

Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb a přiděleného počítače (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 Tlustý Santa

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

Zanedlouho bude duben. Jak víte, v dubnu již supermarkety začínají prodávat výrobky s vánoční tematikou a na severním pólu začínají mít plné ruce práce s přípravou Vánoc. Santa Claus právě řeší důležitý problém: jak se dostat od krbu (příp. od okna nebo od dveří, když v místnosti krb není) k vánočnímu stromečku. A to věru není vůbec snadné, protože lidé teď mají ve svých pokojích plno nábytku a jiného nepořádku. Jelikož Santa nedávno trocha přibral a získal tak krásný obdélníkový půdorys, je opravdu zázrak, že se ke stromečku vůbec dostane. (Prozradíme vám tajemství: občas si při tom pomůže trochou vánoční magie.)

Vaším úkolem bude pomoci Santovi nalézt **nejkratší** cestu ke stromečku.

Soutěžní úloha

Obdélníkovou místnost si můžeme představit jako mřížku tvořenou $r_1 \times s_1$ čtvercovými políčky. Některá políčka jsou prázdná, na jiných jsou umístěny překážky. Políčka si očíslováme od $(0, 0)$ v levém horním rohu do $(r_1 - 1, s_1 - 1)$ v pravém dolním rohu místnosti.

Santu si představíme jako obdélník s rozměry $r_2 \times s_2$. Santa je vždy umístěn tak, aby zakrýval přesně $r_2 \times s_2$ políček místnosti. Pohybuje se pouze čtyřmi směry, vždy rovnoběžně s osami mřížky, a to pokaždé o 1 políčko. Nemůže se otáčet, takže Santova strana délky r_2 je vždy rovnoběžná se stranou místnosti, která má délku r_1 .

Santa začíná v levém horním rohu, kde zabírá políčka od $(0, 0)$ po $(r_2 - 1, s_2 - 1)$. Aby mohl umístit dárky pod stromeček, potřebuje se dostat do protilehlého rohu místnosti, tedy zaujmout políčka od $(r_1 - r_2, s_1 - s_2)$ po $(r_1 - 1, s_1 - 1)$. Během své cesty nesmí Santa nikdy vstoupit na políčko s překážkou.

V některých situacích (které pro vás mají hodnotu 5 bodů) může Santa používat magii. Pomocí magie může najednou přesunout nejvýše m překážek do jiné dimenze a může tak vstoupit i na políčka, kde tyto překážky původně ležely. Magii dokáže Santa plynule přesouvat z jedné překážky na druhou, takže když mu najdete trasu, na níž bude v každém kroku překrývat nejvýše m překážek, bude moci po této trase projít.

Během své cesty nesmí Santa nikdy opustit místnost, a to ani pomocí magie.

Formát vstupu

Na prvním řádku vstupního souboru je pět celých čísel oddělených mezerami. Jsou to rozměry místnosti r_1, s_1 , rozměry Santy r_2, s_2 a množství dostupné magie m .

Následuje popis místnosti na r_1 řádcích, z nichž každý obsahuje řetězec tvořený s_1 znaky. Znak '.' (tečka) označuje prázdné políčko, znak 'X' (velké písmeno iks) překážku.

Můžete předpokládat, že na začátku není pod Santou žádná překážka.

Formát výstupu

Program vypíše do výstupního souboru jeden řádek s řetězcem znaků, který představuje posloupnost kroků Santy z výchozí pozice do cíle. K popisu cesty použijeme světové strany: přesun o řádek nahoru budeme označovat 'S' (sever), přesun o sloupec doleva 'Z' (západ); opačné směry budou 'J' (jih) a 'V' (východ).

Pokud neexistuje žádná posloupnost kroků, která by dovedla Santu do protilehlého rohu místnosti, program vypíše jeden řádek a s textem „Santa je chudak.“ (bez uvozovek).

Jestliže existuje více vyhovujících posloupností, vypište nejkratší z nich. Pokud existuje více vyhovujících posloupností téže minimální délky, můžete vypsát jednu libovolnou z nich.

Zdůrazňujeme, že pro vstupní data s $m > 0$ není třeba hledat posloupnost kroků, při které Santa použije co nejméně magie celkově nebo co nejméně magie najednou. Důležité je pouze minimalizovat počet kroků, které Santa udělá.

Omezení

Testovací vstupy jsou rozděleny do 15 sad. Za každou sadu, kterou váš program celou správně vyřeší, dostanete jeden bod. Ve všech testovacích vstupech platí $1 \leq r_2 \leq r_1 \leq 2000$, $1 \leq s_2 \leq s_1 \leq 2000$ a $0 \leq m \leq r_2 s_2$. V žádném testovacím vstupu nebude současně platit $r_1 = r_2$ a $s_1 = s_2$. (Santa je vždy menší než místnost.) K získání plného počtu bodů musí váš program správně vyřešit všechny takovéto vstupy.

Následující tabulka podrobně popisuje, jak vypadají jednotlivé testovací vstupy. Zvláště upozorňujeme, že když váš program správně vyřeší všechny vstupy s $m = 0$, získá minimálně 10 bodů.

<i>číslo sady</i>	1–10	11	12–15		
<i>max. množství magie m</i>	0	1	libovolné		
<i>číslo sady</i>	1–2	3	4–6, 12	7–11, 13–15	
<i>max. rozměry Santy $r_2 \times s_2$</i>	1×1	10×10	$10 \times s_1$	libovolné	
<i>číslo sady</i>	1, 4	2, 5, 7	8, 13	9, 14	3, 6, 10–12, 15
<i>max. $r_1 \times s_1$</i>	20×20	200×200	$1\,000 \times 1\,000$	$1\,500 \times 1\,500$	$2\,000 \times 2\,000$

Příklady

Vstup:

2 2 1 1 0

..
..

Také výstup „JV“ je správný.

Výstup:

VJ

Vstup:

5 8 2 2 0

.....
...XXXX
..X.....
.....
.....X..

Výstup:

JJJVVVSVVJ

V tomto případě se jedná o jedinou nejkratší posloupnost kroků.

Vstup:

5 8 2 2 0

.....
...XXXX
.XX.....
.....
.....X..

Výstup:

Santa je chudak.

Oproti předchozímu příkladu přibyla překážka na (2, 1). Tu Santa nedokáže obejít.

Vstup:

5 8 2 2 1

.....
...XXXX
.XXXX...
..X...X.
.....X..

Výstup:

JJJVVVSVVJ

Nyní přibyla ještě více překážek, Santa ale může použít magii. Všimněte si, že nevyhovuje cesta „JJJJVVVVV“ – krok před cílem by totiž Santa stál najednou na dvou překážkách a na to nemá dostatek magie.

P-III-5 Úřad

Program: `urad.pas` / `urad.c` / `urad.cpp`

Vstup: `urad.in`

Výstup: `urad.out`

Ministerstvo pro boj s byrokracií má n úředníků. Jsou označeni čísly $1, 2, \dots, n$. Ministr má číslo 1, ostatní čísla nemají žádný zvláštní význam.

Každý úředník kromě ministra má svého šéfa (tj. přímého nadřízeného). Úředník a je nadřízeným úředníka b , jestliže a je buď přímo šéfem úředníka b , nebo je nadřízeným šéfa b . (Jinými slovy, pokud a je buď šéfem b , nebo šéfem šéfa b , nebo ...) Ministerstvo má stromovou strukturu, takže žádný úředník není nadřízeným sebe samého.

Každý úředník včetně ministra řídí jedno oddělení ministerstva. Do tohoto oddělení patří on sám a také všichni úředníci, pro které je nadřízeným. (Občas se ovšem stane, že oddělení je tvořeno jen jedním člověkem.)

Bojovníkům proti byrokracii je třeba pravidelně zvyšovat mzdu, aby dobře bojovali. Ministr to prováděl následujícími příkazy:

1. Je-li mzda úředníka u ostře menší než x , bude mu zvýšena o y .
2. Je-li průměrná mzda v oddělení, které řídí úředník u , ostře menší než x , bude každému úředníkovi z tohoto oddělení mzda zvýšena o y .

Po čase přišly volby a s nimi také nový ministr. Mladý, nezkušený, plný ideálů. Jinými slovy, vůbec nevěděl, jak to chodí. A proto začal vydávat také příkazy třetího typu:

3. Určí nejnižší mzdu úředníka v oddělení, které řídí úředník u . Tuto mzdu nastav každému v tomto oddělení.

Jednou se stalo, že uklízečka omylem vyhodila papír, na kterém měla mzdová účtárna zaznamenány aktuální mzdy. Zůstaly zachovány pouze údaje o mzdách všech úředníků na začátku roku a seznam ministerských příkazů v chronologickém pořadí. Vaším úkolem je zjistit aktuální mzdy.

Soutěžní úloha

Napište program, který dostane popis ministerstva, počáteční mzdu každého úředníka a posloupnost ministerských příkazů. Program vypíše mzdu každého úředníka po provedení všech příkazů.

K získání 12 bodů není třeba implementovat zpracování příkazů nového ministra.

Formát vstupu

Na prvním řádku vstupního souboru je číslo n určující celkový počet úředníků. Následuje n řádků, přičemž i -tý z nich popisuje úředníka číslo i . Každý z těchto řádků obsahuje následující údaje: počáteční mzdu z_i , počet přímých podřízených p_i a seznam jejich čísel. Všechny údaje jsou odděleny mezerami.

Můžete předpokládat, že popis hierarchie ministerstva je korektní. Každý úředník kromě ministra se jako přímý podřízený nachází na vstupu právě jednou, a to takovým způsobem, aby nevznikl žádný cyklus.

Následuje řádek s číslem q , udávajícím počet ministerských příkazů. Zbytek vstupu tvoří q řádků. Každý z nich popisuje jeden příkaz v tom pořadí, v jakém byly prováděny. Popis každého příkazu má jeden z těchto možných tvarů:

- $1 u x y$: odpovídá možnému zvýšení mzdy úředníka u .
- $2 u x y$: odpovídá možnému zvýšení mzdy celého oddělení, které řídí úředník u .
- $3 u$: odpovídá příkazu nového ministra.

Formát výstupu

Program vypíše do výstupního souboru n řádků, na každém z nich bude jedno celé číslo. Číslo na i -tém řádku představuje konečnou mzdu i -tého úředníka.

Omezení

Testovací vstupy jsou rozděleny do 15 sad. Za každou sadu, kterou váš program celou správně vyřeší, dostanete jeden bod. Ve všech testovacích vstupech platí pro počáteční mzdy $0 \leq z_i \leq 1000$. Ve všech testovacích vstupech platí v každém příkazu: $1 \leq u \leq n$, $0 \leq x \leq 10^9$, $0 \leq y \leq 1000$.

Jednotlivé testovací vstupy obsahují různý počet úředníků n a různý počet příkazů q . Liší se také následujícími parametry: hloubka hierarchie h , počet provedení druhého příkazu v_2 , zda se na vstupu vyskytují také příkazy nového ministra.

(Hloubka ministra je 0, jeho přímí podřízení mají hloubku 1, jejich přímí podřízení 2, atd. Hloubka hierarchie je maximum z hloubek úředníků, kteří ji tvoří. Hodnota v_2 udává pouze počet těch příkazů druhého typu, které skutečně změní mzdy úředníků.)

Následující tabulka přehledně popisuje, jaké parametry mají jednotlivé testovací vstupy.

číslo sady	1–3	4	5–6	7–11	12	13–15
n	1 000	100 000	100 000	100 000	100 000	100 000
q	1 000	100 000	100 000	100 000	100 000	100 000
h	1 000	50	50	100 000	100 000	100 000
v_2	1 000	50	100 000	50	100 000	100 000
příkazy 3 u	ne	ne	ne	ne	ne	ano

Příklady

Vstup:

7
10 2 2 6
5 2 3 4
1 0
1 1 5
1 0
2 1 7
1 0
5
2 4 1 5
1 5 2 7
2 2 20 20
2 2 20 100
2 6 2 2

Výstup:

10
25
21
21
28
4
3

První operace se neprovede, neboť průměr daného oddělení je 1, což není méně než 1. (Dané oddělení tvoří úředníci 4 a 5. Oba mají v této chvíli mzdu 1.) Druhá a třetí operace se provedou. Čtvrtá operace se neprovede. Pátá se provede, neboť průměr je 1,5, což je méně než 2.

Vstup:

3
10 2 2 3
4 0
7 0
1
3 1

Výstup:

4
4
4

Ministr má mzdu 10, jeho podřízení mají mzdu 4 a 7. Provede se příkaz třetího typu pro ministra. Nejnižší mzda v jeho oddělení je 4, takže mzdu 4 dostanou všichni.