

# Wstęp do programowania w języku C

## Lista zadań 6 (ver. 1)

1. (10/5) Kolejka to struktura, w której nowe elementy dopisywane są na końcu, a pobierane z niej elementy pochodzą z początku kolejki (podobnie jak w kolejce sklepowej). W kolejce dwustronnej dopisywać i pobierać można z obu końców kolejki. Do implementacji kolejki dwustronnej w tablicy potrzebne są dwa wskaźniki - jeden wskazujący początek kolejki, a drugi - jej koniec. Tablica powinna być wykorzystywana cyklicznie, jeśli na jej początku lub końcu zwolni się miejsce. Zdefiniować czteroelementową tablicę wskaźników (lub odpowiednią strukturę, jeśli ktoś wie jak) reprezentującą kolejkę liczb typu `double` (dodatkowe dwa wskaźniki powinny wskazywać początek i koniec tablicy) i funkcje: (1) inicjalizacji kolejki (2) opróżniania kolejki, (3) dopisywania liczby na początek kolejki, (4) dopisywania liczby na koniec kolejki, (5) pobierania liczby z początku kolejki, (6) pobierania liczby z końca kolejki oraz (7) sprawdzania, czy kolejka jest pusta. Utworzyć z nich moduł z odpowiednim plikiem nagłówkowym. Program główny powinien umożliwić testowanie operacji na kolejce.
2. (10/10) Zdefiniować moduł służący do testowania sprawności różnych algorytmów sortowania liczb typu `double`. Moduł powinien mieć generator pseudolosowych ciągów o zadanej w parametrze długości (użyć funkcji `rand` z biblioteki `stdlib` do otrzymywania pseudolosowej wartości całkowitej), funkcję sprawdzającą, czy wynik sortowania jest poprawny (tzn. czy wynik jest permutacją ciągu wejściowego i czy jest uporządkowany niemalejąco) oraz funkcję obliczającą minimalny, średni oraz maksymalny czas wykonania zadanego algorytmu (podanego jako parametr typu wskaźnik na funkcję) dla podanej liczby zestawów danych zadanej długości (użyć funkcji `clock` z `time.h`). Następnie, używając zdefiniowanego modułu, porównać sprawności wybranego, zaprogramowanego przez siebie, algorytmu sortowania (ale nie sortowania bąbelkowego) oraz standardowej procedury `qsort` z biblioteki `stdlib` dla 100 zestawów wygenerowanych danych o rozmiarach 5000, 10000 oraz 50000. Wyznaczyć należy minimalny, średni i maksymalny czas działania każdego z algorytmów dla podanych rozmiarów danych. W pomiarach nie należy sprawdzać poprawności wyników.
3. (10/10) Napisać program, który rozwiązuje zadanie opisane w Moodlu jako *Lista 6 zadanie 3*. Rozwiązanie tego zadania będzie sprawdzane automatycznie przez sprawdzarkę Moodlową.

29 listopada 2021

Marek Piotrów