

Zadania (kolejki i stosy)

Zadanie 1. Napisz program, który wczytuje liczby całkowite (aż do napotkania znaku końca pliku), a następnie wypisuje je w tej samej kolejności. Skorzystaj z kolejki.

Zadanie 2. Napisz program, który wczytuje liczby całkowite (aż do napotkania znaku końca pliku), a następnie wypisuje je w odwrotnej kolejności. Skorzystaj ze stosu.

Zadanie 3. Napisz program, który wczytuje liczby całkowite (aż do napotkania znaku końca pliku), a następnie wypisuje je od największej do najmniejszej. Skorzystaj z kolejki priorytetowej.

Zadanie 4. Napisz program, który wczytuje liczby całkowite (aż do napotkania znaku końca pliku), a następnie wypisuje je od najmniejszej do największej. Skorzystaj z kolejki priorytetowej.

Zadanie 5. Mając dany stos usuń ostatni element ze stosu. Dodatkowo nie używaj metody `size()`.

Zadanie 6. Mając dany stos, odwróć jego zawartość posługując się kolejką.

Zadanie 7. Mając dany stos, odwróć jego zawartość posługując się dwoma innymi stosami.

Zadanie 8. Mając daną kolejkę, odwróć jej kolejność posługując się stosem.

Zadanie 9. Napisz program, który:

1. wczytuje liczby całkowite (aż do napotkania znaku końca pliku) i umieszcza je na stosie ST1.
2. przenosi elementy ze stosu ST1 na stos ST2 w taki sposób, aby elementy na stosie ST2 były posortowane
3. wypisuje elementy ze stosu ST2.

Zadanie 10. Napisz funkcję, która liczbę całkowitą zapisaną w postaci łańcucha znakowego (`string`) zamienia na stos cyfr tej liczby (`stack<int>`). Kolejne cyfry mają być odkładane na stos od najbardziej znaczącej do najmniej znaczącej. Przetestuj tę funkcję.

Zadanie 11. Napisz program, który dodaje dwie duże liczby naturalne (duże, czyli takie, które nie mieszczą się w standardowych typach całkowitych). Program ma działać wg. następującego algorytmu:

1. wczytaj pierwszą liczbę pod zmienną S1 typu string
2. wczytaj drugą liczbę pod zmienną S2 typu string
3. zamień S1 na ST1 typu `stack<int>` za pomocą funkcji z poprzedniego zadania
4. podobnie zamień S2 na ST2 typu `stack<int>`
5. zdejmuj kolejne cyfry ze stosów ST1 i ST2, dodawaj je i wynik umieszczaj na stosie wynikowym stw, nie zapominaj o uwzględnieniu przeniesienia
6. zdejmuj i wypisuj kolejne cyfry ze stosu STW.

Zadanie 12. Poprawnym wyrażeniem nawiasowym (w skrócie PWN) będziemy nazywali dowolny ciąg nawiasów, który powstaje z wyrażenia arytmetycznego, po usunięciu wszystkiego poza nawiasami. Na przykład PWN są: `()`, `()()`, `((()))`, a *nie są* PWN: `(`, `((()`, `()))()`. Napisz program, który wczytuje ciąg nawiasów i wypisuje komunikat, czy wczytany ciąg jest PWN, czy też nie.

Zadanie 13. To samo co w poprzednim zadaniu, ale dopuszczamy dwa rodzaje nawiasów `()` i `[]`. Przykłady ciągów, które są PWN: `()`, `[]`, `(([]))`, `()[]`, `(([]))`, przykłady ciągów, które nie są PWN: `[`, `([])`, `(([]))`.

Zadanie 14 (*). Napisz program, który sprawdza czy graf nieskierowany, zadany w postaci macierzy sąsiedztwa, jest spójny.