

Algorytmy i struktury danych I

Zestaw zadań nr. 5: Listy (c.d.)

Zadania programowe przygotowujemy w postaci kodu programu oraz dokumentu z krótkim opisem kodu oraz wynikami testów: plik.pdf (używamy latex, word) i umieszczamy w systemie Pegaz.

- Zadanie 1 (programowe)

Uzupełnij swoją implementację listy opartą o wskaźniki o metody dostępne w bibliotece STL dla listy **class list**. Możesz użyć listy pojedynczo lub podwójnie wiązanej.

Zaimplementuj metody:

- `front()` Returns reference to the first element in the list
- `back()` Returns reference to the last element in the list
- `push_front(g)` Adds a new element `g` at the beginning of the list
- `push_back(g)` Adds a new element `g` at the end of the list
- `pop_front()` Removes the first element of the list, and reduces size of the list by 1
- `pop_back()` Removes the last element of the list, and reduces size of the list by 1
- `begin()` Returns an iterator pointing to the first element of the list
- `end()` Returns an iterator pointing to the theoretical last element which follows the last element
- `empty()` Returns whether the list is empty(1) or not(0)
- `insert()` Inserts new elements in the list before the element at a specified position
- `erase()` Removes a single element or a range of elements from the list
- `assign()` Assigns new elements to list by replacing current elements and resizes the list
- `remove()` Removes all the elements from the list, which are equal to given element
- `reverse()` Reverses the list
- `size()` Returns the number of elements in the list
- `sort()` Sorts the list in increasing order

Napisz kod do generacji testowego inputu oraz kod do generowania wzorcowego outputu w oparciu o implementację **class list** w bibliotece STL.