Listy jednokierunkowe

Przyjmując następującą definicję struktury elementu stosu i kolejki rozwiąż poniższe zadania.

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

Zad. 1. Zaimplementuj podstawowe operacje na listach:

- 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 i-tego elementu listy

  `void remove(int i, elem* &lista)`
- zwrócenie i-tego elementu listy bez jego usuwania

  `int read(int i, elem* lista)`

Zad. 2. Napisz funkcję zwracającą liczbę elementów podanej listy

`int size(elem* lista)`

Zad. 3. Napisz procedurę wypisującą wszystkie elementy listy

`void print(elem* lista)`

Zad. 4. Napisz procedurę wypisującą wszystkie elementy listy w odwrotnej kolejności korzystając ze stosu.

Zad. 5. Napisz procedurę wypisującą wszystkie elementy listy w odwrotnej kolejności nie wykorzystując żadnej dodatkowej struktury danych.

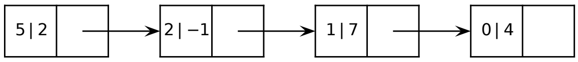
Zad. 6. Napisz procedurę usuwającą wszystkie elementy listy

`void destroy(elem* &lista)`

Zad. 7. Napisz funkcję zwracającą wskaźnik do elementu listy zawierającego w polu `dane` wartość x

`elem* search(int x, elem* lista)`

Zad. 8. Listy jednokierunkowe mogą posłużyć do reprezentowania wielomianów. Załóżmy, ża zamiast pola `dane` w rekordzie (strukturze) `elem` mamy pola (typu całkowitego) `expo` (wykładnik potęgi) i `coef` (współczynnik przy x). Wówczas lista na



Rysunek 1.1: Lista do zadania 8 reprezentująca wielomian $2x^5 - x^2 + 7x + 4$

rys. 1.1 reprezentuje wielomian $2x^5 - x^2 + 7x + 4$. Napisz funkcję zwracającą listę reprezentującą sumę dwóch wielomianów. Można założyć, że elementy listy są uporządkowane według malejących wartości `expo`.

```
elem* polyadd(elem* l1, elem* l2)
```

Zad. 9. Ułamki Fareya $F = (F_1, F_2, \ldots)$ to ciąg zdefiniowany następująco. $F_1 = \left(\frac{0}{1}, \frac{1}{1}\right)$, $F_2 = \left(\frac{0}{1}, \frac{1}{2}, \frac{1}{1}\right)$, $F_3 = \left(\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}\right)$, $F_4 = \left(\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1}\right)$, czyli wyraz F_i powstaje z F_{i-1} przez wstawienie ułamka $\frac{a+b}{c+d}$ pomiędzy każde dwa sąsiednie ułamki $\frac{a}{c}$ i $\frac{b}{d}$, jeżeli tylko $c+d \leqslant i$. Napisz program, który dla liczby n utworzy listę reprezentującą F_n. Należy zacząć od listy dla F_1 i przekształcać ją w kolejne wyrazy, aż do F_n.

```
elem_farey* ulamek_fareya(int n)
```