

Zestaw zadań nr. 6

Zadania programowe przygotowujemy w postaci kodu programu oraz dokumentu z krótkim opisem kodu oraz wynikami testów: plik.pdf (używamy latex, word) i umieszczamy w systemie Pegaz.

- Zadanie 1 (programowe)

Zaimplementuj strukturę danych **stos** oraz metody dla operacji charakterystycznych dla tej struktury:

- wstaw to stosu: $O(1)$
- pobierz ze stosu: $O(1)$
- sprawdź czy jest pusty: $O(1)$

Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu w oparciu o implementację **class stack** w bibliotece STL.

- Zadanie 2 (programowe)

Zaimplementuj strukturę danych **kolejka cykliczna** opartą na tablicy oraz metody dla operacji charakterystycznych dla tej struktury:

- wstaw to kolejki: $O(1)$
- pobierz z kolejki: $O(1)$
- sprawdź czy jest pusta: $O(1)$

Jaka jest zaleta wynikająca z używania kolejki cyklicznej?

Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu w oparciu o biblioteczne implementację kolejki cyklicznej.

- Zadanie 3 (programowe)

Zaimplementuj algorytm sortowania tablicy A typu $\text{array}[1..N]$ zbudowanej z dziesiętnych liczb k -cyfrowych przy użyciu tablicy dziesięciu kolejek. Typ liczba to $\text{array}[1..k]$ cyfr, najbardziej znaczący bit jest pod indeksem 1.

Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu w oparciu o biblioteczne implementację funkcji **sort**.

- Zadanie 4 (programowe)

Napisz funkcję sprawdzającą czy zadane wyrażenie nawiasowe składające się z nawiasów okrągłych i kwadratowych jest poprawne. Zakładamy, że dana jest funkcja `Da-jZnak:integer`, która przyjmuje następujące wartości: -1 dla `(`, 1 dla `)`, -2 dla `[`, 2 dla `]` i 0 na oznaczenie końca wyrażenia nawiasowego.

Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu.