

# Wstęp do programowania w języku C

## Lista zadań 6 (ver. 2)

1. (10/5) Kolejka to struktura, w której nowe elementy dopisywane są na końcu, a pobierane z niej elementy pochodzą z początku kolejki (tzw. FIFO). Do implementacji kolejki w tablicy potrzebne są dwa wskaźniki - jeden wskazujący początek kolejki, a drugi - jej koniec. Tablica powinna być wykorzystywana cyklicznie, jeśli na jej początku zwolni się miejsce. Dopisywanie nowego elementu do pełnej kolejki powinno powodować dwukrotne zwiększenie rozmiaru tablicy (dla której pamięć należy rezerwować w stercie). Zdefiniować strukturę reprezentującą kolejkę liczb typu `double` i funkcje: (1) inicjalizacji kolejki o zadanym rozmiarze, (2) opróżniania kolejki, (3) dopisywania liczby na końcu kolejki, (4) pobierania liczby z początku kolejki oraz (5) 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 dla podanej liczby zestawów danych zadanej długości (użyć funkcji `clock` z `time.h`). Następnie, używając zdefiniowanego modułu, porównać sprawności dwóch wybranych i zaprogramowanych przez siebie algorytmów sortowania (ale nie sortowanie bąbelkowe) 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