Listy cykliczne i dwukierunkowe

Przyjmując następującą definicję struktury elementu listy dwukierunkowej rozwiąż podane zadania.

```
struct elem {
    int dane;
    elem * poprz;
    elem * nast;
};
```

Zad. 1. Zaimplementuj podstawowe operacje na listach dwukierunkowych:

- wstawienie elementu x do listy $(a_1, a_2, \ldots, a_{i-1}, a_i, \ldots, a_n)$ pomiędzy elementy a_{i-1} oraz a_i

  `void insert(int x, int i, elem* &a)`
- usunięcie pierwszego elementu listy

  `void remove(elem* &lista)`
- usunięcie wskazanego elementu listy

  `void remove(int i, elem* &lista)`

Zad. 2. Lista dwukierunkowa może zostać „odwrócona” na dwa sposoby. Można pozmieniać wskaźniki we wszystkich elementach tak, aby dostać odwrotny porządek lub można pozostawić strukturę listy bez zmian i parami pozamieniać dane elementów listy. Zaimplementuj jeden ze sposobów.

`void reverse(elem* &lista)`

Zad. 3. Napisz procedurę przekształcającą listę jednokierunkową w listę cykliczną

`void to_cyclic(elem* lista)`

Zad. 4. Napisz procedurę zmieniającą kierunek wskaźników (pole `nast`) w liście *cyklicznej* jednokierunkowej

`void reverse_cyclic(elem* lista)`

Zad. 5. Załóżmy, że pole „dane” jest typu znakowego i może przechowywać znaki: ’+’, ’−’ (dwuargumentowy), ’∗’, ’/’, ’a’ – ’z’. Wówczas lista może reprezentować wyrażenie arytmetyczne w zapisie przedrostkowym (w notacji polskiej). Napisz funkcję sprawdzającą, czy lista reprezentuje poprawnie skonstruowane wyrażenie w takiej beznawiasowej notacji. Przykład: wyrażenie $10/(4-2)$ w postaci przedrostkowej to: / 10 − 4 2.

`bool is_valid_pn(elem* lista)`