

Přestupy na nádraží

Termín odevzdání:	28.10.2012 23:59:59
Hodnocení:	5.5000
Max. hodnocení:	5.0000 (bez bonusů)
Odevzdaná řešení:	2 / 10 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat program, který bude rozhodovat o možnostech přestupu mezi vlaky na nádraží.

Předpokládáme, že na nádraží jezdí za den právě 3 vlaky - vlak A, B a C. Vlaky jezdí denně, přijíždějí a odjíždějí vždy ve stejný čas. Pro každý vlak známe hodinu a minutu příjezdu a odjezdu. Úkolem programu je rozhodnout, mezi kterými vlaky lze přestoupit. Z vlaku X lze přestoupit do vlaku Y pokud vlak Y odjíždí alespoň 5 minut po příjezdu vlaku X (5 minut je na přestup). Dále, přestoupit můžeme pouze tehdy, pokud od příjezdu X do příjezdu Y jsou nejvýše 3 hodiny včetně. Pokud by vlak Y přijel později, zdrželi bychom se na nástupišti déle než 3 hodiny a byli bychom z nádraží vykázáni jako potenciální bezdomovci. Oba limity (5 min, 3h) jsou brány včetně, je to i demonstrováno v ukázkách níže.

Vstupem programu je 6 časových údajů o příjezdech a odjezdech vlaků, každý údaj je tvořen hodinou a minutou. Hodiny a minuty jsou oddělené dvojtečkou.

Výstupem programu je rozhodnutí, mezi kterými vlaky lze přestupovat. Odpovědi jsou postupně pro vlaky A, B a C, každá na zvláštním řádku. Formát výstupu je zřejmý z ukázek níže. Pokud lze přestupovat na více vlaků, jsou vlaky, na které lze přestoupit, uvedeny abecedně:

Z vlaku B lze přestoupit na vlaky A a C.

Naopak, následující odpověď by byla vyhodnocena jako nesprávná:

Z vlaku B lze přestoupit na vlaky C a A.

Nezapomeňte na odřádkování za každou (tedy i poslední) řádkou výpisu.

Pokud vstup není platný (na vstupu jsou nečíselné nebo nesmyslné hodnoty), program tuto situaci detekuje a vypíše chybové hlášení. Formát chybového hlášení je opět uveden v ukázkách níže. Za chybu je považováno, pokud je na vstupu:

- nečíselná hodnota,
- chybějící oddělovač (dvojtečka) nebo
- nesmyslná hodnota (hodnota mimo rozsah 0-23, resp. 0-59).

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 považována 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 výstupy a přečtěte si sekci FAQ, jak využít přesměrování vstupů/výstupů k 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 (ale tato úloha by ani s jedním omezením neměla mít problém). Testovací prostředí dále zakazuje používat některé "nebezpečné funkce" -- funkce pro spouštění programu, pro práci se sítí, ... Pokud jsou tyto funkce použité, program se nespustí. Možná ve svém programu používáte volání:

```
int main ( int argc, char * argv [] )
{
    ...

    system ( "pause" ); /* aby se nezavřelo okno programu */
    return 0;
}
```

Toto nebude v testovacím prostředí fungovat - je zakázáno spouštění jiného programu. (I pokud by se program spustil, byl by odmítnut. Nebyl by totiž nikdo, kdo by pauzu "odmáčkl", program by čekal věčně a překročil by tak maximální dobu běhu.) Pokud tedy chcete zachovat pauzu pro testování na Vašem počítači a zároveň chcete mít jistotu, že program pobeží správně, použijte následující trik:

```
int main ( int argc, char * argv [] )
{
    ...

#ifdef __PROGTEST__
    system ( "pause" ); /* toto progtest "nevidi" */
#endif /* __PROGTEST__ */
    return 0;
}
```

Ukázka práce programu:

Zadejte čas příjezdu vlaku A:

12:00

Zadejte čas odjezdu vlaku A:

12:30

Zadejte čas příjezdu vlaku B:

12:10

Zadejte čas odjezdu vlaku B:

12:20

Zadejte čas příjezdu vlaku C:

12:40

Zadejte čas odjezdu vlaku C:

13:00

Z vlaku A lze přestoupit na vlaky B a C.

Z vlaku B lze přestoupit na vlaky A a C.

Z vlaku C nelze přestupovat.

Zadejte čas příjezdu vlaku A:

10:00

Zadejte čas odjezdu vlaku A:

10:10

Zadejte čas příjezdu vlaku B:

10:05

Zadejte čas odjezdu vlaku B:

10:15

Zadejte čas příjezdu vlaku C:

10:06

Zadejte čas odjezdu vlaku C:

10:16

Z vlaku A lze přestoupit na vlaky B a C.

Z vlaku B lze přestoupit na vlaky A a C.

Z vlaku C lze přestoupit na vlak B.

Zadejte čas příjezdu vlaku A:

10:00

Zadejte čas odjezdu vlaku A:

10:30

Zadejte čas příjezdu vlaku B:

13:00

Zadejte čas odjezdu vlaku B:

13:10

Zadejte čas příjezdu vlaku C:

13:01

Zadejte čas odjezdu vlaku C:

14:00

Z vlaku A lze přestoupit na vlak B.

Z vlaku B lze přestoupit na vlak C.

Z vlaku C lze přestoupit na vlak B.

Zadejte čas příjezdu vlaku A:

23:00

Zadejte čas odjezdu vlaku A:

23:30

Zadejte čas příjezdu vlaku B:

23:50

Zadejte čas odjezdu vlaku B:

0:15

Zadejte čas příjezdu vlaku C:

0:10

Zadejte čas odjezdu vlaku C:

0:45

Z vlaku A lze přestoupit na vlaky B a C.

Z vlaku B lze přestoupit na vlak C.

Z vlaku C lze přestoupit na vlak B.

Zadejte čas příjezdu vlaku A:

12:00

Zadejte čas odjezdu vlaku A:

12:65

Nesprávný vstup.

Zadejte čas příjezdu vlaku A:

12:00

Zadejte čas odjezdu vlaku A:

12:abc

Nesprávný vstup.

Vzorová data:

[Download](#)

☐ Referenční řešení

2

14.10.2012 19:13:07

Download

Stav odevzdání:

Ohodnoceno

Hodnocení:

5.5000

Hodnotitel: automat

Program zkompilován

Test 'Základní tst s parametry podle ukázky': Úspěch

Dosaženo: 100.00 %, požadováno: 100.00 %

Max doba běhu: 0.006 s (limit: 2.000 s)

Celková doba běhu: 0.033 s

Využití paměti: 11720 KiB (limit: 18434 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test mezních hodnot': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Max doba běhu: 0.006 s (limit: 2.000 s)

Celková doba běhu: 0.037 s

Využití paměti: 11720 KiB (limit: 18434 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Kontrola ošetření nesprávných vstupů': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Max doba běhu: 0.024 s (limit: 2.000 s)

Celková doba běhu: 0.092 s

Využití paměti: 11720 KiB (limit: 18434 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test náhodnými hodnotami': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Max doba běhu: 0.006 s (limit: 2.000 s)

Celková doba běhu: 0.158 s

Využití paměti: 11720 KiB (limit: 18434 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00)

Celkové procentní hodnocení: 100.00 %

Bonus za včasné odevzdání: 0.50

Celkem bodů: 1.00 * (5.00 + 0.50) = 5.50

SW metriky:

Funkce:

2

--

-- --

Řádek kódu:

91 45.50 ± 23.50

69 main(int,char **)

Cykloomatická složitost:

65 32.50 ± 16.50

49 main(int,char **)

1	14.10.2012 19:06:25	Download
Stav odevzdání:	Ohodnoceno	
Hodnocení:	0.0000	

- **Hodnotitel: automat**
 - Chyba při základní kompilaci **[Zpřístupnit nápovědu (274 B)]**
- Celkové procentní hodnocení: 0.00 %
- Bonus za včasné odevzdání: 0.50
- Celkem bodů: $0.00 * (5.00 + 0.50) = 0.00$