  $n$  jest liczbą pierwszą. *Liczba pierwsza to taka liczba naturalna większa od 1, która ma dokładnie dwa dzielniki naturalne: jedynkę i siebie samą.*

## 4 Zadania dodatkowe

Są to zadania nieobowiązkowe które być może będą przerobione w ramach zajęć, w zależności od tego jak szybko będzie przerabiany materiał.

1. Napisz pseudokod algorytmu który: pobiera od użytkownika liczbę  $a$ , na następnie sprawdza czy kwadrat tej liczby jest liczbą parzystą. Jeżeli tak to należy wypisać tę liczbę podzieloną przez dwa, a jeżeli nie to tę liczbę pomnożoną razy 2.
2. Narysuj schemat blokowy algorytmu który pobierze od użytkownika długość trasy w kilometrach  $S$  oraz prędkość poruszania się  $V$ , oraz obliczy i wyświetli ile czasu zajmie pokonanie trasy  $S$  z prędkością  $V$ . W przypadku gdy zostanie wprowadzona nieprawidłowa wartość  $S$  lub  $V$  (np. wartości ujemne lub zero), należy ponownie zapytać się użytkownika o wprowadzenie poprawnych wartości. Napisz też pseudokod tego algorytmu.

3. Narysować schemat blokowy algorytmu który pobierze od użytkownika dwie liczby  $a$  i  $b$ , a następnie obliczy pole i obwód prostokąta o takich stronach. Wynik należy wyświetlić.

W przypadku gdy użytkownik poda jako liczbę  $a$  lub  $b$  wartość ujemną lub zero, należy wyświetlić komunikat o źle wprowadzonych danych i poprosić o ponowne wprowadzenie liczb  $a$  i  $b$ .

Napisać również pseudokod algorytmu.

4. Narysować schemat blokowy algorytmu, który pobierze od użytkownika liczbę, a następnie obliczy pierwiastek kwadratowy tej liczby korzystając z algorytmu Babilońskiego. W przypadku wprowadzenia liczby ujemnej wypisać komunikat i zakończyć działanie algorytmu. Napisz również pseudokod tego algorytmu. Opis metody:

$$x_0 = \frac{S}{2}$$

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{S}{x_n} \right)$$

W powyższych równaniach  $x_0$  to przybliżenie początkowe (przyjmujemy połowę liczby),  $x_{n+1}$  jest kolejnym przybliżeniem wartości  $\sqrt{S}$ . Kończymy algorytm jeżeli  $|x_n - x_{n+1}| \leq 0.1$ . Więcej o tym algorytmie można przeczytać na Wikipedia [https://pl.wikipedia.org/wiki/Metody\\_obliczania\\_pierwiastka\\_kwadratowego](https://pl.wikipedia.org/wiki/Metody_obliczania_pierwiastka_kwadratowego).

5. Napisz pseudokod algorytmu który: pobiera od użytkownika listę wartości. Następnie należy sprawdzić czy wszystkie pobrane wartości są parzyste. Należy wypisać stosowny komunikat (tak, wszystkie parzyste lub nie, są wartości nieparzyste). Proszę użyć pętli do wykonania tego zadania.
6. Napisz pseudokod algorytmu który sprawdzi czy podana przez użytkownika lista wartości jest posortowana w kolejności rosnącej.
7. Napisz kod algorytmu który wykona scalanie dwóch posortowanych list (patrz przykład niżej). Listy proszę zdefiniować oddzielnie, wartości w nich muszą być posortowane.

```
1 a = [1, 2, 5, 7]
2 b = [3, 4, 6]
3
4 wynik: [1, 2, 3, 4, 5, 6, 7]
```