

Да се напише клас **List**, който да реализира двойно-свързан списък. Класът трябва да е шаблонен и да реализира следните методи:

- **bool** pushBack(**const** T&) - добавя нов елемент в края;
- **bool** pushFront(**const** T&) - добавя нов елемент в началото;
- **bool** popBack() - премахва елемент от края, ако има такъв;
- **bool** popFront() - премахва елемент от началото, ако има такъв;
- **int** getSize() **const** - връща броя на елементите в списъка;
- **const** T& getFirst() **const** и T& getFirst() - връща първият елемент от списъка;
- **const** T& getLast() **const** и T& getLast() - връща последният елемент от списъка.

За реализацията можете да използвате следната структура:

```
struct Node {  
    T data;  
    Node *prev;  
    Node *next;  
  
    Node() { }  
    Node(const T& data, Node *prev, Node *next)  
        :data(data), prev(prev), next(next) { }  
};
```

Можете да реализирате списъка, като пазите 2 указателя - към първият и последният Node. Също така, може да го реализирате, като винаги имате една инстанция на Node, която не съдържа данни. Тя служи за да затвори списъка в кръг, което улеснява всички операции, тъй като вече няма гранични случаи - всяка операция е или вмъкване между два Node-а или премахване на Node, заобиколен с два Node-а.