

Podstawy programowania
Projekt końcowy Produkty

Mamy podany plik tekstowy, który zawiera dane o produktach różnej kategorii, np. książkach, filmach, muzyce, ect. Każda linia pliku opisuje jeden produkt. Pierwsza wartość w linii reprezentuje nazwę kategorii produktu, druga wartość to unikatowy kod tego produktu, z kolei trzecia wartość to jego ilość w magazynie. Poniżej znajduje się przykład takiego pliku.

```
muzyka XP234 100
film YS132 21
ksiazka DQ983 65
muzyka XP234 30
film YW276 3
. . . .
```

Napisz następujące funkcje:

- (1) Funkcję, która przeczyta dane z pliku tekstowego i umieści je na liście list jednokierunkowych. Lista główna będzie przechowywać informację o kategorii produktów, z kolei podlisty będą reprezentować kolejne produkty tej kategorii. Lista główna powinna mieć tyle elementów, ile różnych kategorii znajdzie się w pliku tekstowym. Z kolei podlisty powinny reprezentować produkty. Uwaga: w podliście danej kategorii powinien znaleźć się tylko jeden element opisujący dany produkt. Jeśli produkt o podanym kodzie występuje kilka razy w pliku tekstowym, na podliście powinien się pojawić tylko raz z sumowaną wartością magazynową.
- (2) Funkcję, która będzie wypisywać zawartość listy list na standardowe wyjście.
- (3) Funkcję, która pomniejszy stan magazynowy produktu o podanej kategorii i kodzie produktu. Uwaga, jeśli stan magazynowy po pomniejszeniu o określoną wartość będzie wynosił mniej niż 0, wówczas funkcja powinna zmniejszyć stan tylko do wartości 0 i wygenerować odpowiedni komunikat.
- (4) Funkcję, która przepisze do pliku tekstowego wszystkie dane z listy list. Plik wyjściowy powinien mieć taki sam format jak plik wejściowy.
- (5) Funkcję, które usuną kolejne elementy listy list.

Wywołaj zaimplementowane funkcje w formie menu.

Przykładowe typy danych dla listy list:

```
typedef struct element_podlisty{
    char kod[M];
    long stan;
    struct element_podlisty *pod_next;
} Element;

typedef struct element_listy_glownej{
    char kategoria[N];
    struct element_listy_glownej *next;
    struct element_podlisty *pod_next;
} GElement;
```