Úkolem je realizovat program, který bude počítat čas a cenu za odvoz zboží.

Předpokládáme, že potřebujeme odvézt zboží. Dopravce má k dispozici přepravní kapacitu, kterou nám může nabídnout. Obecně je k dispozici 1 nebo více dopravních prostředků. Pro každý dopravní prostředek známe:

- časový interval <from;to>, kdy je k dispozici,
- počet kusů zboží, které za den přepraví a
- cenu za jeden den použití dopravního prostředku.

Pro známou nabídku budeme opakovaně vyhodnocovat cenové a časové kalkulace. Pro zadaný počáteční den a zadaný počet kusů zboží chceme určit:

- kdy všechno zboží přepravíme (poslední den) a
- kolik za dopravu celkem zaplatíme.

Naším cílem je zboží přepravit co nejdříve, tedy při plánování nehledíme na cenu. Dopravce stanoví cenu podle počátečního a koncového data pronájmu – zaplatíme součet denních pronájmů všech dopravních prostředků, které v tomto intervalu jsou k dispozici. Toto je pro dopravce výhodné, protože zaplatíme např. i nevyužitou dopravní kapacitu některých dopravních prostředků v poslední den přepravy. Na druhou stranu toto omezení zjednodušuje algoritmus výpočtu. Konkrétní ukázku výpočtu ceny najdete v poznámkách.

Vstupem programu je zadání nabízených dopravních prostředků, po kterém následuje seznam problémů k vyřešení. Nabídka dopravních prostředků je zadána ve tvaru:

{ [2 - 6, 3, 7] , [4-9,2,10],[15-30, 4, 12], [12-12,1,1] }

složené závorky ohraničují zadávání nabídky, parametry jednotlivých dopravních prostředků jsou v hranatých závorkách. Dopravní prostředek je zadán intervalem, kdy je nabízen (dvě celá nezáporná čísla oddělená pomlčkou), denní přepravní kapacitou (celé kladné číslo) a cenou za den pronájmu (celé kladné číslo).

Po zadání nabídky následuje seznam problémů k vyřešení. Každý řešený problém je zadán dvojicí celých čísel: počátečním dnem a počtem kusů přepravovaného zboží. Zadávání řešených problémů je ukončeno dosažením konce vstupu (EOF).

Výstupem programu je řešení jednotlivých zadaných problémů. Pro každý zadaný problém je zobrazena informace o posledním dni přepravy a celkové ceně za pronájem. Pokud pro zadaný počáteční den / počet kusů zboží nelze přepravu realizovat, je zobrazena informace o příliš nízké kapacitě (viz ukázkové běhy).

Pokud vstup není platný, program tuto situaci detekuje, vypíše chybové hlášení a ukončí se. Formát chybového hlášení je opět uveden v ukázkách níže. Za chybu je považováno:

- chybějící složené závorky pro seznam dopravních prostředků,
- chybějící hranaté závorky pro zadání dopravního prostředku,

- chybějící čárka oddělující dopravní prostředky,
- neplatná nebo chybějící čísla (intervaly, kapacita, cena),
- počátek nebo konec intervalu je záporný,
- počátek intervalu je větší než konec intervalu,
- kapacita je záporná nebo nulová,
- cena je záporná nebo nulová,
- chybějící čárka oddělující interval/kapacitu/cenu u dopravního prostředku,
- dopravních prostředků je více než 100000,
- neplatné zadání řešeného problému (nečíselný nebo záporný počáteční den, nečíselný, nulový nebo záporný počet kusů zboží).

Pokud program detekuje chybu, přestane se dotazovat na další vstupní hodnoty, vypíše chybové hlášení a ukončí se. Chybu je tedy potřeba detekovat okamžitě po načtení hodnoty (neodkládejte kontrolu vstupních údajů až za načtení celého vstupu). Chybové hlášení vypisujte na standardní výstup (nevypisujte jej na standardní chybový výstup).

Dodržte přesně formát všech výpisů. Výpis Vašeho programu musí přesně odpovídat ukázkám. Testování provádí stroj, který kontroluje výpis na přesnou shodu. Pokud se výpis Vašeho programu liší od referenčního výstupu, je Vaše odpověď považovaná za nesprávnou. Záleží i na mezerách, i na odřádkování. Nezapomeňte na odřádkování za posledním řádkem výstupu (a za případným chybovým hlášením). Využijte přiložený archiv s testovacími vstupy a usnadněte si testování Vašeho programu.

Váš program bude spouštěn v omezeném testovacím prostředí. Je omezen dobou běhu (limit je vidět v logu referenčního řešení) a dále je omezena i velikost dostupné paměti (toto by neměl být problém). Zadaný problém lze řešit algoritmy s různou efektivitou. Rozumná implementace naivního algoritmu projde povinnými testy, ale nezvládne časový limit testů bonusových. Takové řešení bude hodnoceno 100% bodů. Pro zvládnutí bonusových testů je potřeba efektivnější algoritmus, který dokáže rychle zpracovat dlouhé časové intervaly a velký počet dopravních prostředků:

- první bonus zadává krátké časové intervaly,
- druhý bonus zadává dlouhé časové intervaly a malé množství zboží k přepravě,
- třetí bonus zadává dlouhé časové intervaly a velké množství zboží k přepravě.

Ukázka práce programu:

Moznosti dopravy:
{ [2 - 6, 3, 7] , [4-9,2,10],[15-30, 4, 12], [12-12,1,1] }
Naklad:
0 6
Konec: 3, cena: 14
2 6
Konec: 3, cena: 14
1 16
Konec: 5, cena: 48
3 25
Konec: 12, cena: 89
3 26

Konec: 15, cena: 101
5 81
Konec: 30, cena: 257
5 82
Prilis velky naklad, nelze odvezt.
2 7
Konec: 4, cena: 31
30 2
Konec: 30, cena: 12

Moznosti dopravy:
```
{
  [ 17-74, 5, 44],
  [ 57-78, 35, 19],
  [ 39-77, 43, 29],
  [ 56-95, 44, 9],
  [ 8-94, 2, 8],
  [ 52-87, 22, 14],
  [ 31-77, 6, 39],
  [ 64-92, 26, 35],
  [ 43-60, 29, 32],
  [ 46-63, 7, 49]
}
```
Naklad:
14 10
Konec: 17, cena: 76
32 21
Konec: 33, cena: 182
35 9
Konec: 35, cena: 91
10 21
Konec: 17, cena: 108

Moznosti dopravy:
```
{ [ 10-9, 1, 1 ] }
```
Nespravny vstup.

Moznosti dopravy:
```
{ [ 1-5, 1, 1 ] }
```
Naklad:
-1 10
Nespravny vstup.

Poznámky:

- Ukázkové běhy zachycují očekávané výpisy Vašeho programu (tučné písmo) a vstupy zadané uživatelem (základní písmo). Zvýraznění tučným písmem je použité pouze zde na stránce zadání, aby byl výpis lépe čitelný. Váš program má za úkol pouze zobrazit text bez zvýrazňování (bez HTML markupu).
- Znak odřádkování (\n) je i za poslední řádkou výstupu (i za případným chybovým hlášením).

- Program si část vstupních dat musí uložit do paměti, ideálně do pole. Protože je velikost vstupu omezena 100000 dopravními prostředky, lze alokaci provést staticky.
- Počítání ceny nájmu je trochu neintuitivní, ale zjednodušuje implementaci. V ukázkovém běhu #1 je pro vstup 2 7 vypočteno dokončení přepravy v den 4 za cenu 31. Zaplatíme pronájem prvního dopravního prostředku ve dnech 2, 3 a 4 (tj. kapacita 3*3 a cena 3*7) a pronájem druhého dopravního prostředku v den 4 (tj. kapacita 2 a cena 10). Pro náš náklad o velikosti 7 by stačil pouze pronájem prvního dopravního prostředku ve dnech 2, 3 a 4. Zadání ale určuje, že zaplatíme pronájem všech nabízených dopravních prostředků, které jsou pro zvolený časový interval k dispozici (zde dny 2–4). Pokud by výpočet měl zohledňovat částečné pronájmy jen některých dopravních prostředků v některé dny, byl by program mnohem komplikovanější.
- Pokud chcete úlohu vyřešit i s bonusovými testy, doporučujeme ukládat hodnoty celkové ceny a celkového počtu zboží v datovém typu long long. Pro řešení bez bonusů postačuje typ int.