

# Wstęp do programowania w języku C

## Lista zadań 4 (ver. 2)

1. (10/5) Zbiór liczb całkowitych może być zapisany w tablicy o elementach typu `int`, gdzie elementy zbioru są posortowane rosnąco, a na początku tablicy zapisana jest liczba jego elementów. Zakładamy, że stała `MAX_SET_SIZE=8000` określa maksymalną liczbę elementów w zbiorze. Zdefiniować za pomocą `typedef` nowy typ reprezentujący takie zbiory. Następnie zdefiniować dla tego typu operacje (jako oddzielne funkcje): (1) czytania zbioru, (2) wypisywania zbioru, (3) sumy zbiorów, (4) różnicy symetrycznej zbiorów, (5) opróżniania zbioru oraz (6) dodawania do zbioru pojedynczego elementu (jeśli go tam nie ma) i (7) usuwania ze zbioru pojedynczego elementu (jeśli tam jest). Zadbać o to, aby liczba operacji, którą wykonują te funkcje była proporcjonalna do wielkości tych zbiorów (a nie do stałej `MAX_SET_SIZE`). Zakładamy, że zbiór  $A = \{a_1, \dots, a_n\}$ , gdzie  $a_1 < \dots < a_n$  podany jest na wejściu jako ciąg liczb  $n, a_1, \dots, a_n$  oddzielonych białymi znakami. Funkcja czytania zbioru powinna sprawdzać, czy podano poprawnie dane i sygnalizować błąd odpowiednim komunikatem. Sformować z nich moduł i stworzyć plik nagłówkowy. Następnie napisać drugi moduł zawierający w funkcji `main` zestaw wywołań testujący poprawność zdefiniowanych funkcji.
2. (10/10) Zdefiniować za pomocą `typedef` nowy typ reprezentujący liczbę wymierną jako `long long int`, w której licznik i mianownik są zapisane na odpowiednich połowach ciągu bitów. Licznik i mianownik zapisywane w tym typie nie powinny mieć nietrywialnego wspólnego dzielnika, a mianownik powinien być nieujemny. Dla nowego typu zdefiniować funkcje wykonujące cztery podstawowe działania oraz operacje czytania i wypisywania liczby wymiernej. Do ich sformułowania przydatny będzie konstruktor (funkcja, która tworzy liczbę wymierną z licznika i mianownika) oraz dwie funkcje wyodrębniające odpowiednio licznik i mianownik z liczby wymiernej. Wprowadzane i wypisywane liczby wymierne powinny mieć postać  $n/m$ . Sformować z nich moduł i stworzyć plik nagłówkowy. Następnie napisać drugi moduł zawierający funkcję `main` testującą poprawność zdefiniowanych funkcji.
3. (10/10) Napisać program, który rozwiązuje zadanie opisane w Moodlu jako *Lista 4 zadanie 3*. Rozwiązanie tego zadania będzie sprawdzane automatycznie przez sprawdzarkę Moodlową.

15 listopada 2021

Marek Piotrów