

# Zadanie školského kola súťaže ZENIT v programovaní

Kategória A a B, 27.10.2020

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: Ako dobre nám bolo

10 bodov

Pandémia COVID-19 je tu s nami už pár mesiacov a spolu s ňou aj pravidelné každodenné reporty o počte nových potvrdených prípadov na Slovensku.

Vypíšte na výstup, koľko pozitívne testovaných ľudí pribudlo pred mesiacom, 27.9.2020. Správnu odpoveď zrejme nepoznáte z hlavy, preto sa nebojte použiť internet. Dajte si však pozor na to, že prírastok za daný deň sa zverejňuje vždy až v nasledujúci deň.

### Vstup a výstup

Váš program nečíta nič zo vstupu.

Vypíšte jedno číslo, denný prírastok potvrdených prípadov na Slovensku za deň 27.9.2020. Nevypisujte žiadny text navyše, lebo to náš automatický testovač nepochopí.

### Príklad

Túto úlohu zvládnete aj bez príkladu.

## B: Budúci vedúci

20 bodov

Nie každý je stvorený vodca. Nie každý však nie je stvorený vodca.

Do Kancelárie Súbojných Pochodov (KSP) napochodovalo  $n$  účastníkov Pochodu na Stanie sa Vodcom (tento atraktívny názov bol zvolený jedným zamestnancom KSP).

Zamestnanci KSP si o každom pochodujúcom zapísali, ako vedúcovsky prepochovali cez ich dvere.

Pochodujúci bude vedúci, ak je jeho vedúcovkosť ostro vyššia, ako vedúcovkosť všetkých pochodujúcich, čo prišli po ňom.

Zistite, ktorí pochodujúci budú vedúci.

### Vstup a výstup

V prvom riadku vstupu je  $n$  - počet pochodujúcich.

V druhom riadku vstupu je  $n$  nezáporných celých čísel  $v_i$ : vedúcovkosť pochodujúcich, v poradí, v ktorom prepochovali cez dvere KSP. Prvé číslo označuje prvého pochodujúceho, posledné  $n$ -tého.

Na jeden riadok výstupu vypíšte zoznam čísel (oddelených medzerami): poradové čísla pochodujúcich, ktorí budú vedúci. Tieto čísla môžete vypísať v ľubovoľnom poradí.

V prvej sade vstupov (10 bodov) platí  $1 \leq n \leq 1000$ . V druhej sade platí  $1 \leq n \leq 300000$ .

$0 \leq v_i < 10^9$ .

### Príklad

vstup

```
5
5 3 1 4 2
```

výstup

```
1 5 4
```

*Prvý, štvrtý a piaty pochodujúci budú vedúci - po žiadnom z nich už neprišiel niekto s väčšou vedúcovkosťou ako oni. Napríklad druhý pochodujúci nebude vedúci, lebo neskôr prišiel štvrtý a ten mal vedúcovkosť vyššiu.*

## C: Citrónový program

30 bodov

Niektorí programátori radi pijú citrónovú vodu. Samko si povedal, že to tiež skúsi, no zrovna dnes mu odstavili vodu. Napil sa teda čistej citrónovej šťavy, a tak sa mu od tej kyslosti zatočili oči, že vidí všetko naopak.

Samko by od vás potreboval program, ktorý mu pomôže určiť, či slovo, ktoré vidí, sa číta rovnako bez ohľadu na to, či ho číta normálne, alebo naopak (od zadu). Teda či je dané slovo palindróm.

Pozor, Samko si váš program však chce prečítať aj keď ho vidí naopak, preto musí byť aj váš kód sám o sebe palindróm. Teda ak by sme ho prepísali znak po znaku od konca (vrátane medzier a nových riadkov), musí výjsť na znak rovnaký text. Dajte si špeciálne pozor na znak nového riadka na konci súboru. Editor ho tam často pridá bez vášho vedomia, ak tam nie je.

### Vstup a výstup

Na jedinom riadku vstupe je jedno slovo tvorené malými a veľkými písmenami anglickej abecedy a číslami.

Na výstup vypíšte **ano**, ak slovo na vstupe je palindróm, inak vypíšte **nie**. Ak váš program nie je palindróm, testovač vám to oznámi slovami **Program nie je palindrom** a kódom **CERR** teda chyba počas kompilácie.

### Príklad

vstup	výstup
<input type="text" value="kajak"/>	<input type="text" value="ano"/>
vstup	výstup
<input type="text" value="palindrom"/>	<input type="text" value="nie"/>

*Palindrom od zadu je mordnilap.*

## D: Dialpad

30 bodov

Koník si zakúpil mobilný telefón.

Keďže kone sú veľké a nemotorné, krehké ajfóny mu nevyhovovali - vybral si radšej starú dobrú nokiú.

Tá má na sebe dialpad deviatich číslic:



Keďže kone sú od prírody, používajú ako číslice len prirodzené čísla, a na ich nokiách sa číslica 0 ani nevyskytuje.

Teraz si ide Koník uložiť všetkých svojich príbuzných (Strelca, Pešiaka...) do kontakto.

Je tu ale háčik: Koník totiž nemôže len tak postláčať číslice na dialpade. Najprv si môže ľubovoľnú zvoliť, potom však môže stlačiť len takú, na ktorú by vedel skočiť z poslednej stlačenej číslice šachový kôň – musí byť teda položená práve o jeden stĺpec/riadok a o dva riadky/stĺpce od nej.

### Vstup a výstup

V prvom riadku je číslo  $t$  – počet Koníkových príbuzných.

Každý z nasledujúcich  $t$  riadkov obsahuje jedno nezáporné celé číslo  $n$ : číslo, ktoré má kôň vyskakať.

Ak kôň dokáže dané číslo vyskakať, vypíšte pre daného príbuzného do jedného riadku text **hi jo, do toho!**, inak vypíšte **brrrr Konik, to nedavas....**

$$1 \leq t \leq 10$$
$$0 \leq n \leq 10^{1000}$$

## Príklad

vstup

```
3
2943
1
3819
```

výstup

```
hijo, do toho!
hijo, do toho!
brrrr Konik, to nedavas...
```

Pri prvom príbuznom vie medzi všetkými susednými číslicami Koník prestláčať. Pri druhom mu stačí položiť prst na jednotku a je to hotovo. Pre tretieho však Koník nevie stlačiť po jednotke deviatku.

## E: Ešte to prekontroluj

35 bodov

Internet je zradné miesto a preto sa z času na čas stane, že nejaká posielaná informácia sa v ňom stratí a nedorazí do určeného cieľa. Takáto strata informácie je o dosť nepríjemnejšia, ak ide o výsledky testovania na COVID-19. Napríklad, laboratórium pošle internetom na vzdialený server informácie o výsledkoch testovania pacientov, no cestou sa niektoré pakety stratia a zoznam pacientov s výsledkami dorazí na server bez niektorých pacientov alebo s chýbajúcimi výsledkami u niektorých.

Toto nebezpečenstvo je ale väčšinou riešené takzvaným kontrolným súčtom dát - checksum. Pred odoslaním dát nad nimi zavoláme nejakú funkciu. Výsledok tejto funkcie pošleme s dátami. Na druhom konci spojenia nad prijatými dátami zavoláme rovnakú funkciu a pozrieme sa, či sa výsledok zhoduje s tým odoslaným. Ak sa zhodujú, dáta sú v poriadku<sup>1</sup>, inak vieme, že sú poškodené, a môžeme si ich od odosielateľa vyžiadať znova.

Dávid pracujúci v Národnom centre zdravotníckych informácií vymyslel vlastný checksum. Pomôžte mu ho naprogramovať.

## Úloha

Poznáte pole čísel reprezentujúcich záznamy o testovaných. Zoberte ľubovoľné dve čísla z poľa a pridajte naspäť ich súčet. Toto opakujte, kým v poli nie je už iba jedno číslo.

Vypíšte najmenší možný súčet všetkých vami pridaných čísel.

## Vstup a výstup

Na prvom riadku vstupu sa nachádza číslo  $1 \leq t \leq 10$  udávajúce počet testovaní, pre ktoré je potrebné vyrátať checksum. Nasleduje  $t$  testovaní.

Každé testovanie sa skladá z dvoch riadkov. Na prvom riadku je  $1 \leq n \leq 100\,000$  počet testovaných. V druhom riadku je  $n$  čísel  $1 \leq a_i \leq 1\,000\,000$  oddelených medzerou reprezentujúcich záznamy o testovaných.

Súčet  $n$  v jednom testovacom súbore nepresiahne 300 000.

Pre každé testovanie vypíšte jeho checksum vyrátaný podľa Dávidovho algoritmu.

## Príklad

vstup

```
2
4
1 8 20 21
2
9 7
```

výstup

```
88
16
```

Na vstupe sú dve testovania.

V prvom sme zobrali 1 a 8 a ich súčet sme vložili naspäť. V poli je teraz 9 20 21. Zoberieme 9 a 20 a ich súčet vložíme naspäť. Máme 29 a 21. Nakoniec zoberieme 29 a 21, ich súčet vložíme naspäť. V poli ostane už len 50, vtedy končíme. Súčet nami vložených čísel je  $9 + 29 + 50 = 88$ .

V druhom testovaní sčítame  $9 + 7 = 16$  a vložíme to naspäť. Ostane nám už iba 16, končíme.

## F: Feta číz

30 bodov

Konzumenti Syrových Produktov (KSP) zorganizovali promočnú marketingovú aktivitu na spopularizovanie Feta syra.

<sup>1</sup>Samozrejme, môže sa stať, že poškodené dáta majú rovnaký checksum ako tie pôvodné. To ale nevádi. Pravdepodobnosť takejto situácie ja rádovo nižšia než toho, že checksum odhalí chybu.

Keďže svet je na tom tak, ako na tom je, vymysleli spôsob ako nikto nemusí s nikým prísť do kontaktu. Každý z  $n$  záujemcov pošle KSP poštou lístok so svojím menom. KSP potom tieto lístky podpíše, náhodne preusporiadajú, a odošlú späť.

Záujemca vyhrá jedno balenie Feta syra ak sa mu pošťastí a poštou mu príde späť práve jeho lístok.

KSP prijalo  $n$  lístkov. Matej Niva, vedúci KSP, ich zamieša, zatiaľ čo Jozef Čedar má za úlohu nakúpiť balenia Feta syra. Hm, ale koľko ich má kúpiť? Ak ich kúpi primálo, bude musieť ísť zas. Ak ich kúpi priveľa, bude musieť skonzumovať všetko čo ostalo, a neostane mu miesto na lahodný Čedar.

Pomôžte Jozefovi s jeho dilemou, a zrátajte mu, koľko výhercov môžu očakávať.

### Vstup a výstup

V jedinom riadku vstupu je celé číslo  $n$  - počet záujemcov čo poslali KSP lístok.

Vypíšte jedno reálne číslo  $o$  - očakávaný počet výhercov balenia Feta syra.  $o = \sum_{x=0}^n x \cdot p_x$ , kde  $x$  je počet výhercov feta syra (záujemcov, ktorí po náhodnom preusporiadaní dostanú naspäť svoj lístok), a  $p_x$  je pravdepodobnosť, že po zamiešaní bude výhercov práve  $x$ .

Vypíšte presne šesť čísel za desatinnou čiarkou.

Vo všetkých vstupoch platí  $0 \leq n \leq 1000$ . Sú štyri sady vstupov. V jednej zaručene  $n \leq 10$ , v inej zaručene  $n \leq 20$ .

### Príklad

vstup

2

výstup

1.000000

*KSP dostane dva lístky, náhodne ich vloží späť do obálky, a zašle naspäť. Je teda polovičná pravdepodobnosť, že nikto nevyhrá, a polovičná pravdepodobnosť, že obaja záujemcovia vyhrajú. Očakávaný počet je teda  $o = 0 \cdot 0.5 + 2 \cdot 0.5 = 1$*

## G: GreedIsGood

40 bodov

V dávných, dobrých časoch, bola Warcraft 3 populárna hra. Gurmánom počítačových hier medzi vami možno niečo hovorí hláška **GreedIsGood**. Tento cheatcode hráčovi pridá ľubovoľné množstvo zlata a dreva, čo sú zdroje potrebné na všetko, od najatia jednotiek po stavanie budov.

Nemenovaný hráč známy vo svete Warcraftu len ako 'fejzo' by však takto nikdy nepodvádzal.

V tomto momente má presne  $e$  zlata a  $p$  dreva<sup>2</sup>. Má na výber spomedzi  $n$  rôznych, unikátnych hrdinov, ktorých môže pridať do svojej armády - chce si najat' presne dvoch. Každý má nejakú bojovú silu, a nejakú cenu (buď v dreve, alebo v zlate).

Pomôžete mu zistiť, aký najväčší súčet bojovej sily vie pridať svojej armáde!

### Vstup a výstup

V prvom riadku vstupu sú tri celé čísla  $n$ ,  $e$  a  $p$ : počet hrdinov a fezjove zásoby zlata a dreva.

Nasledujúcich  $n$  riadkov popisuje jedného hrdinu tokenmi  $s_i c_i z_i$ .  $s_i$  je sila daného hrdinu,  $c_i$  je jeho cena a  $z_i$  je zdroj - E alebo P - v ktorom je hrdinova cena.

Platí  $2 \leq n \leq 200\,000$  a  $1 \leq s_i, c_i, e, p \leq 10^9$ . V polovici vstupov  $n \leq 1000$ .

Každého uvedeného hrdinu si môže fejzo najat' iba raz. Ak si fejzo nemôže najat' žiadnych dvoch hrdinov, vypíšte v jednom riadku **Reality is often disappointing**.

Inak vypíšte najväčší súčet síl dvoch hrdinov, ktorých si fejzo vie oboch naraz zakúpiť a nepresiahnuť pritom svoj rozpočet zlata a dreva.

### Príklad

vstup

4 10 20  
5 3 E  
6 8 P  
8 10 P  
9 13 P

výstup

14

*Najlepšie, čo fejzo vie spraviť, je zakúpiť si druhého a tretieho hrdinu. Bude ho to stáť  $8+10 = 18$  dreva, čo je pod jeho rozpočet 20, a súčet ich síl je  $8+6=14$*

<sup>2</sup>'e' stojí za 'ezlato', 'p' za 'pdrevo'

vstup

```
2 10 10
6 6 E
6 6 E
```

výstup

```
Reality is often disappointing
```

*Aby si zakúpil oboch hrdinov, potreboval by fezjo 12 zlata. Má ho však len 10.*

vstup

```
2 10 10
6 6 E
6 6 P
```

výstup

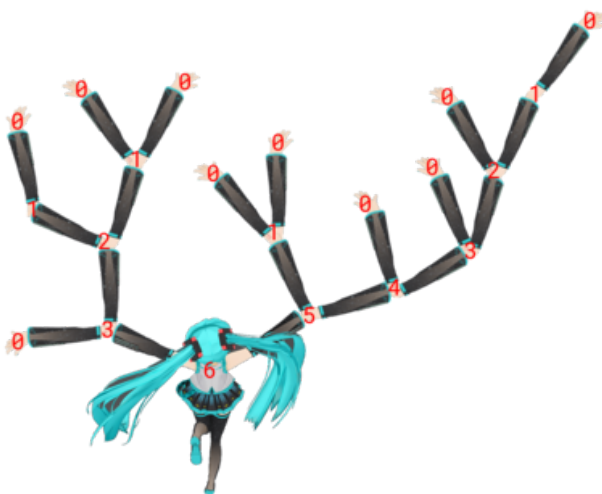
```
12
```

## H: Hašterivé sny

50 bodov

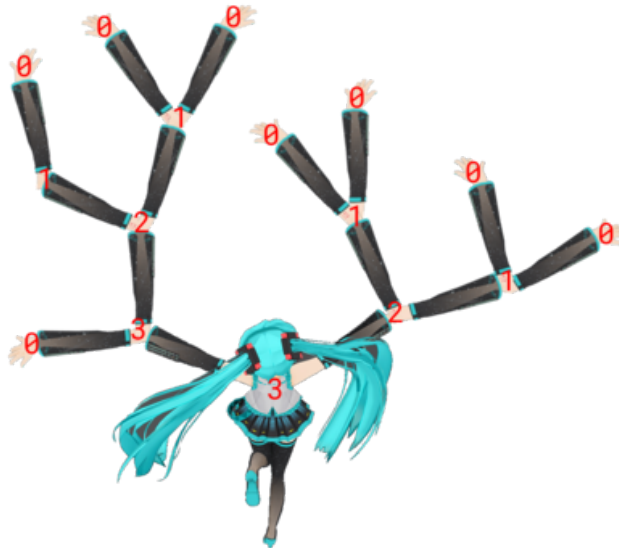
Minulú noc sa Jozefovi sníval znova ten istý sen. Keď sa na seba pozrel, videl, ako mu z tela vychádzajú dve ruky. Síce na jedného človeka nadpriemerný počet<sup>3</sup>, ale to nebolo to surreálne pozorovanie. Ako sa tak pozeral od ramien smerom k zápästiam, na ich miestach nenašiel dlane a prsty, ale krk a znova sa vynárajúce dve ramená. Toto sa pár krát opakovalo, niekedy z krku vychádzalo iba jedno rameno, inokedy konečne prsty.

Bežný človek má dosah na vzdialenosť dĺžky jednej ruky. Jozef však v týchto snoch vďaka svojmu vzrastu dokázal dosiahnuť na mnohonásobok toho na čo bol zvyknutý. Za ten čas čo strávil v každonočných snoch a videl ako sa jeho ruky nekontrolovateľne hmýria, trasú a siahajú na všetko naokolo, zvládol odpozorovať dva kľúčové fakty: vie aký veľký má dosah a vie, že keďže je pravák, tak pre každý krk má pravé rameno aspoň taký veľký dosah ako to ľavé.



*jeden príklad valídneho sna s dosahom 6 (v každom krku máme napísaný maximálny dosah)*

<sup>3</sup>[veľmi málo ľudí má viac ako 2 ruky, pomerne veľké množstvo si však nedáva pozor pri ťažkých strojoch]



tu zase vidíme príklad postihnúťého sna - v hneď prvom krku má pravé rameno menší dosah ako ľavé

### Úloha

Jozef má vo svojom sne hlavu pripevnenú ku krku. Z každého krku môžu vychádzať najviac dve ramená (ak by ich vychádzalo viac tak by to bolo nerealistické), všetky ukončené krkami. Keďže je pravák tak pre každý krk platí, že pravé rameno má aspoň taký veľký dosah ako ľavé.

Jozef si o svojich snoch píše do denníčka, avšak človek čo sa zobudí si zo svojho sna pamätá iba veľmi málo a tak jeho denníček obsahuje iba zoznam čísel udávajúcich jeho najväčší dosah z danej noci.

Všetkých by nás zaujímalo, koľkými rôznymi spôsobmi mohli z Jozefa vychádzať jeho končatiny, tak to pre každý zápis v denníčku zistite.

### Formát vstupu

Prvý riadok obsahuje číslo  $N$  - počet nocí ( $1 \leq N \leq 10^5$ ). Nasleduje  $N$  riadkov, na každom riadku jedno celé číslo  $d_i$  - maximálny dosah počas  $i$ -tej noci ( $0 \leq d_i \leq 2 * 10^6$ ).

### Formát výstupu

Na výstup vypíšete  $N$  riadkov, na každom riadku jedno číslo - odpoveď na otázku pre danú noc modulo  $10^9 + 7$ .

### Príklad

vstup

```
2
0
1
```

výstup

```
1
2
```

## I: Iné číslo

80 bodov

Matematici majú niekedy zvláštne nápady, a v horšom prípade s nimi ešte obťažujú informatikov. A presne to sa stalo aj v tejto úlohe.

Matúš je matematik a má nezáporné celé číslo  $n$ . Matúš by však chcel mať číslo, ktoré by bolo deliteľné kladným celým číslom  $m$ . Napadlo ho (lebo je matematik) že zmení niektoré cifry v tomto čísle. (Pozor, cifry bude len meniť, nesmie žiadne pridávať ani odoberať.)

Aby si Matúš neminul veľa papiera na výpočty, bolo by vhodné spraviť tých zmien čo najmenej.

### Úloha

Pre dané  $n$  a  $m$  zistite, či vie Matúš povoleným spôsobom upraviť  $n$  na násobok  $m$ , a ak áno, tak koľko najmenej cifier potrebuje zmeniť.

### Formát vstupu

V jedinom riadku vstupu sú čísla  $n$  a  $m$ .  
Platí  $0 \leq n \leq 10^{11}$  a  $1 \leq m \leq 10^{11}$ .

### Formát výstupu

Vypíšte jediný riadok a v ňom jedno celé číslo. Ak Matúš nevie dosiahnuť svoj cieľ, vypíšte číslo -1 (mínus jedna). V opačnom prípade vypíšte ľubovoľné číslo, ktoré má vyrobiť: teda také číslo, ktoré je násobkom  $m$ , má rovnaký počet cifier ako číslo  $n$  a líši sa od čísla  $n$  na čo najmenej pozíciách. Vypísané číslo nesmie začínať zbytočnými nulami.

### Príklady

vstup

123 10

*Stačí jedna zmena: poslednú cifru z 3 na 0.*

vstup

123 141

*Opäť stačí jediná zmena, tentokrát na inej pozícii.*

vstup

9 123

*Samotná nula je špeciálnym prípadom, keďže nezačína zbytočnými nulami.*

vstup

12 123

*Žiadne dvojciferné číslo nie je násobkom čísla 123.*

výstup

120

výstup

423

výstup

0

výstup

-1