

Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb (tzn. knihy, kalkulačky, mobily, apod.).

Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Po skončení soutěže bude pro každou úlohu váš poslední odevzdaný program automaticky otestován pomocí předem připravených 15 sad testovacích vstupních dat. Každá sada je tvořena jedním nebo několika testovacími vstupy. Jeden bod získáte za každou sadu vstupů, kterou váš program celou správně vyřeší – tzn. pro každý testovací vstup z této sady dá program správný výsledek ve stanoveném časovém a paměťovém limitu. Za každou úlohu tedy můžete získat 0 až 15 bodů.

Sady vstupů jsou navrženy tak, aby každé korektní řešení získalo nějaké body, i když je založeno na neefektivním algoritmu. Časové a paměťové limity zajišťují, že plný počet bodů získá pouze efektivní řešení. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

P-III-4 Asfaltistán

Program: `asfalt.pas / asfalt.c / asfalt.cpp`
Vstup: `asfalt.in`
Výstup: `asfalt.out`

V Asfaltistánu zjistili, že mají jedno velké území dosud nedotčené asfaltem. Rozhodli se to napravit tak, že napříč územím postaví silnici. Nicméně ministr financí si klade podmínky. Silnice musí být sjízdná pro auta a musí být postavena co nejlevněji. Je totiž krize a musí se šetřit.

Projektanti si rozdělili území na $R \times S$ čtverců o straně 1 km a pro zjednodušení si pro každý čtverec spočítali jeho průměrnou nadmořskou výšku. Zjistili, že silnice mezi dvěma sousedními čtverci je sjízdná, pokud je rozdíl jejich nadmořských výšek maximálně 1 obří sáh.

Navíc mohou stavět mosty a tunely – mostem je možno propojit dva čtverce ve stejném sloupci nebo řádku, pokud je jejich nadmořská výška shodná a nadmořská výška všech čtverců mezi nimi je nižší. Tunelem je také možno propojit dva čtverce o stejné nadmořské výšce ve stejném sloupci nebo řádku, jen nadmořská výška všech čtverců mezi nimi musí být vyšší.

Vaším úkolem je najít nejlevnější silnici mezi políčkem $(0, 0)$ a políčkem $(R - 1, S - 1)$. Silnice musí být na políčkách $(0, 0)$ a $(R - 1, S - 1)$ na povrchu, tedy ne na mostě ani v tunelu.

Poznamenejme, že na jednom políčku může jeden most/tunel končit a současně další začínat. Také si všimněte, že je teoreticky možné, že se na optimální cestě budou

dva mosty nebo tunely křížovat. To je povolené, šikovní asfaltistánsťi inženýři to dokáží vyřešit tak, aby se auta nesrážela.

Formát vstupu

Na vstupu dostanete na prvním řádku celá čísla R, S, c_S, c_M a c_T . Čísla c_S, c_M a c_T udávají ceny za postavení jednoho políčka silnice, mostu a tunelu (v asfaltových dolarech).

Cena za políčko mostu nebo tunelu se počítá jen za políčka *uvnitř*, čili začátek a konec tunelu/mostu se nepočítá.

Na každém z dalších R řádků dostanete S čísel $h_{i,j}$ oddělených mezerou – nadmořské výšky jednotlivých políček v obřích sázích. Na i -tém řádku v j -tém sloupci je uvedeno číslo $h_{i,j}$.

Formát výstupu

Na první řádek výstupu vypište celkovou cenu silnice v asfaltových dolarech. Na následující řádky vypište trasu silnice; na každý řádek dvojici (i_k, j_k) udávající políčko, přes které silnice prochází. Políčka, která se nachází pod mostem nebo nad tunelem, vynechte. První souřadnice vždy udává řádek a druhá sloupec. Pokud řešení neexistuje, vypište na jediném řádku výstupu slovo **NEEXISTUJE**.

Všimněte si, že celková cena může být značně velké číslo, které se do 32-bitové proměnné nemusí vejít.

Velikost vstupu

Ve všech vstupech použitých při testování platí $1 \leq R, S \leq 1\,000, 1 \leq c_S, c_M, c_T \leq 1\,000\,000$ a $0 \leq h_{i,j} \leq 1\,000\,000$.

Ve vstupech za alespoň 5 bodů mimo to platí $1 \leq R, S \leq 50$.

Ve vstupech za alespoň 5 bodů také platí, že alespoň jedna z nejlevnějších silnic neobsahuje žádný tunel ani most. Tyto vstupy nemusí být různé od těch uvedených v předchozím odstavci.

Příklady

Vstup:

2 7 1 20 20
5 1 1 5 5 5 5
5 5 5 5 9 9 5

Výstup:

10
0 0
1 0
1 1
1 2
1 3
0 3
0 4
0 5
0 6
1 6

Vstup:
3 3 1 1 1
42 32 32
32 22 12
22 12 2

Výstup:
NEEXISTUJE

Vstup:
2 7 1 1 20
5 1 1 5 5 5 5
5 5 5 5 9 9 5

Výstup:
8
0 0
0 3
0 4
0 5
0 6
1 6

Vstup:
2 7 1 20 1
5 1 1 5 5 5 5
5 5 5 5 9 9 5

Výstup:
8
0 0
1 0
1 1
1 2
1 3
1 6

P-III-5 Básník Honzík II

Program: basnik.pas / basnik.c / basnik.cpp

Vstup: basnik.in

Výstup: basnik.out

Honzík poté, co jste mu včera pomohli pozvat Boženku na procházku, zjistil, že v sobě ukrývá velký básnický talent, a chtěl by po vás poradit ještě jednou. Rozhodl se totiž, že složí nejkrásnější báseň všech dob. Sousedovic Kája Máchů mu poradil, že v nejkrásnějších básních má každá sloka dva verše. Pak od svého kamaráda Jardy Seifertíka zjistil, že každá správná básnička má mít přesně N slok. Nakonec se od Elišky Hezkohorské dozvěděl, že tyto básně mají takzvaný cyklický sdružený rým. To znamená, že mají schéma $ab\ bc\ cd\ de\ ef\ \dots\ yz\ za$, tedy že druhý verš každé sloky se rýmuje s prvním veršem té následující; výjimku tvoří druhý verš poslední sloky, který se rýmuje s prvním veršem první sloky.

Honzík má skutečně bohatou fantazii a navíc i váš včerejší program, takže pro něj nebyl problém vymyslet jednotlivé sloky i jejich pozici v básni. Co ale čert nechtěl, od některých slok vymyslel několik různých variant, které sice vyjadřují stejnou myšlenku, ale obsahují jiné dvojice rýmů.

A právě s výběrem jednotlivých slok do výsledné básně by chtěl pomoci. Pro každou z N výsledných slok dostanete S_i variant, které Honzík vymyslel, a vašim úkolem je sestavit básničku tak, aby měla cyklický sdružený rým.

Abyste si nemuseli lámat hlavu s tím, které verše se rýmují a které ne, očísloval Honzík všechny možné rýmy přirozenými čísly. Každou sloku pak popsal dvojicí (x, y) obsahující čísla rýmů v obou verších. Za slokou (x, y) se tedy může vyskytovat sloka (a, b) právě tehdy, když $y = a$.

Soutěžní úloha

Napište program, který pro každou sloku číslo i ($1 \leq i \leq N$) dostane S_i možných variant a z nich sestaví básničku, která splňuje požadované schéma rýmů.

Formát vstupu

První řádek vstupního souboru obsahuje přirozené číslo N , které udává počet slok básničky. Následuje N skupin řádků, přičemž skupina i začíná řádkem obsahujícím číslo S_i následované S_i řádky, každý s dvěma čísly A a B . Číslo A udává rým, na který končí první verš varianty, a číslo B udává rým, na který končí druhý verš.

Formát výstupu

Program vypíše do výstupního souboru buď řádek s řetězcem NEEXISTUJE, pokud básničku sestavit nelze, nebo N řádek, přičemž i -tý řádek bude obsahovat číslo k ($1 \leq k \leq S_i$), které určuje vybranou variantu pro sloku číslo i . Pokud existuje více řešení, vypíše libovolné z nich.

Velikost vstupu

Všechny vstupy použité při testování mají $N \leq 1000$, $1 \leq S_i \leq 1000$ pro všechna i a $1 \leq A, B \leq 10^9$.

Ve vstupech celkem ohodnocených 5 body mimo to platí $N \leq 10$ a $S_i \leq 10$.

Příklady

Vstup:

5
1
1 1
2
1 5
1 6
2
5 2
6 3
1
3 8
3
10 2
8 1
4 2

Výstup:

1
2
2
1
2

*Jednotlivé sloky básničky budou mít
rýmy 11 16 63 38 81.*

Vstup:

2
1
2 1
1
1 3

Výstup:

NEEXISTUJE