

Zadania na ćwiczenia**Zadanie 1**

Wykonaj na wymyślonym stosie operacje

PUSH(5)

PUSH(3)

PUSH(2)

POP()

POP()

PUSH(8)

POP()

POP()

POP()

Wykonaj na wymyślonej kolejce operacje

ENQUEUE(5)

ENQUEUE(3)

ENQUEUE(2)

DEQUEUE()

DEQUEUE()

ENQUEUE(8)

DEQUEUE()

DEQUEUE()

DEQUEUE()

Zadanie 2

Uzupełnij procedury ENQUEUE i DEQUEUE o wykrywanie błędów przepełnienia i niedomiaru.

Zadanie 3

Zaimplementuj dwa stosy w jednej tablicy $A[1, n]$, tak aby do przepełnienia któregoś z nich dochodziło dopiero wtedy, gdy liczba elementów w obu stosach osiąga n .

Zadanie 4

Pokaż, jak zaimplementować kolejkę, używając dwóch stosów. Pokaż, jak zaimplementować stos za pomocą dwóch kolejek. na takim stosie.

Zadanie 5 Zaimplementuj stos za pomocą listy jednokierunkowej L . Operacje PUSH i POP powinny działać w czasie $O(1)$.

Zadanie 6 Zaimplementuj kolejkę za pomocą listy jednokierunkowej L . Operacje ENQUEUE i DEQUEUE powinny działać w czasie $O(1)$.

Zadanie 7

Napisz algorytm wyszukiwania środkowego elementu w posortowanej tablicy dwukierunkowej.

DOMOWE

ZADANIE DOMOWE (5 pkt) Napisz pseudokod procedury:

MinNaKoniec(S)

Begin

...

End.

, która działa na stosie S, w ten sposób, że przesuwa wszystkie jego elementy o wartości minimalnej na koniec stosu, a resztę pozostawia bez zmian (w takiej kolejności jak były wcześniej).

Np. dla stosu postaci (patrząc od góry trójka jest na szczycie stosu)

$$3, 2, 4, 5, 8, 2, 4$$

Zamieni go na stos postaci:

$$3, 4, 5, 8, 4, 2, 2$$

Wskazówka: Możesz użyć pomocniczego/pomocniczych stosów w swojej procedurze.