

## Program wspomagający zarządzanie siecią hurtowni

Głównym zadaniem programu jest „obsługa zamówienia”. Program wczytuje informacje o zamawianych artykułach i planuje, z których hurtowni będą wysyłane poszczególne artykuły. Jedno zamówienie obejmuje artykuły, które mają być dostarczone do jednej miejscowości.

Program przechowuje dane o kilku hurtowniach usytuowanych w odległych od siebie miastach. Decyzja o wyborze hurtowni ma wpływ na koszt przesyłki (koszt jest proporcjonalny do odległości między miejscowością, z której pochodzi zamówienie, a hurtownią).

Dane o artykułach dostępnych w każdej z hurtowni są uaktualniane po realizacji każdego zamówienia. Nie rozróżniamy etapu planowania dystrybucji i jej realizacji, dlatego uaktualnienie stanu hurtowni dokonywane jest bezpośrednio po obsłudze zamówienia tj. po ustaleniu, które artykuły będą pobrane z których hurtowni.

### Funkcje programu

1. Obsługa zamówienia, aktualizacja stanu hurtowni, obliczenie kosztu realizacji zamówienia (wartości artykułów i koszty transportu).
2. Generowanie raportu dobowego o artykułach przeterminowanych (z zerowaniem stanu tych artykułów) i artykułach, których termin przydatności jest krótszy niż zadany próg.
3. Generowanie raportu dobowego o przewidywanym (w horyzoncie  $n$  dni) wyczerpaniu zapasów w każdej z hurtowni.

### Dane dla programu

1. **Dane o miejscowościach**, w których są hurtownie lub skąd mogą pochodzić zamówienia: Nazwa miejscowości i odległości do innych miejscowości połączonych bezpośrednio np. w przypadku Krakowa wymieniony byłby np. Tarnów, ale nie byłoby Rzeszowa bo nikt rozsądny nie jedzie do Rzeszowa inną drogą niż przez Tarnów (nie rozróżniamy istnienia lub nie obwodnic). W przypadku Tarnowa będzie podana odległość do Krakowa i do Rzeszowa. Połączenia są dwukierunkowe. Jeżeli jest podane połączenie z Krakowa do Tarnowa, to nie musi być podawane połączenie z Tarnowa do Krakowa – przyjmujemy, że jest. W pliku `miejscowosci.dat` żadne połączenie nie jest powtórzone, a w pliku `miejscowosci_v2.dat` każde połączenie jest zapisane dwa razy.

Format plików `miejscowosci.dat` oraz `miejscowosci_v2.dat` (nazwy bez znaków diakrytycznych)

Miejscowość

{liczba kilometrów do} Miejscowość

{liczba kilometrów do} Miejscowość

...

{liczba kilometrów do} Miejscowość

Miejscowość

{liczba kilometrów do} Miejscowość

{liczba kilometrów do} Miejscowość

...

{liczba kilometrów do} Miejscowość

Miejscowość

{liczba kilometrów do} Miejscowość

{liczba kilometrów do} Miejscowość

...

{liczba kilometrów do} Miejscowość

...

## 2. Dane o artykułach

Część danych o każdym artykule jest wspólna dla wszystkich hurtowni (nazwa, ID), a część nie (cena, data ważności, liczba sztuk).

Format pliku `artykuly.dat`:

ID Artykuł\n

ID Artykuł\n

...

ID Artykuł\n

## 3. Dane o hurtowniach

Dla każdego artykułu pamiętane są:

– ilość (zakładamy, że wszystkie artykuły są liczone w sztukach),

– cena,

– termin ważności (data),

Format pliku `hurtownie_stan0.dat`:

Miejscowość\n

ID sztuk cena dzien miesiac rok\n

ID sztuk cena dzien miesiac rok\n

...

ID sztuk cena dzien miesiac rok\n

Miejscowość\n

ID sztuk cena dzien miesiac rok\n

ID sztuk cena dzien miesiac rok\n

...

ID sztuk cena dzien miesiac rok\n

...

## 4. Dane o zleceniu

Format pliku `zlecenie.dat`:

Dzien miesiac rok Nr\_zlecenia Miejscowosc\_odbiorcy\n

sztuk Artykuł\n

sztuk Artykuł\n

sztuk Artykuł\n

sztuk Artykuł\n

...

Dzien miesiac rok Nr\_zlecenia Miejscowosc\_odbiorcy\n

sztuk Artykuł\n

sztuk Artykuł\n

sztuk Artykuł\n

sztuk Artykuł\n

...

## Dane wyjściowe

1. Plik danych `hurtownie_stan1.dat` zawierający zaktualizowane dane po realizacji zleceń.
2. Plik `przeterminowane.dat` zawierający nazwy artykułów, dla których przewidywany termin przydatności do użycia jest krótszy niż  $n$  dni ( $n$  może być równe 0).
3. Plik `koncowki.dat` zawierający nazwy artykułów, dla których przewidywany termin wyczerpania zapasu jest krótszy niż  $n$  dni.
4. Plik `realizacja_zleceń.dat` zawierający dane o koszcie i sposobie (z której hurtowni) realizacji zamówienia.

## Uwagi:

- Określenie w formatach plików: „Artykuł” oznacza nazwę artykułu (może zawierać spację). Nazwa każdego artykułu zaczyna się od litery.
- Data ważności: 0 0 0 oznacza, że nie ma ograniczenia ważności.
- W jednej hurtowni może być kilka partii jednego artykułu, ale każda o innym terminie ważności. Jeżeli w hurtowni jest kilka „serii” takiego samego artykułu, to w pierwszej kolejności wydawany jest artykuł o najbliższej dacie ważności.
- Dla uproszczenia nie rozróżniamy dni roboczych i świątecznych.
- Rozważane zlecenie jest zleceniem zbiorczym – grupującym wszystkie zlecenia indywidualne pochodzące od odbiorców w jednej miejscowości. Nie ma zatem kilku zleceń do tej samej miejscowości w jednym dniu.
- Po wyczerpaniu artykułu wpis w hurtowni nie jest usuwany, jedynie zerowana jest liczba sztuk.
- Program powinien „obsługiwać” różne pliki danych (inna liczba miejscowości, hurtowni, artykułów).
- Program nie pobiera informacji o dacie z systemu – daty realizacji zamówienia oraz daty ważności artykułów nie są „zsynchronizowane” z czasem rzeczywistym. Jako bieżącą datę (w obrębie programu - nie trzeba zmieniać daty systemowej) należy przyjąć datę ostatnio realizowanego zamówienia.

## Możliwe rozszerzenia programu dla ambitnych:

- \* Program znajduje najtańszy sposób realizacji kolejnego zamówienia.
- \* Na podstawie podanej historii zleceń program generuje raport o przewidywanych terminach wyczerpania zapasu.

## Punktacja:

1	Poprawny program w wersji podstawowej	+50
2	W hurtowni może być więcej niż 1 partia jednego artykułu	+10
3	Dynamiczny przydział pamięci dla nazw, struktur danych o miejscowościach	+10
4	Dynamiczny przydział pamięci dla innych struktur danych	+10
5	* Optymalizacja drogi powrotu po skompletowaniu zamówienia (z rekurencją)	+15
6	* Minimalizacja kosztu zamówienia	+15
7	* Przewidywanie terminu wyczerpania zapasu	+20
8	Wersja z uzupełnianiem danych wejściowych o połączenia „powrotne”	+5
9	Prawidłowy podział na funkcje, minimalizacja informacji przekazywanej do funkcji	+10
10	Rozdzielenie funkcji w różnych plikach, użycie pliku nagłówkowego	+5
11	Niebezpieczne wczytywanie danych (łańcuchów) – bez kontroli długości łańcucha	-5
12	Brak kontroli realizacji funkcji odwołujących się do systemu (np. <code>malloc</code> , <code>fopen</code> )	-5
13	Styl – nazwy funkcji, zmiennych, komentarze	-10
14	Styl – wcięcia, klamry, czytelność	-10

15	Pojawienie się ostrzeżeń kompilatora, niezgodność z C90	-10
16	Stosowanie zmiennych globalnych itp.	-20
17	Przekroczenie terminu oddania programu	-20
18	Stosowanie instrukcji <code>goto</code> itp.	-50

ad 1: Wersja podstawowa – funkcjonalność:

- czyta i zapamiętuje dane z plików (wczytanych z podanego jako parametr katalogu, będącego podkatalogiem bieżącego katalogu. Przykładowe wywołanie programu: `./a.out katalog`),
- oblicza koszt realizacji zamówienia,
- aktualizuje stan hurtowni,
- zapisuje plik z aktualnymi danymi hurtowni.

Wersja podstawowa – wymogi pozafunkcjonalne:

- użycie instrukcji `typedef`,
- podział programu na funkcje.

ad 5: Jeżeli algorytm realizacji zamówienia polega na „odwiedzaniu” kolejnych hurtowni, to w momencie skompletowania całości znajdujemy się w miejscowości – nazwijmy ją „końcową”, która prawdopodobnie nie jest miejscowością startu). Prosty sposób powrotu do punktu startu jest podążanie „po śladach”. Punkty przewidziane w wierszu 5 będą przydzielane w przypadku optymalizacji kosztu tego powrotu – dobrze byłoby zastosować algorytm rekurencyjny (o większej złożoności niż np. algorytm Dijkstry, ale ciekawszy z punktu widzenia programowania).

ad 6: Minimalizowany koszt obejmuje także koszt powrotu (z pkt. 6), więc w sumie przysługuje do 30 punktów.

ad 8: W tej wersji program czytając dane o połączeniu z miejscowości A do miejscowości B zapamiętuje, że także miejscowość B jest połączona z A (informacja o tym połączeniu nie jest powtarzana przy opisie miejscowości B).

Uzyskanie 100 punktów jest traktowane jako wykonanie zadania w 100%. W zależności od jakości i stopnia realizacji części nadobowiązkowej (\*) programu, istnieje możliwość przesunięcia nadmiarowych punktów na ocenę z kolokwium (w wyjątkowych przypadkach również zwolnienia z kolokwium).

Uzupełnienia (będą sprecyzowane w najbliższym czasie – do końca grudnia):

- Opis wersji danych wejściowych.
- Formaty plików wyjściowych.
- Format informacji o wybranej wersji programu.