

ProFIIT 2006 – zadania problémov pre korešpondenčné kolo súťaže

Úvod

Poznáte film *Policajt v škôlke*? Arnold Schwarzeneger sa ako policajt v prestrojení za učiteľa dostal do materskej škôlky. Snáď náročnejšie ako práca policajta bolo zvládnuť triedu malých detí... Vy ste sa dostali do podobnej situácie. Nie ste síce policajtom, ale zhodou okolností vás presvedčil chorý priateľ, aby ste ho zastúpili v jeho práci. Ten hravo zvláda triedu v škôlke plnej nadpriemerne inteligentných a nadaných detí. Na jeho miesto nastupujete v mylnom presvedčení, že predsa s nadanými deťmi to bude hračka. Že je to oveľa horšie ako by ste si predstavili len v tom najhoršom sne, sa presvedčíte už v prvý deň. Jedinou vašou záchranou bude počítač a vaša znalosť programovania. Zvládnete to?

Upozornenie: Pre všetky problémy platí takéto pravidlo: posledný riadok výstupu má byť odriadkovaný presne tak, ako všetky ostatné riadky výstupu (t.j. na konci každého riadku je ENTER).

Šifra

Na prvý deň v škôlke máte pripravenú úlohu, o ktorej predpokladáte, že zaberie deťom celý deň. Cieľom je rozšifrovať zašifrovaný text. Po krátkom predstavení sa hneď snažíte zaujať deti touto úlohou. Porozprávate im o tom, ako dôležité je šifrovať správy, najmä ak sa o nich nemá nikto cudzí dozvedieť. Potom im rozdáte dvojice papierikov, z ktorých jeden obsahuje text a druhý je pokreslený krížikmi a bodkami. Na papieriku s krížikmi a bodkami majú deti vystrihnúť okienka na miestach, kde sú bodky. Takto vystrihnutý papierik majú priložiť na papierik s textom. Tým prekryjú časť textu a cez vystrihnuté okienka prečítajú rozšifrovaný text. Deti dlho vystrihovanie nebaví, a tak si položia kartičky vedľa seba a z hlavy chľbia rozšifrované texty. Aby ste si zachovali tvár a ukázali, že vy to vyriešite ešte rýchlejšie ako oni, rozhodnete sa uplatniť vaše programátorské schopnosti. Vašou úlohou sa teda stane napísanie programu, ktorý bude zašifrované texty rozšifrovať.

Vstup

Vstup je treba ako v každej úlohe načítať zo štandardného vstupu (klávesnice). Prvý riadok vstupu bude obsahovať celé číslo n ($1 \leq n \leq 100$) vyjadrujúce počet úloh. Každá úloha má v prvom riadku dve čísla p, r ($1 \leq p, r \leq 100$). Potom nasleduje r riadkov pôvodného textu, každý obsahujúci p znakov s ASCII hodnotami z ($32 \leq z \leq 126$). Hneď za pôvodným textom je uvedený popis papierika, ktorým sa má text prekryť. Papierik má rovnaké rozmery ako text a obsahuje len znaky # (mriežka) – nepriehľadná časť papierika a . (bodka) – diera, t.j. časť papierika, cez ktorú vidno.

Výstup

Výstupom programu je n riadkov, každý obsahujúci text čítaný po riadkoch, ktorý vidno, keď prekryjeme pôvodný text papierikom s dierami.

Vzorový vstup

```
1
41 3
Kde bolo, tam bolo, za zamkom, kde byvala
  Snehulienka, za palacom, kde byvala Popo
luska, bol raz jeden velky prevelky hrad.
.#####.#####.#####.#####.#####
#####.#####
#####.#####.#####.##
```

Vzorový výstup

Kolobezka

Kontrolujte zátvorky!

Veľmi nadané deti po vás stále chcú nejaké hlavolamy a zaujímavé úlohy. Nič vás nenapadá, až vám padne zrak na vytlačený program, ktorý sa váľa na zemi. Je plný zátvoriek. Guľatých, hranatých, zložených... „Kontrolujte zátvorky!“, vykričíte a vyrovnávate pokrčený papier s programom. „Každý výraz je správne ozátvorkovaný, ak všetky podvýrazy, ktoré obsahuje, sú správne ozátvorkované, teda majú ľavú zátvorku, dobre ozátvorkovaný výraz a pravú zátvorku“, vysvetľujete. „Napríklad, $(a + (b =) (a))$ je správne ozátvorkovaný výraz, ale výraz $(a +) b =) (a ()$ nie je správne ozátvorkovaný.“

Predpokladajte, že v programoch, ktoré majú deti kontrolovať, sú takéto typy zátvoriek:

- ()
- { }
- []
- < >
- (* *)

Dvojica $(*$ je považovaná za jeden symbol.

To, že ste vymysleli úlohu pre deti, však nie je všetko. Ak vám totiž dve deti povedia iný výsledok, musíte ich vedieť rozsúdiť. Vašou úlohou je teda napísať program, ktorý vyrieši úlohu a skontroluje, či je ozátvorkovanie správne. Ak nie je správne, je potrebné určiť, na ktorom symbole výrazu je chyba, aby ste ju mohli ukázať deťom. Nezabudnite, že zátvorka $(*$ je považovaná za jeden symbol, tak isto ako $*)$. Každý symbol, ktorý nie je zátvorkou, sa považuje tiež za jeden symbol.

Vstup

V prvom riadku vstupu je číslo n ($1 \leq n \leq 10000$) vyjadrujúce počet výrazov, ktoré je treba skontrolovať. Pre každý výraz je najprv na samostatnom riadku číslo d ($1 \leq d \leq 2500$) vyjadrujúce počet znakov vo výraze, nasledované samotným výrazom tiež na samostatnom riadku. Vo výraze sa môžu vyskytovať znaky s ASCII hodnotou z ($32 \leq z \leq 126$).

Výstup

Výstupom je n riadkov, každý je výsledkom kontrol príslušného výrazu na vstupe. Výsledkom je ANO, ak je výraz dobre ozátvorkovaný. V prípade, že výraz nie je dobre ozátvorkovaný, výsledkom je NIE nasledované jednou medzerou a číslom vyjadrujúcim pozíciu chyby – poradové číslo (číslované od 1) prvého symbolu, ktorý ozátvorkovanie pokazil. Ak výrazu chýbajú ukončovacie zátvorky, výsledkom je NIE nasledované jednou medzerou a číslom znamenajúcim počet symbolov vo výraze plus 1. Pozn. slová *znak* a *symbol* nie sú v tomto príklade synonymá.

Vzorový vstup

```
3
15
<a(*(*c*))]r+ +>
21
<[ak](-)r>o(*vt{az}*)
1
{
```

Vzorový výstup

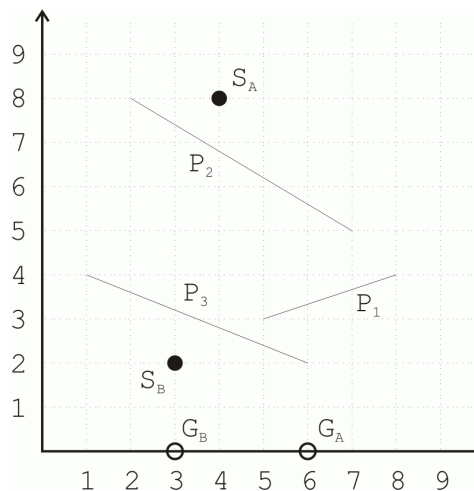
```
NIE 7
ANO
NIE 2
```

Guľôčky

Takmer všetky deti sa vám podarilo zamestnať. V kúte však sedí malé dievčatko, ktoré smutne pozerá pred seba. Keď ju oslovíte, podarí sa vám z nej dostať len to, že sa volá Zuzka a rada sa hrá s guľôčkami. Zátvorky a iné úlohy ju rozhodne nezaujímajú. Nezostáva vám nič iné, ako pre ňu vymyslieť úlohu.

Nakoniec vás napadne zaujímavá úloha: Na stenu nalepíte tri paličky a nakreslíte dve bodky. Nákres je na obr. 1, kde paličky sú označené P_1 , P_2 a P_3 a bodky S_A a S_B . „Kam na dlážku dopadne guľôčka, keď ju pustíš z bodu S_A a kam dopadne guľôčka, keď ju pustíš z bodu S_B ?“

Zuzka sa zamyslí. Žmúľá guľôčku a po chvíli sa uistí, že môže uvažovať ideálne podmienky a zanedbať horizontálne odchýlky spôsobené kinetickou energiou. Potom na zem nakreslí bodky a označí ich G_A a G_B a nakoniec priloží guľôčku k bodu S_A a pustí ju. Guľôčka dopadne približne do bodu G_A a keď spustí guľôčku z bodu S_B , zeme sa dotkne takmer v bode G_B . Zuzka je nadšená a vy jej musíte vytvárať ďalšie situácie, ktoré má ona riešiť. Čím sú však zložitejšie, tým sú odchýlky od ideálneho prípadu väčšie a vám nezostáva iné ako napísať program, ktorý povie Zuzke, kam by guľôčka dopadla v ideálnom prípade.



Obr. 1 Príklad rozloženia paličiek a bodiek v úlohe Guľôčky.

Vstup

Prvý riadok vstupu obsahuje celé číslo n ($1 \leq n \leq 1000$) predstavujúce počet blokov programu. Každý blok opisuje jedno rozloženie paličiek a bodiek. Prvý riadok bloku obsahuje číslo p ($0 \leq p < 1000$) vyjadrujúce počet paličiek. Potom v bloku nasleduje p riadkov, každý opisujúci súradnice koncov paličky x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 10000$) oddelené jednou medzerou. Žiadna palička nie je vodorovná a žiadne dve paličky sa nepretínajú, ani nedotýkajú. Ďalší riadok bloku obsahuje číslo b ($0 < b \leq 10000$) vyjadrujúce počet bodiek. Potom nasleduje b riadkov, každý obsahujúci súradnice x, y bodu ($0 \leq x, y \leq 10000$), z ktorého sa spúšťa guľôčka. Predpokladajte, že všetky x -ové súradnice, t.j. ako koncov paličiek tak aj bodov, z ktorých sa guľôčka spúšťa, sú navzájom rozdielne.

Výstup

Výstupom bude x -ová súradnica bodu, kam spadne guľôčka zo vstupu, každá v samostatnom riadku.

Ukážkový vstup

```
1
3
5 3 8 4
2 8 7 5
1 4 6 2
2
4 8
3 2
```

Ukážkový výstup

```
6
3
```

Farebné podpostupnosti

Deti sú už unavené z riešenia úloh a začínajú sa nudiť. Jednému z nich napadne, že nebudú stále riešiť úlohy, ktoré vy vymyslíte im, ale oni dajú úlohu vám. Dohodnete teda aspoň to, že budete môcť používať počítač. Deti si hodnú chvíľu šepkajú až nakoniec víťazoslávne vyjde z hlúčka Janko a rozostaví pred vami rad farebných kociek a pod neho druhý, kratší rad kociek. „Koľkokrát sa tá menšia farebná postupnosť nachádza v tej väčšej?“

Aby ste mohli úlohu vyriešiť v počítači, rozhodnete sa farby reprezentovať písmenami. Keď rozmýšľate, ako úlohu vyriešiť, opakujete si, čo je to podpostupnosť... Ak máme postupnosť $X = x_1 x_2 \dots x_m$, tak druhá postupnosť $Z = z_1 z_2 \dots z_k$ je podpostupnosťou postupnosti X , ak existuje striktné rastúca postupnosť $\langle i_1, i_2, \dots, i_k \rangle$ indexov z X takých, že pre všetky $j = 1, \dots, k$ platí, že $x_{i_j} = z_j$. Napríklad $Z = abdb$ je podpostupnosť $X = acbdab$ s príslušnými indexami $\langle 1, 3, 4, 6 \rangle$.

Úloha je teraz jasná. Napíšte program, ktorý zistí počet výskytov Z v X ako podpostupnosti, ktorá má každá rôznu postupnosť indexov.

Vstup

Prvý riadok vstupu obsahuje číslo n ($1 \leq n \leq 500$) vyjadrujúce počet blokov. Každý blok predstavuje jednu úlohu a pozostáva zo štyroch riadkov. Prvý riadok bloku obsahuje číslo x ($1 \leq x \leq 8000$) – dĺžku reťazca X , ktorý sa nachádza na druhom riadku bloku. Tretí riadok bloku obsahuje z ($1 \leq z \leq 100$) – dĺžku reťazca Z , ktorý sa nachádza na štvrtom riadku bloku. Reťazce X aj Z pozostávajú výlučne z malých znakov anglickej abecedy.

Výstup

Pre každú úlohu vypíšete do samostatného riadku počet rôznych výskytov Z v X . Pre každý vstup platí, že ani Z ani žiadna jeho predná časť ani žiadna jeho zadná časť nebude mať viac ako 10^{100} (10 na 100) rôznych výskytov v X .

Vzorový vstup

```
2
5
abaab
2
ab
10
ezwzqzdyzt
2
zz
```

Vzorový výstup

```
4
6
```

Ako funguje počítač?

Váš počítač, pomocou ktorého sa vám darí rýchlo riešiť hlavolamy, deti veľmi zaujme a pýtajú sa vás, ako funguje. Deti sa tvária, že rozumejú, čo sú to registre, pamäť aj inštrukcie počítača. Najradšej by váš počítač rozobrali, aby to všetko videli. Aby sa počítač nestal ich obeťou, vymyslíte pre nich úlohu, pomocou ktorej by si mohli počítač „ohmatať“ iným spôsobom.

Úloha znie takto: počítač má 10 registrov a 1000 slov pamäte. Každý register alebo slovo v pamäti obsahuje 3-ciferné celé číslo medzi 0 a 999. Inštrukcie sú zakódované 3-cifernými číslami a sú uložené v pamäti. Kódy jednotlivých inštrukcií sú nasledovné ($0 \leq a, b \leq 9$):

- 1ab znamená skonči
- 2ab znamená nastav register a na b
- 3ab znamená pripočítaj b do registra a
- 4ab znamená vynásob register a číslom b
- 5ab znamená nastav register a na hodnotu registra b
- 6ab znamená pripočítaj hodnotu registra b do registra a
- 7ab znamená vynásob register a hodnotou registra b
- 8ab znamená nastav register a na hodnotu z pamäte, ktorá má adresu v registri b
- 9ab znamená nastav hodnotu v pamäti, ktorej adresa je v registri b na hodnotu registra a
- 0ab znamená choď na pozíciu z registra a, ak hodnota v registri b nie je 0

Všetky registre a všetky slová pamäte obsahujú na začiatku hodnotu 000. Obsah pamäte je načítaný zo štandardného vstupu. Prvá inštrukcia, ktorá sa vykoná, je na adrese 0. Inštrukcie sa vykonávajú postupne, pokiaľ nie je vykonávaná inštrukcia skoku. Po inštrukcii z adresy 999 bude vykonaná inštrukcia 000. Všetky výsledky sú „orezané“ na 3-ciferné číslo použitím modulo 1000, teda použije sa zvyšok po delení výsledku číslom 1000. Úlohou je zistiť počet inštrukcií, ktoré sa vykonajú (započítava sa aj inštrukcia na skončenie programu). Program určite končí inštrukciou 100.

Samozrejme, že ak chcete deti skontrolovať, aj vy musíte túto úlohu vyriešiť.

Vstup

Prvý riadok vstupu obsahuje celé číslo n ($1 \leq n \leq 100$) vyjadrujúce počet úloh. Každá úloha je reprezentovaná blokom. Prvý riadok bloku obsahuje celé číslo k ($1 \leq k \leq 1000$), ktoré predstavuje počet inštrukcií, ktoré sa zapíšu do pamäte od adresy 0. V bloku nasleduje k riadkov, každý obsahuje jednu inštrukciu. Všetky ostatné slová v pamäti sa nastavujú na hodnotu 000.

Výstup

Výstupom je n riadkov, každý obsahujúci počet vykonaných inštrukcií pre príslušnú úlohu. Počet vykonaných inštrukcií je minimálne 1 a maximálne 2000000 pre jeden blok.

Vzorový vstup

3
12
239
249
253
445
442
349
445
442
349
634
053
100
12
239
249
259
445
442
349
445
442
349
634
053
100
5
662
317
257
572
100

Vzorový výstup

76
28
5

Ste lepší ako Gauss?

Pri nadaných deťoch si spomeniete na známu príhodu, ktorá sa rozpráva o Carlovi Friedrichovi Gaussovi – slávnom matematikovi. Keď bol ako chlapec v škole, učiteľ si chcel od detí oddýchnuť, a tak ich zamestnal úlohou, o ktorej očakával, že ju deti budú dlho riešiť. Mali spočítať všetky čísla od 1 po 100. Pokým ostatné deti pomaly spočítavali čísla, malý Gauss rýchlo došiel k správnejmu výsledku 5050. Všimol si, že čísla môžu byť zoskupené po dvoch tak, že ich súčet je vždy 101 ($1+100, 2+99, \dots$) a z toho zistil, že súčet čísel od 1 po k je $k(k+1)/2$.

Pre nadané deti ste vymysleli trochu zložitejšiu úlohu. Neuvažujete len intervaly od 1 po k , ale uvažujete aj také intervaly, ktoré môžu začínať aj iným prirodzeným číslom ako 1. Napríklad interval od 5 do 7 má súčet 18 ($5+6+7=18$). Interval od 3 do 6 má tiež súčet 18. Toto využijete na vytvorenie úlohy pre deti, ktorou je nájsť pre danú sumu s všetky intervaly, ktorých súčet je s .

Ako aj u ostatných úloh ju potrebujete rýchlo vyriešiť, aby ste deti skontrolovali. Vytvorte preto program na riešenie tejto úlohy.

Vstup

Prvý riadok vstupu obsahuje číslo n ($1 \leq n \leq 1000$) vyjadrujúce počet úloh. Potom nasleduje n riadkov, každý pre jednu úlohu. Každý z n riadkov obsahuje jedno celé číslo predstavujúce sumu s ($1 \leq s \leq 1000000000$).

Výstup

Pre každú z n súm má program do riadku vypísať jednotlivé intervaly, ktorých súčet dáva požadovanú sumu. Každý interval je v tvare $[a, b]$, kde a je začiatok ($a \geq 1$) a b je koniec intervalu (a aj b patria do intervalu), pričom $b > a$. Intervaly sú uzatvorené v okrúhlych zátvorkách a medzi intervalmi nie je medzera. Intervaly sú usporiadané od najmenšieho začiatku intervalu po najväčší začiatok intervalu (t.j. podľa čísel a).

Ukázkový vstup

```
3
18
32
99
```

Ukázkový výstup

```
( [3, 6] [5, 7] )
(
[4, 14] [7, 15] [14, 19] [32, 34] [49, 50] )
```

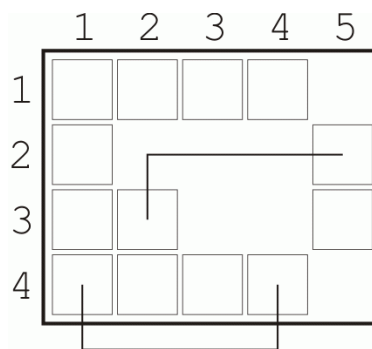
Chceme sa hrať!

Nie všetkým deťom sa ešte chce riešiť Vaše úlohy. Sú unavené a chcú sa hrať. Čakajú, že im vymyslíte nejakú novú zaujímavú hru. Najlepšia hra, ktorú sa vám podarí narýchlo vymyslieť, používa obdĺžnikovú dosku rozdelenú na štvorce. Každý štvorec môže, ale nemusí obsahovať kúsok hry. Príklad hry je na obr. 2.

Dôležitý aspekt hry je, či sa dva kúsky hry dajú spojiť cestou, ktorá spĺňa nasledujúce podmienky:

1. Cesta pozostáva len z rovných častí, ktoré sú buď horizontálne alebo vertikálne.
2. Cesta nekrižuje žiaden kúsok hry.

Je dovolené, aby cesta vyšla aj mimo hraciu plochu.



Obr. 2 Príklad hracej plochy v úlohe Chceme sa hrať!

Vstup

V prvom riadku vstupu je číslo n ($1 \leq n \leq 500$) vyjadrujúce počet blokov, každý pre jednu úlohu. Každý z n blokov obsahuje v prvom riadku dve celé čísla s, v ($1 \leq s, v \leq 75$) oddelené jednou medzerou. Číslo s predstavuje šírku a v predstavuje výšku hracej plochy. Potom v bloku nasleduje v riadkov, ktoré opisujú obsah hracej plochy. Každý riadok obsahuje s znakov. # ak je na danej pozícii kúsok hry, v opačnom prípade bodka. Nasleduje číslo k ($1 \leq k \leq 1000$) na samostatnom riadku znamenajúce počet dvojíc kúskov hry, pre ktoré je potrebné zistiť, či medzi nimi existuje cesta. Poslednú časť bloku predstavuje k riadkov, každý obsahuje štyri čísla x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq s$ a $1 \leq y_1, y_2 \leq v$) oddelené vždy jednou medzerou a predstavujúce súradnice dvoch rozdielnych kúskov hry, pre ktoré je treba zistiť, či medzi nimi existuje cesta.

Výstup

Pre každý z n blokov má program vypísať k riadkov. Každý riadok obsahuje buď slovo ANO, jednu medzeru a počet zahnutí cesty s najmenším počtom zahnutí, alebo slovo NIE, ak sa cesta medzi dvoma kúskami hry nájsť nedá.

Vzorový vstup

```
1
5 4
####.
#...#
##...#
####.
3
5 2 2 3
1 4 4 4
1 4 2 3
```

Vzorový výstup

```
ANO 1
ANO 2
NIE
```

Paličky

Večer, unavený po dni, ktorý ste síce zvládli, ale vás úplne vyčerpal, padnete unavený do postele a ihneď zaspíte. V sne sa vám prelínajú rôzne momenty, kde vidíte deti, počujete ako riešia úlohy, alebo sa dohadujú, akú úlohu dajú vám. Potom sa vám zdá, že vám dávajú ďalšiu úlohu, ktorú musíte vyriešiť. Spotený sa zobudíte a spomínate, čo to bola za úloha. Keď si spomeniete, uškrniete sa, pretože to je práve úloha, ktorú by ste im mohli niekedy dať.

Je takáto: V rade je p paličiek rôznej dĺžky. Ak sa pozrieme z ľavej strany, vidíme x paličiek. Ak sa pozrieme z pravej strany, vidíme y paličiek. Pretože majú rôznu veľkosť, zakrývajú jedna druhú. Vždy väčšia zakrýva menšie. Koľko je rôznych možností rozostavenia paličiek, ak máme dané čísla p , x , y ?

Samozrejme, ak chcete zadať úlohu deťom, musíte ju vedieť riešiť. Preto napíšete program, ktorý úlohu vyrieši.

Vstup

Prvý riadok obsahuje celé číslo n ($1 \leq n \leq 1000$) vyjadrujúce počet úloh. Nasleduje n riadkov, v každom sú porade čísla p , x , y ($1 \leq p, x, y \leq 13$) oddelené 1 medzerou. Číslo p je počet paličiek, ktoré sú poukladané do radu. x je počet paličiek, ktoré je vidieť z ľavej strany a y je počet paličiek, ktoré je vidieť z pravej strany.

Výstup

Pre každú úlohu vypíšete jedno číslo, ktoré reprezentuje počet možností usporiadania p paličiek, aby platilo, že z ľavej strany vidíme x a z pravej y paličiek.

Ukážkový vstup

```
3
4 1 2
2 1 1
7 3 2
```

Ukážkový výstup

```
2
0
675
```