

# Zadanie školského kola súťaže ZENIT v programovaní

Kategória A a B, 18.10.2022

V prípade nejasností konzultujte záložku **Pomoc** na stránke [zenit.ksp.sk](http://zenit.ksp.sk), alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

## A: A v koze bola...

10 bodov

Bola raz jedna galaxia, a v galaxii bola slnečná sústava, a v slnečnej sústave bola planéta, a na planéte bola krajina, a v krajine bolo mesto, a v meste bola ulica, a na ulici bola budova ZENITu, a v budove ZENITu bola podozrivo prázdna miestnosť, a v podozrivo prázdnej miestnosti bola obyčajná koza.

Podozrivá miestnosť mala tvar kvádra. Koľko bolo v miestnosti rohov?

### Vstup a výstup

Váš program nenačíta nič zo vstupu.

Vypíšte, koľko bolo v tejto miestnosti rohov.

### Príklad

vstup

výstup

*Tento výstup by dostal OK, ak by boli v miestnosti v budove ZENITu 3 rohy. To však nie je pravda.*

## B: Bus je na konečnej

15 bodov

Krtko cestuje každé ráno do školy a každý večer zo školy. Cestuje autobusom. Tomu ale trasa trvá o dosť dlhšie ako električke, ktorou kedysi chodil, a tak si chce "skrátit" čas, keďže vystupuje až na konečnej.

Začal rátať, koľko ľudí pristúpi a koľko ich vystúpi. Potom uvidí, koľko ľudí s ním ostane na poslednej zastávke.

Bohužiaľ, Krtko zaspal a nevie, ako sa menili počty ľudí v autobuse.

Ešteže sedíš v autobuse aj ty a počty vieš, takže vieš Krtkovi pomôcť!

### Vstup a výstup

Na prvom riadku vstupu dostanete číslo  $1 \leq n \leq 1000$  určujúce počet všetkých nastúpení/vystúpení pred konečnou zastávkou. Nasleduje  $n$  riadkov, kde na  $i$ -tom z nich je číslo  $p_i$  určujúce, koľko ľudí na  $i$ -tej zastávke nastúpilo (ak je kladné)/vystúpilo (ak je záporné).

Na začiatku je v autobuse 0 ľudí (ty ani Krtko sa do počtu nerátate). Tiež v autobuse nikdy nebude záporný počet ľudí, ani ich nebude viac ako 100.

Vášou úlohou je vypísať, koľko ľudí zostalo v autobuse na konečnej zastávke.

### Príklady

vstup

výstup

  
  

*Na prvej zastávke nastúpili dvaja ľudia, o zastávku ďalej jeden vystúpil. V autobuse teda zostal jeden človek*

vstup

```
5
15
-7
7
-14
1
```

výstup

```
2
```

### C: Cikcakovité zvieratká

20 bodov

Malý Matúško sa práve učí strihať. Maminka mu kúpila vlnkové a špicaté nožničky. Z poličky si zobral atlas zvietatiek. Najviac ho zaujali obrázky hadov a dážd'oviek. Zobral si svoje vlnkové a špicaté nožničky a *jedno* zo zvieratiek rozstrihal na niekoľko častí. Niektoré rezy vznikli špicatými nožničkami, niektoré vlnkovými. Potom jednotlivé časti zamiešal a niektoré pootáčal (v smere zľava doprava). Vašou úlohou je zistiť, či dané zvieratko bolo had alebo dážd'ovka (iné zvieratká Matúško nestrihal).

Celá dážd'ovka vyzerá tak, že má na okrajoch guľaté zátvorky otočené dovnútra a medzi nimi je niekoľko (aspoň 1) znakov =, podľa toho, aké má dlhé telo, napr. (=) alebo (===) alebo (=====) .

Celý had vyzerá tak, že na jednej strane má zátvorku otočenú dovnútra a na druhej má jazyk, teda znak < alebo >, ktorý je ale vždy otočený užšou časťou k hadovi. Telo je tvorené znakmi = (aspoň jedným). Had môže byť napr. (= < alebo (=== < alebo (===== <, ale aj >=) alebo >===) alebo >=====).

Rez vlnkovými nožničkami vznikne tak, že sa na danom mieste zvieratko rozdelí na dve časti a na pravý okraj ľavej časti sa pridá znak ( a na ľavý okraj pravej časti sa tiež pridá znak (. Napríklad (=====) môžeme rozdeliť na (===== ( a (=====).

Rez špicatými nožničkami vznikne tak, že sa na danom mieste zvieratko rozdelí na dve časti a na pravý okraj ľavej časti sa pridá znak < a na ľavý okraj pravej časti sa tiež pridá znak <. Napríklad (=====) môžeme rozdeliť na (===== < a <=====).

#### Vstup a výstup

Na prvom riadku vstupu sa nachádza jediné číslo  $1 \leq N \leq 1000$  – počet kúskov hada (alebo dážd'ovky). Nasleduje  $N$  riadkov, kde na každom sa nachádza jeden kúsok hada (alebo dážd'ovky).

Každý z  $N$  riadkov je dlhý aspoň 3, a obsahuje dva krajné kúsky hada (alebo dážd'ovky (alebo pozostatok strihania)), teda jeden zo znakov <, >, (, ) a aspoň jeden stredný kúsok (=). Každý riadok je dlhý najviac 50.

Súčet dĺžok všetkých riadkov nepresiahne 25 000.

Zistite, či je na vstupe had alebo dážd'ovka. Ak had, vypíšte **had**, ak dážd'ovka, vypíšte **dazdovka**. Je zaručené, že kúsky na vstupe vznikli z hada alebo dážd'ovky postupom popísaným vyššie.

#### Príklad

vstup

```
1
(=====)
```

výstup

```
dazdovka
```

*Matúško si ani raz nestrihol, zvieratko na vstupe je dážd'ovka.*

vstup

```
3
>=====)
)=====)
)=)
```

výstup

```
had
```

*Pred rozstrihaním mohlo stvorenie vyzeráť napríklad takto: >=====))=====))=), čo znamená, že to musel byť had.*

vstup

```
2
>=====<
(=====>
```

výstup

```
had
```

*Pred rozstrihaním mohlo stvorenie vyzeráť napríklad takto: (=====>>=====<, čo znamená, že to musel byť had.*

vstup

výstup

3
(==(
(==(
(==(

dazdovka
----------

Ak tretí kúsok otočíme, tak zistíme, že pred rozstrihaním mohlo stvorenie vyzeráť napríklad takto:  $((=((((=)))=))$ , čo znamená, že to musela byť dáždovka.

## D: Dobalím a dobehnem

32 bodov

Jožko nie je fanúšik ťažkých ruksakov. Najmä keď sa chystá na turistiku, chce, aby jeho dva ruksaky (na každú stranu jeden) obsahovali len najväčšie nevyhnutnosti, vďaka čomu ušetrí každý gram. Po zložitom rozhodovaní nakoniec usúdil, že z ruksakov vyhodí svoje staré vysvedčenia, vinohradnícke nožnice, 20 metrov medeného drôtu, niekoľko A4 papiera (to mu hádam v prípade potreby požičajú kamaráti), viacero reklamných externých nabíjačiek...

Vďaka týmto opatreniam má teraz ruksaky, ktoré zvládne nie len zdvihnúť, ale s nimi aj utekať na vlak. Neskôr si však uvedomil, že si zabudol zbaliť svoju sadu šrobovákov, maliarsku lepiacu pásku a plavecké okuliare. Čo ak bude jeden z jeho kamarátov potrebovať na turistike otvoriť svoj notebook? Alebo sa pôjdu kúpať do horského potoka?

Jožko usúdil, že je neprípustné, aby bez týchto 3 predmetov na turistiku šiel. No zároveň nechce, aby jeho ruksaky boli ťažšie, než zvládne odniesť. Keďže jeho vlak odchádza už o pol hodinu a bežnému človeku trvá chôdza na stanicu 20 minút, Jožko má už len len 20 minút na to, aby zistil, či sa veci do ruksakov zmestia. Pomôžte mu zistiť, či na turistiku môže ísť.

### Vstup a výstup

V prvom riadku vstupu sú dve celé čísla  $A$  a  $B$ : koľko voľného miesta má Jožko v jednotlivých ruksakoch.

V druhom riadku vstupu sú tri celé čísla  $s$ ,  $p$  a  $o$  - koľko miesta zaberajú šrobováky, páska, a okuliare.

Všetky čísla na vstupe sú v rozmedzí 1 až 1 000.

Ak vie Jožko naraz zmestiť všetky tri veci do svojich dvoch ruksakov, vypíšete **Ano**. Inak vypíšete **Nie**.

### Príklady

vstup

výstup

15 24
11 12 13

Ano
-----

Jožko zmestí šrobováky a okuliare (11 + 13) do druhého ruksaku, a pásku (12) do prvého.

vstup

výstup

20 15
1 2 30

Nie
-----

Nech sa Jožko snaží ako chce, okuliare (30) do žiadneho ruksaku nenapchá.

## E: Extrémne masívne zadanie

35 bodov

Merlin a Duško sú udatní bojovníci, ktorí už skolili veľa príšer. Keď zabíjajú nejakú príšeru, tá pri tom hrozne kričí, z čoho potom bojovníkom píska v ušiach. Najhoršie je, že výkriky príšer sú veľmi komplikované. Bojovníci by chceli, aby príšera radšej kričala proste len AAAAAAAAAAAAAAAAAA... Používajú preto špeciálne zbrane - keď Merlin príšeru *dingne* svojím mečom, vie zmeniť vo výkriku príšery jeden konkrétny výskyt písmena na ľubovoľné iné. *Dongnutím* kladiva si zas Duško vie vybrať nejaké písmeno z výkriku a všetky jeho výskyty posunie o 1 smerom k A.

Bojovníkov ich útoky stoja rôznu energiu a na zabitie príšery jej chcú dokopy minúť čo najmenej. Pomôžte vymyslieť Merlinovi a Duškovi stratégiu, pri ktorej bude príšera kričať len písmená 'A'.

### Vstup a výstup

Na prvom riadku vstupu sú 2 čísla, množstvo energie spotrebovanej na *Ding* a *Dong*.

Na druhom riadku je reťazec tvorený veľkými písmenami anglickej abecedy - aktuálny krik príšery.

Vypíšte jediné číslo - minimálny počet energie minutej na to, aby príšera kričala 'AAAAAAA...' - reťazec tvorený len znakmi 'A'.

### Obmedzenia

Ceny dingů a dongů sú kladné celé čísla neprevyšujúce 150 000.

Vstupy sú rozdelené do 5 sád, platia pre ne nasledovné obmedzenia na dĺžky kriku príšery: 10, 100, 100, 100 000, 100 000.

Navyše v druhej a štvrtej sade príšera kričí iba písmená 'A' a 'B'.

Pozor, výsledok môže byť pomerne veľké číslo, no vždy sa zmestí do 64-bitovej premennej.

### Príklady

vstup	výstup
<div style="border: 1px solid black; padding: 5px;">2 4 ZJEMVAS</div>	<div style="border: 1px solid black; padding: 5px;">12</div>

*Najviac sa oplatí aby Merlin dingol všetky písmenká okrem Á-čka*

vstup	výstup
<div style="border: 1px solid black; padding: 5px;">2 4 BABABABAZ</div>	<div style="border: 1px solid black; padding: 5px;">6</div>

*Merlin dingne Z a Duško dongne všetky B.*

## F: Fakt veľa mostov

45 bodov

Po poklese cien stavebných materiálov KMS (Komisia Mostového Stavania) rozhodla, že sa budú stavať mosty. Fakt veľa mostov.

Navrhli  $N$  mostov, ktoré môžu postaviť cez miestny potok. Potok tečie zľava doprava a na jeho hornom aj dolnom brehu je  $N$  križovatiek. Každý most je určený tým, ktoré dve križovatky (jednu na dolnom a jednu na hornom brehu) spája. Každá križovatka je vhodná na spojenie s práve jednou križovatkou na opačnom brehu.

Problém však nastáva, keď sa mosty majú križovať - teda ak by sa mal postaviť most medzi križovatkami  $a$  a  $b$ , a zároveň  $x$  a  $y$ , pričom  $a < b$  a zároveň  $x > y$ . Napríklad most medzi 3 a 8 by sa križoval s mostom medzi 4 a 5.

KMS si za návrhmi pevne stojí. Zistite, koľko najviac mostov vedia postaviť, aby sa žiadne z nich nekrižovali.

### Vstup a výstup

V prvom riadku je číslo  $N$  - počet návrhov na mosty, ako aj počet križovatiek na hornom a dolnom brehu potoka.

V každom z ďalších  $N$  riadkov sú dve čísla  $a$  a  $b$  - návrh na postavenie mosta medzi križovatkou  $a$  na dolnom brehu a križovatkou  $b$  na hornom brehu. Križovatky sú očíslované od 1 po  $N$  zľava doprava. Každé číslo od 1 po  $N$  sa objaví práve raz ako  $a$  a práve raz ako  $b$ .

Vypíšte jedno číslo - najväčší počet mostov, ktoré sa dajú naraz postaviť bez toho, aby sa križovali.

V prvej sade platí  $1 \leq N \leq 10$ .

V druhej sade platí  $1 \leq N \leq 1000$ .

### Príklady

vstup	výstup
<div style="border: 1px solid black; padding: 5px;">9 1 1 3 2 2 3 4 9 5 6 7 8 6 7 8 5 9 4</div>	<div style="border: 1px solid black; padding: 5px;">5</div>

Vieme postaviť napríklad prvý, druhý, piaty, šiesty a siedmy most. Nedá sa postaviť šesť mostov bez toho, aby sa niektoré križovali.

## G: Goniometria

45 bodov

Iveta mala mať zajtra prednášku o goniometrii. Nanešťastie však pri venčení svojho polárneho medvedíka prechladla, a tak ju musí zrušiť.

Keďže je to prvá prednáška dňa, nestačí len nalepiť oznam na dvere - tí, čo si ho prečítajú a zistia, že prednáška nie je, by už premárnili šancu túto informáciu využiť a ostať dlhšie v posteli.

Rozhodla sa teda, že svojim študentom napíše emailovú správu. To má výhodu, že email si každý študent ešte v ten večer zaručene prečíta. Má to však nevýhodu, že Iveta musí napísať každému študentovi, ktorý chodí na jej prednášky, po jednom. A to je kopa práce.

Iveta však vie, že sa niektorí jej študenti medzi sebou rozprávajú. Ak sa študent A rozpráva so študentom B, a Iveta pošle email napríklad študentovi A, tak ten to povie B. Ak sa B navyše rozpráva so študentkami C a D, tak im o tom povie, a tak ďalej. Takto sa informácia o zrušenej prednáške vie rozšíriť k viacerým študentom vďaka jedinému emailu.

Pomôžte Ivete - zistite, koľko najmenej emailov musí rozposlať, aby sa o zrušenej prednáške dozvedeli všetci jej študenti.

### Vstup a výstup

V prvom riadku vstupu je číslo  $N$  - počet študentov, ktorí chodia na Ivetinu prednášku. Sú očíslovaní od 1 po  $N$ .

V druhom riadku je číslo  $I$  - počet dvojíc študentov, o ktorých Iveta vie, že sa navzájom rozprávajú.

V každom z nasledujúcich  $I$  riadkov sú dve čísla študentov  $a$  a  $b$ , ktorí sa medzi sebou rozprávajú.

Vypíšte jedno číslo - počet emailov ktoré musí Iveta poslať aby sa informáciu každý dozvedel.

### Obmedzenia

Platí  $1 \leq N \leq 200\,000$ ,  $0 \leq I \leq 3N$ ,  $1 \leq a < b \leq N$ . Zároveň žiadna dvojica  $ab$  nebude na vstupe viackrát.

Nech  $K$  je najväčší počet študentov, ktorí sa nakoniec dozvedia o zrušenej prednáške, ak Iveta pošle jeden email (t.j. najväčšia skupinka študentov, ktorí sa (ne)priamo rozprávajú).

V prvej sade platí  $K \leq 3$ , v druhej  $K \leq 4$  a v tretej  $K \leq N$ .

### Príklad

vstup

5
2
2 3
2 4

výstup

3
---

Prvý a piaty študent sa s nikým nerozprávajú, Iveta im teda musí poslať email. Potom stačí tretí email poslať hociktorému študentovi z 2, 3, 4, tí si to medzi sebou (možno cez 2.) povedia.

## H: Hra s nabíjacím traktorom

50 bodov

Združenie Energeticky Nenáročných Inovatívnych Traktorov (ZENIT) prinieslo na trh elektrický traktor. Priniesli hračkársku verziu aj na Každoročné Stretnutie Poľnohospodárov (KSP). Tí ju potom darovali Korešpondenčnému Semináru z Programovania, ktorí s ňou vymysleli sústredkovú hru. O tejto sústredkovej hre je táto úloha.

Hra spočívala v tom, že na ceste bol položený hračkársky traktor na ručné nabíjanie. Postupne k nemu dobehli členovia tímu, zdvihli ho, v časovom limite pokrútili kľukou na nabíjanie a položili ho naspäť na cestu. Keď sa pri traktore vystriedali všetci z tímu, prišli vedúci, pustili traktor a odmerali, ako ďaleko hore kopcom vedel traktor prejsť.

Ako prezieravý člen sústredka vieš o každom účastníkovi povedať, ako veľmi vie za pridelený čas nabiť traktor. Taktiež vieš predpovedať, či je daný účastník nešikovný. Nešikovní účastníci položia traktor na cestu otočený naopak (a traktor po spustení pôjde do opačného smeru, ako by šiel predtým, než ho daný účastník nabíjal).

Vyber si z účastníkov  $K$ -členný tím tak, aby váš traktor prešiel čo najďalej hore kopcom.

## Vstup a výstup

V prvom riadku vstupu sú dve čísla  $N$  a  $K$  - počet účastníkov a počet ľudí v tíme.

V druhom riadku je  $N$  celých čísel  $u_i$  popisujúcich účastníkov. Účastník  $i$  nabije traktor tak, že prejde o  $|u_i|$  milimetrov viac. Ak je  $u_i$  záporné, účastník je nešikovný a traktor otočí.

Traktor začína otočený smerom hore kopcom.

Formálne, ak vyberieme účastníkov s hodnotami  $a_1, \dots, a_K$ , pričom  $M$  z nich je záporných, traktor prejde  $(|a_1| + \dots + |a_k|) \cdot (-1)^M$  milimetrov hore kopcom.

Vypíšte najväčšie číslo  $v$  také, že sa dá vybrať  $K$  účastníkov, ktorí nabijú a otočia traktor tak, aby prešiel  $v$  milimetrov hore kopcom.

Vo vstupoch platí  $1 \leq K \leq N \leq 300\,000$  a  $1 \leq |u_i| \leq 10^9$ .

Navyše, v prvej sade vstupov platí  $1 \leq N \leq 15$ , v druhej  $K \leq 2$ , a v tretej  $K \leq 5$ . Vo zvyšných dvoch sadách neplatia obmedzenia navyše.

## Príklady

vstup

5 3
5 -1 -6 7 -9

výstup

22
----

Ak vyberieme posledných troch účastníkov, tí nabijú traktor tak, že prejde  $|-6|+|7|+|-9| = 22$  milimetrov. Tretí a posledný účastník traktor otočia, takže ostane smerovať hore kopcom.

vstup

5 3
5 -1 6 7 -9

výstup

18
----

Teraz je najlepšie zobrať do tímu prvého, tretieho a štvrtého účastníka. Najviac by nám traktor nabil piaty, ale ten je nešikovný a traktor by otočil dolu kopcom. Ak by sme zobrali aj druhého účastníka, traktor by skončil otočený hore kopcom, ale druhý účastník je slabý a neoplatí sa nám to.

## I: Iná Oceánska Idylka

55 bodov

Alicka, Ela a Eliška sa vybrali do Turecka na Inú Oceán... ou nie, na Exkluzívnu Grécku Oceánsku Idylku. Počas voľného programu, keď sedeli na pláži s počítačmi a riešili ZENIT, vymysleli namiesto toho novú krypto stabilnú mincu *Zenitoken*. Na rozdiel od ostatných stabilných mincí, ktoré si stabilne držia hodnotu v porovnaní s dolárom, ich minca v porovnaní s dolárom každý deň stabilne stúpa alebo klesá.

Na začiatku má Zenitoken hodnotu 0 dolárov. Následne každý deň jeho hodnota stúpne alebo klesne presne o 1. Zenitoken je až taký revolučný, že nemá problémy ísť aj do záporných hodnôt.

Keďže dovolenka ich stála nemalé peniaze, chceli by si našporiť, dokonca zarobiť niekoľko peňazí. Ako vlastníčky Zenitokenu, každý deň získajú profit rovný jeho hodnote. Získanie peňazí teda nie je vôbec taký problém ako je si ich neminúť. Alicka, Ela a Eliška chcú pred ostatými hosťami nejako vyzerieť a preto potrebujú mať na spoločnom konte za určité obdobia veľmi špecifické hodnoty - vyhnú sa tak nepríjemným škandálom medzi tunajšou smotánkou.

Poradte dievčatám, ako manipulovať trhom, aby to nemuseli kalkulovať ony a mohli si spokojne užívať.

## Vstup a výstup

Na prvom riadku vstupu dostanete počet období  $D$ , ktoré potrebujú kalkulovať. Nasleduje  $D$  riadkov, na  $i$ -tom z nich dva čísla  $1 \leq A_i \leq 10^5$  a  $0 \leq |N_i| \leq 10^{18}$ , počet dní v danom období a suma, ktorú majú za dané obdobie zarobiť.

Na výstup vypíšete  $D$  riadkov. Ak sa za dané obdobie dá zarobiť požadovaná suma, vypíšete na  $i$ -ty riadok medzerami oddelených  $A_i$  čísel udávajúcich hodnoty Zenitokenu v jednotlivé dni. Hodnota v prvý deň musí byť 0, súčet hodnôt musí byť  $N_i$ , a susedné hodnoty sa musia líšiť presne o 1. Ak sa požadovaná hodnota za dané obdobie nedá vykalkulovať, vypíšete slovo "nene".

Sada	1 – 3	4 – 7	8 – 9	10 – 11
$1 \leq D \leq$	$10^2$	$10^3$	$10^4$	$10^5$
$1 \leq A_i \leq$	$10^1$	$10^3$	$10^4$	$10^5$
$1 \leq \text{Sum}A_i \leq$	$10^3$	$10^4$	$10^5$	$10^6$

V sadách 4, 5, 8, 10 je navyše  $|N_i| \leq 10^3$ .

### Príklad

vstup

```
3
5 2
7 3
1 1
```

výstup

```
0 -1 0 1 2
0 1 2 1 0 -1 0
nene
```

V prvom období dievčatá dokážu vykalkulovať  $0 + (-1) + 0 + 1 + 2 = 2$  doláre. V treťom období si nepomôžu, v prvý deň má Zenitoken hodnotu 0

## J: Jeden zásadný rozdiel

60 bodov

Pred pár týždňami sa dokončil v štátnej správe projekt za trinásťcifernú sumu (v mikrocentoch).

Bol to program, ktorý dostal na vstupe matematický výraz v tzv. postfixovej notácii a vyrátal jeho výsledok.

Postfixová notácia funguje tak, že namiesto toho, aby sme operátory písali medzi operandami, píšeme ich za nimi. Napríklad  $(x+y)*z$  zapíšeme ako  $xy+z*$ .

V takejto notácii sa výraz ľahko vyhodnocuje programom. Stačí začať s prázdny zásobníkom a čítať výraz zľava doprava. Keď prečítame operand, pridáme ho na vrch zásobníka. Keď prečítame nejaký operátor 0, zoberieme dve vrchné hodnoty zo zásobníka, aplikujeme na ne operátor a výsledok vrátime do zásobníka. V pseudokóde by to teda vyzeralo takto:

```
a = zasobnik.pop()
b = zasobnik.pop()
zasobnik.push(b 0 a)
```

Na konci procesu bude jediná hodnota v zásobníku výsledok celého výrazu. Môžete si overiť, že tento postup vyráta z výrazu  $xy+z*$  hodnotu  $(x+y)*z$ .

V implementácii programu v štátnej správe je však jeden zásadný rozdiel. Použili namiesto zásobníka (stack) frontu (queue). Tá sa od zásobníka líši tým, že zatiaľ čo pri zásobníku pridávame aj odoberáme prvky z vrchu, pri fronte pridávame prvky na vrch, ale odoberáme ich zospodu. To značne zmení, ako vyhodnotíme daný výraz.

Už je však neskoro, suma je vyplatená a program spustený. Ostáva jediné - napísať ďalší program, ktorý bude prepisovať výrazy z postfixovej notácie na takú, ktorú náš program vyhodnotí rovnako, ako by správny program so zásobníkom vyhodnotil originálnu.

### Vstup a výstup

V prvom riadku vstupu je číslo  $T$  - počet výrazov, ktoré máme prepísať.

V každom z ďalších  $T$  riadkov je jeden výraz. Malé písmenká anglickej abecedy reprezentujú operandy, a veľké písmenká reprezentujú operátory.

Vypíšte pre každý výraz  $A$  taký výraz  $B$ , že  $A$  sa vyhodnotí postupom popísaným v úlohe so zásobníkom rovnako, ako sa vyhodnotí výraz  $B$ , keď pri tom použijeme frontu. Aby bol výstup jednoznačný, budeme predpokladať že žiadny z operátorov nie je asociatívny ani komutatívny ( $aXb \neq bXa$ ,  $(aXb)Xc \neq aX(bXc)$ ).

Platí  $1 \leq T \leq 100$ .

Výrazy budú korektné výrazy v postfixovej notácii, ich dĺžka nepresiahne 50 000 znakov a súčet ich dĺžok nepresiahne 200 000 znakov.

V prvej sade platí, že dĺžka výrazu je najviac 3 znaky.

V druhej a tretej sade platí, že dĺžka výrazu je najviac 100 znakov.

V posledných troch sadách neplatia žiadne obmedzenia navyše.

### Príklad

vstup

```
3
xyPzX
aaAaaAA
abcDEfGhiJK
```

výstup

```
yxzPX
aaaaAAA
cbDaihfeJGK
```

V prvom výraze napushujeme do fronty  $y,x,z$ . Vyberieme z nej  $y$  a  $x$ , a pushneme  $(xPy)$ . Vyberieme  $z$ ,  $(xPy)$  a pushneme  $((xPy)Xz)$ . To je výraz, ktorý by vyhodnotil správny program pre  $xyPzX$ .