Zaimplementuj strukturę danych „lista jednokierunkowa” w sposób następujący:

* Pojedynczy element struktury powinien przechowywać wartość typu double.
* Zdefiniuj metodę addTo(lista l, double arg, unsigned pos), która pozwoli na dodanie nowego elementu na wskazaną pozycję listy;
* Zdefiniuj metode deleteFrom(lista l, unsigned pos), która pozwoli na usunięcie elementu ze wskazanej pozycji listy;

Rozwiąż następujące zadanie.

Masz do dyspozycji dwa pliki:

V3\_\*.tsk - zawiera dane wejściowe Twojego algorytmu;

solv\_V3\_\*.tsk – zawiera docelowy wynik działania algorytmu.

Otwórz plik V3\_\*.tsk w trybie binarnym, wczytuj jego zawartość bajt bo bajcie.

* Wczytaj bajt, to będzie kod polecenia.
* Jeżeli wczytany bajt jest kodem znaku ‘<’:

Wczytaj kolejny bajt, to będzie numer pozycji, **pos**,

Wczytaj kolejny bajt, to będzie wartość, **arg**,

Dodaj wartość **arg** na pozycję **pos**.

* Jeżeli wczytany bajt jest kodem znaku ‘>’:

Wczytaj kolejny bajt, to będzie numer pozycji, **pos**,

Usuń element listy o numerze **pos**.

* Jeżeli wczytany bajt jest kodem znaku ‘!’:

Wczytaj kolejny bajt, to będzie numer pozycji, **pos**,

Przenieś element o podanym numerze na koniec(tail), jeżeli wartość w polu data <= 0 albo na początek (head), jeżeli >0.

Po wczytaniu całego pliku znajdź największą sumę wartości pól data dwóch sąsiednich elementów. Jeżeli lista zawiera tylko jeden element, zwróć wartość jego pola data. Jeżeli lista jest pusta zwróć 0.

Wynik wyświetl.