

Instrukcja – Algorytmy i Struktury Danych

Stos

Stos to struktura danych typu LIFO (Last In, First Out), gdzie ostatni dodany element jest pierwszym, który zostanie usunięty. Typowe operacje stosu to:

- `push()` – dodaje element na wierzchołek stosu,
- `pop()` – usuwa element z wierzchołka stosu,
- `top()` – zwraca element znajdujący się na wierzchołku stosu bez jego usuwania,
- `isEmpty()` – sprawdza, czy stos jest pusty.

Zaimplementuj stos za pomocą tablicy lub dynamicznej listy w C++. Stos powinien mieć następujące operacje:

1. **`void push(int value)`** – dodaje element na stos.
2. **`int pop()`** – usuwa element z wierzchołka stosu i zwraca jego wartość. Jeśli stos jest pusty, zwraca błąd.
3. **`int top()`** – zwraca wartość elementu na wierzchołku stosu bez jego usuwania. Jeśli stos jest pusty, zwraca błąd.
4. **`bool isEmpty()`** – sprawdza, czy stos jest pusty.
5. **`bool isFull()`** – sprawdza, czy stos osiągnął maksymalną wielkość (w przypadku implementacji na tablicy o stałym rozmiarze).

Zadania

Zadanie 1: Implementacja stosu z tablicą

Zaimplementuj stos za pomocą tablicy o stałym rozmiarze. Zakładamy, że stos przechowuje maksymalnie 100 elementów.

Zadanie 2: Sprawdzanie poprawności nawiasów

Napisz program, który sprawdza, czy nawiasy w danym wyrażeniu matematycznym są poprawnie zbalansowane. Skorzystaj z wcześniej zaimplementowanego stosu.

Zadanie 3: Odwracanie kolejności elementów

Napisz program, który odwróci kolejność elementów na stosie. Rozważ wariant odwracania stosu z użyciem drugiego stosu/bez użycia drugiego stosu.

Zadanie 4: Konwersja liczby dziesiętnej na system binarny

Napisz program, który przyjmuje liczbę całkowitą w systemie dziesiętnym i konwertuje ją na system binarny, wykorzystując stos.

Opis zadania:

- Program przyjmuje liczbę dziesiętną i przekształca ją na jej odpowiednik binarny.
- Aby to zrobić, liczba dziesiętna jest dzielona przez 2, a reszty z dzielenia są odkładane na stos.

Po zakończeniu procesu dzielenia program wyciąga wszystkie elementy ze stosu i drukuje wynik w postaci liczby binarnej.

Zadanie 5: Ocenianie wyrażeń infiksowych (klasycznych) z wykorzystaniem stosów

Napisz program, który ocenia wyrażenie infiksowe (zwykłe wyrażenie z operatorami umieszczonymi między operandami, np. $3 + 5 * 2$) przy użyciu dwóch stosów: jednego do przechowywania operandów i drugiego do operatorów. Program powinien obsługiwać operatory $+$, $-$, $*$, $/$ oraz uwzględniać priorytety operatorów.

Opis zadania:

- Zadanie wymaga analizy wyrażenia i przetwarzania go w taki sposób, aby poprawnie uwzględniać priorytety operatorów.
- Wyrażenie jest przetwarzane od lewej do prawej, operandy są umieszczane na stosie operandów, a operatory na stosie operatorów.
- Gdy operator na wierzchu stosu operatorów ma wyższy lub równy priorytet niż bieżący operator, dokonuje się odpowiednich obliczeń.