

Kod zawiera:

- implementacje listy jednostronnej (singly link list) opartej na wskaźnikach
- funkcję pomocniczą `split()` do podziału stringa w zależności od separatora
- funkcję do wczytania danych z pliku i dodania ich do listy
- trzy funkcje - zadania

Funkcja wczytująca dane:

dla każdej linii w pliku `presidents.txt`, ze względu na występowanie podwójnego imienia sprawdza czy taka sytuacja występuje, jeżeli tak to do pola `name` zapisuje oba imiona rozdzielone spacją. Sprawdzenie dokonywane jest poprzez sprawdzenie czy na pozycji trzeciej (trzecie "słowo" w wierszu) znajduje się data (dokładniej czy pierwszy znak jest cyfrą) jeżeli tak, oznacza to że pierwsze słowo to imię a drugie nazwisko, w przeciwnym razie pierwsze i drugie słowo do imiona a trzecie słowo jest nazwiskiem. Przy wystąpieniu tej drugiej opcji ustawiany jest `offset = 1`, dzięki któremu pozostałe wartości zostają dobrze odczytane i przypisane.

Modyfikacja pliku z danymi:

- dodanie znaku końca linii w ostatnim wierszu ze względu na usuwanie go z podstringa `party`

Uwagi:

- klasa `List` powinna nazywać się np. `PresidentList`, ponieważ wartości przechowywane przy jej pomocy są typu `President`, a nie pozwala ona na przechowywanie danych dowolnego typu.
- nazwy funkcji używają notacji z `'_'` zamiast `camelCase`
- funkcje (zadanie 1 i 2) zwracające więcej niż jeden element zwracają listę elementów

```

1 > #include<iostream>...
5
6     class President;
7     class Node;
8     class List;
9
10    // function declarations
11    std::string* split(std::string s, std::string delimiter);
12    void load_presidents_data(List& list);
13    List najkrotszy_czas(List& list);
14    List dana_partia(List& list, std::string party);
15    President prezydent_w_roku(List& list, short year);
16
17    // class definitions
18 > class President{...
34
35 > class Node{...
59
60 > class List{...
119
120 > int main() ...
183
184    // function definitions
185 > std::string* split(std::string s, std::string delimiter) {...
211
212 > void load_presidents_data(List& list){...
256
257    // zadanie 1
258 > List najkrotszy_czas(List& list){...
282
283    // zadanie 2
284 > List dana_partia(List& list, std::string party){...
299
300    // zadanie 3
301 > President prezydent_w_roku(List& list, short year){...

```