

Języki i narzędzia programowania I (Info, II rok) 19/20

Kokpit ► Moje kursy ► JNP1.INFO.II.19/20 ► Temat 1 ► Zadanie 1

Zadanie 1

== Kasa Biletowa ==

Napisz program wspomagający pracę kasy sprzedającej bilety na połączenia tramwajowe. Tramwaje są czynne od 5:55 do 21:21 (nie jeżdżą poza tymi godzinami i pasażerowie nie korzystają z połączeń obejmujących więcej niż jeden dzień). Program na standardowe wejście dostaje wiersze w niżej opisanych formatach.

1. Dodanie nowego kursu do rozkładu

numer_kursu czas_1 przystanek_1 czas_2 przystanek_2 ... czas_n przystanek_n

Parametr numer_kursu jest liczbą całkowitą, być może z dodatkowymi zerami na początku (ignorujemy je). Numery kursów nie powtarzają się.

Parametry czas_x są w zwykłym formacie, np. 7:15 albo 19:01. Czasy dla kolejnych przystanków są rosnące i kurs nie przejeżdża więcej niż raz przez jeden przystanek.

Parametry przystanek_x są nazwami przystanków – są to niepuste napisy składające się z liter alfabetu angielskiego oraz znaków '_' i '^'.

2. Dodanie biletu do rozkładu

nazwa_biletu cena czas_ważności

Parametr nazwa_biletu może zawierać tylko litery z alfabetu angielskiego oraz spacje i jest to niepusty napis.

Parametr cena jest liczbą z dwoma miejscami dziesiętnymi, np. 6.23.

Parametr czas_ważności jest to niezerowa liczba naturalna bez zer wiodących – liczba minut, przez które bilet jest ważny. Na przykład bilet dwuminutowy skasowany o 8:00 jest ważny od 8:00 do 8:01 (pierwsza minuta od 8:00:00 do 8:00:59, a druga od 8:01:00 do 8:01:59).

3. Pytanie o bilety

? przystanek_1 numer_kursu_1 przystanek_2 numer_kursu_2 ... przystanek_n numer_kursu_n
przystanek_n+1

Podróżny chce kupić najtańszy zestaw biletów obejmujący podaną trasę. Zestaw może składać się co najwyżej z trzech biletów. Podróżny wsiada do tramwaju numer_kursu_1 na przystanku przystanek_1, na przystanku przystanek_2 wysiada z niego i wsiada do tramwaju numer_kursu_2 itd., a na końcu wysiada na przystanku przystanek_n+1.

Przedziały ważności biletów są domknięte i można je dowolnie łączyć, np. jadąc tramwajem od 10:10 do 10:29 można kupić dwa bilety 10-minutowe – jeden ważny od 10:10 do 10:19, a drugi od 10:20 do 10:29.

Zestaw biletów należy wypisać na standardowe wyjście w następującym formacie:

! nazwa_biletu; nazwa_biletu_2; ...; nazwa_biletu_n

Pasażerowie nie lubią czekać na przystanku i przesiadki są zawsze na styk, np. tramwaj przyjeżdża na przystanek o 15:00 i następny odjeżdża z tego przystanku o 15:00. Jeżeli trzeba by było czekać, to nie proponujemy biletów, tylko wypisujemy na standardowe wyjście:

:-(nazwa_przystanku_gdzie_trzeba_czekać

Jeśli nie da się kupić biletów na podaną trasę, to wypisujemy na standardowe wyjście:

:-|

Jeśli istnieje więcej niż jeden zestaw biletów w minimalnej cenie, to należy wypisać dowolny z nich.

Na koniec należy wypisać łączną liczbę biletów proponowanych we wszystkich odpowiedziach na pytanie (trzeci wariant formatu wejściowego).

== Wymagania funkcjonalne ==

Program czyta dane wiersz po wierszu ze standardowego wejścia. Program analizuje dane wejściowe wiersz po wierszu. Program ignoruje puste wiersze. Dla każdego niepustego wiersza program decyduje, czy wiersz ten jest poprawny. Jeśli wiersz nie jest poprawny, to informacje w nim zawarte nie są brane pod uwagę przy dalszym przetwarzaniu, a program wypisuje na standardowe wyjście diagnostyczne wiersz z informacją o błędzie, zawierającą numer błędnego wiersza i dokładną, oryginalną jego postać. W szczególności wiersz z danymi spoza ustalonych w treści godzin funkcjonowania tramwajów jest niepoprawny. Wiersze numeruje się od 1. Należy zastosować format dokładnie zgodny z podanym przykładem danych wyjściowych.

```
g++ -Wall -Wextra -O2 -std=c++17 kasa.cc -o kasa
```

Oczekiwane rozwiązanie nie powinno zawierać definicji własnych struktur i klas, a przynajmniej takich, które zawierają dane. Zamiast tego należy intensywnie korzystać z kontenerów i algorytmów dostarczanych przez standardową bibliotekę języka C++. Obsługę wejścia i wyjścia należy zrealizować za pomocą strumieni.

Rozwiązanie należy umieścić w pliku kasa.cc, który należy umieścić w repozytorium SVN w katalogu

grupaN/zadanie1/ab123456+cd123456

lub

grupaN/zadanie1/ab123456+cd123456+ef123456

gdzie N jest numerem grupy, a ab123456, cd123456, ef123456 są identyfikatorami członków zespołu umieszczającego to rozwiązanie. Katalog z rozwiązaniem nie powinien zawierać innych plików, ale może zawierać podkatalog prywatne, gdzie można umieszczać różne pliki, np. swoje testy. Pliki umieszczone w tym podkatalogu nie będą oceniane. Nie wolno umieszczać w repozytorium plików dużych, binarnych, tymczasowych (np. *.o) ani innych zbędnych.

 przykład.err

 przykład.in

 przykład.out

Status przesłanego zadania

Status przesłanego zadania	To zadanie nie wymaga wysyłania niczego online
----------------------------	--

Stan oceniania	Nie ocenione
----------------	--------------

Ostatnio modyfikowane	-
-----------------------	---

Komentarz do przesłanego zadania	► Komentarze (0)
----------------------------------	------------------

◀ Czytanki

Przejdź do...

Czytanki ►