Zadania krajského kola súťaže ZENIT v programovaní

6.12.2012

V prípade nejasností konzultujte záložku **Pomoc** na stránke `riesky.sk/zenit` 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 servri (nie na vašom lokálnom počítači).

A: Výzva do boja!

10 bodov — Časový limit: 100 ms

Tak ako vy v programovaní, tak aj Lukáš sa v tancovaní natoľko zlepšil, že ho poslali na krajské kolo veľkej súťaže! V tejto úlohe nečítajte žiaden vstup a vypíšte (bez úvodzoviek, nezabudnúť na výkričník):

`„Lukas si dnes vytancuje vitazstvo v krajskom kole!"`

B: Telefónny zoznam

20 bodov — Časový limit: 500 ms

Po príchode na krajské kolo sa Lukáš musel registrovať pri veľkom stole plnom papierov a neporiadku. Všetci pobehovali okolo a riešili problémy. Napríklad jedna organizátorka chcela vytlačiť zoznam telefónnych čísel všetkých zúčastnených. Pritom ale nariekala, že sa míňa v tlačiarni toner a to ešte nie sú vytlačené ani potvrdenia o bezinfekčnosti.

Keď to Lukáš videl, tak sa v ňom prebudil optimalizátor – niekedy sa pri tlačení zoznamu predsa dá niečo ušetriť! Napríklad namiesto

```
4513
4523
4524
4524
4777
5777
```

by stačilo vytlačiť iba

```
4513
  23
   4

 777
5777
```

Postupujeme zľava doprava a ak sa daná cifra zhoduje s cifrou nad ňou, potom ju môžeme vynechať. Ak sa cifry nezhodujú, potom s optimalizáciou v danom riadku končíme. Všimnite si, že vo výslednom zozname môže chýbať aj niekoľko cifier v stĺpci za sebou. Vtedy platí, že sú všetky zhodné s cifrou, ktorá je vytlačená nad nimi (taká vždy existuje, keďže prvé číslo tlačíme vždy celé). V uvedenom prípade sme takto ušetrili 10 cifier.

Aby Lukáš dokázal prítomných organizátorov presvedčiť, rád by vedel pre daný zoznam čísel, koľko symbolov sa ušetrí. Napíšte mu program, ktorý mu umožní presadiť svoje riešenie!

Na vstupe je na prvom riadku počet telefónnych čísel N ($1 \leq N \leq 10,000$) a na každom z ďalších N riadkov reťazec cifier 0 až 9. Môžete predpokladať, že všetky čísla sú rovnako dlhé. Ich dĺžka je aspoň 1 a nepresahuje 50 (môžu sa začínať aj nulou). Iné znaky ako cifry 0 až 9 sa na vstupe nevyskytnú. Na výstup vypíšte jedno číslo: počet ušetrených cifier pri použití Lukášovho algoritmu. Poradie čísel v zozname nemôžete meniť. Hint: odporúčame reťazce na vstupe čítať ako typ `string` a nie ako čísla.

Príklady:

vstup

```
6
4513
4523
4524
4524
4777
5777
```

výstup

```
10
```

vstup

```
10
134567432541243980
024343214565619981
024343214565771334
024343214565771335
024343214565771335
124343214565771335
124343224565771335
124343224565778456
124343225445643134
123434324668454313
```

výstup

```
78
```

C: Transparenty

25 bodov — Časový limit: 500 ms

Lukáš sa úspešne zaregistroval a teraz sa prechádza medzi pripravujúcimi sa účastníkmi a fanúšikmi a nasáva atmosféru. Keď prechádzal okolo početnej skupiny fanúšikov, úplnou náhodou si všimol, že sa práve chystá vytlačiť z tlačiarne organizátorov veľký transparent s nápisom **Maroško je najlepší tanečník Samby, Maroško je najlepší na svete**. Lukáš si tiež všimol, že na chvíločku nedáva nikto pozor a že má trošku času. Za ten čas môže napríklad nastaviť prihlásenej osobe zaujímavý status na Facebooku. Alebo dokáže spraviť jeden search and replace v texte transparentu! No nebol by skvelý transparent s nápisom **Lukáš je najlepší tanečník Samby, Lukáš je najlepší na svete**? Alebo **Maroško je sotva priemerný tanečník Samby, Maroško je sotva priemerný na svete**? Keďže Lukáš má na úpravu transparentu viacero možností a tiež je možné, že sa mu podarí v budúcnosti upraviť viacero transparentov, napíšte mu program, ktorý bude túto funkciu realizovať. Tak si bude vedieť dopredu pozrieť potenciálny výsledok svojho záškodníctva.

Vstupný súbor obsahuje tri riadky. Na prvom je čo chceme nahradiť. Na druhom je čím chceme nahradzovať. Na treťom riadku je samotný text transparentu. Všetky tri riadky majú najviac 500 znakov a obsahujú len veľké písmená anglickej abecedy a medzery. Každý riadok obsahuje aspoň jedno písmeno. Môžete predpokladať, že každé dve slová v každom riadku sú oddelené práve jednou medzerou a že vo vstupnom súbore nie sú žiadne medzery navyše.

Na výstup vypíšte na jedinom riadku textu výsledok nahradenia. Nahrádzanie sa realizuje nasledovne. Najprv nájdeme všetky výskyty vzorky: postupujeme zľava doprava a počítame len vzorky, ktoré sa neprekrývajú s predchádzajúcim výskytom. Následne všetky naraz nahradíme výsledkom.

Napríklad, nech `AAABAAAA` je text a nech chceme nahradiť `AA` reťazcom `BAA`. Výskyty vzorky sú $\overline{\mathtt{AA}}\mathtt{AB}\overline{\mathtt{AA}}\,\overline{\mathtt{AA}}$. Všimnite si, že reťazec `AA` pozostávajúci z druhého a tretieho písmena nie je výskyt vzorky, lebo prvé `A` už je použité. Po nahradení dostávame $\overline{\mathtt{BA}}\mathtt{AAB}\overline{\mathtt{BA}}\,\overline{\mathtt{BA}}$.

Príklady:

vstup

```
MAROSKO
LUKAS
MAROSKO JE NAJLEPSI TANECNIK SAMBY
```

výstup

```
LUKAS JE NAJLEPSI TANECNIK SAMBY
```

vstup

```
HIP HIP
HURA
HIP HOP HURA HIP HIP HURA HIP HURA HIP
```

výstup

```
HIP HOP HURA HURA HURA HIP HURA HIP
```

vstup

```
PALMA
ZENIT
ZENIT JE NAJLEPSIA SUTAZ
```

výstup

```
ZENIT JE NAJLEPSIA SUTAZ
```

D: Šifra

25 bodov — Časový limit: 500 ms

Lukáš sa počas tréningového roztancovávania ocitol v podobnej situácii ako počas školského kola, avšak v inej roli. Vidí, ako si chlapec v jednom rohu posiela šifrované správy s krásnou tanečníčkou v opačnom rohu. Lukáš je teraz v roli jedného z ľudí, ktorí podávajú lístoček a ako správny zvedavý Man In The Middle by si rád prečítal posielané správy. Mohol by sa predsa naučiť niečo o nadväzovaní nových kontaktov!

Na rozdiel od Lukáša v školskom kole zvolil náš šifrant o niečo sofistikovanejší spôsob – substitučnú šifru. Pri substitučnej šifre nahrádzame písmená inými písmenami a všetky ostatné symboly nechávame na mieste. Uvažujme 26 veľkých písmen anglickej abecedy. Ak napríklad A nahrádzame znakom C, B nahrádzame znakom A a C nahrádzame znakom B, potom šifra textu „AB, BC.“ je „CA, AB.“. Aby sme vedeli text jednoznačne dešifrovať, je potrebné, aby bolo každé písmeno obrazom práve jedného z písmen. Skutočnosť, ktoré písmeno sa zobrazí na ktoré je kľúčom k šifrovaniu a dešifrovaniu správ. Pre úspešnú komunikáciu musia túto informáciu mať obe komunikujúce strany. Zdá sa, že na začiatku komunikácie náš tanečník tento kľúč poslal tanečníčke. Lukáš bohužiaľ túto správu nezachytil, ale podarilo sa mu zachytiť dve správy, o ktorých vie ich šifrovanú aj nešifrovanú podobu, pretože sa pozeral cez rameno tanečníka. Tieto dve správy máte ako príklady vstupu a výstupu.

Na vstupe vášho programu bude tretia správa tanečníka, ktorá je šifrovaná rovnakým kľúčom ako uvedené dve v príklade. Vstup pozostáva len zo znakov veľkých anglických písmen, medzier, bodiek a symbolov nových riadkov (jeden znak `\n`, ASCII kód 10). Výstup bude rozšifrovaný text. Dbajte na presné zachovanie formátu správy, všetky nepísmenové znaky musia ostať na mieste! Nepredpokladajte o vstupe nič rozumné: ani že je text zmysluplný, ani že v ňom nesmú byť dve bodky za sebou, ani že neobsahuje prázdny riadok, ani že na konci riadku nie je medzera navyše (ktorú musíte uviesť vo výstupe)! Dĺžka textu je najviac 2000 znakov, vstup pozostáva z najviac 100 riadkov.

Príklady:

vstup

```
S XJKYPGXPVX LJG LY SXCBMXMY PXTYF. PUJ
LJG LX DBCYGSMY
DB EGNPVX X BWDYCBCXMY
NBENR
OVBEMXO.
```

výstup

```
Z AMFITEATRA SME SI ZAVOLALI TAXIK. TYM
SME SA DOVIEZLI
DO CENTRA A OBDIVOVALI
NOCNU
WROCLAW.
```

vstup

```
C FXSDBJ HVBQVXJG AG EZUWX.
```

výstup

```
V KAZDOM PROGRAME JE CHYBA.
```

E: Rozmiestnenie do sál

30 bodov — Časový limit: 500 ms

Na našom podujatí sa uskutoční niekoľko súťaží v rôznych tanečných disciplínach. Mnohí súťažiaci sa zúčastňujú na viac ako jednej disciplíne a preto organizátori navrhli časový harmonogram jednotlivých disciplín tak, aby vyhovoval každému. Teraz by radi rozdelili jednotlivé udalosti do jednotlivých tanečných sál.

Na vstupe je číslo P ($1 \leq P \leq 100$) – počet jednotlivých súťažných udalostí. Na každom z nasledujúcich P riadkov je popis tanečnej udalosti: interval, kedy by sa mala uskutočniť a jej názov. Tento riadok má tvar `HH:MM - HH:MM nazov`. `HH` označuje číslo od 8 do 19 vrátane (v prípade 8 a 9 je to len jeden znak, v ostatných prípadoch dva). `MM` sú vždy dva znaky a označujú minútu v rozmedzí `00` až `59`. Názov je reťazec pozostávajúci len z malých a veľkých písmen anglickej abecedy. Žiadny názov nie je dlhší ako 30 znakov a žiadne dve udalosti nemajú rovnaký názov. Môžete predpokladať, že štartovný čas je vždy menší ako koncový čas.

Vašou úlohou je nájsť rozdelenie do miestností spĺňajúce nasledovné obmedzenia:

- V každom momente bude v každej miestnosti prebiehať najviac jedna udalosť. Ak jedna udalosť začína v tej istej minúte ako iná končí, nevadí to.
- Neexistuje rozdelenie do miestností spĺňajúce predchádzajúci bod, ktoré by používalo menej miestností.
- Spomedzi všetkých rozmiestnení, ktoré spĺňajú predchádzajúce dva body môžete nájst a vypísať ľubovoľné.

Na prvom riadku výstupu vypíšte počet použitých miestností M. Na zvyšných M riadkov výstupu vypíšte zoznam udalostí, ktoré sa uskutočnia v jednotlivých miestnostiach. Každý riadok označuje jednu z miestností. Riadky môžete vypísať v ľubovoľnom poradí a udalosti v rámci riadku tiež. Predpokladajte, že miestností máte v budove k dispozícii ľubovoľne veľa.

Príklady:

vstup

```
4
8:00 - 10:00 samba
9:00 - 12:00 rumba
11:00 - 15:00 odzemok
12:00 - 13:00 disco
```

výstup

```
2
odzemok samba
rumba disco
```

vstup

```
4
17:14 - 18:43 tango
8:15 - 8:22 wienerwalc
10:17 - 15:56 hophop
8:22 - 9:46 chacha
```

výstup

```
1
chacha wienerwalc tango hophop
```

vstup

```
8
14:00 - 17:00 G
9:00 - 12:00 B
8:00 - 11:00 A
11:00 - 13:00 D
16:00 - 18:00 H
12:00 - 19:00 E
13:00 - 15:00 F
10:00 - 18:00 C
```

výstup

```
4
A D F H
B E
C
G
```

F: Zátvorky

25 bodov — Časový limit: 1000 ms

Tanečníci sa zoradili do dlhého radu, aby si nacvičili úvodné zábery do oficiálneho videa podujatia. Režisér ale nie je spokojný s rozostavením dám a pánov. Aby bol spokojný, na niektoré miesta do davu musí pridať ďalších tanečníkov.

Dámy budeme označovať znakom `(` a pánov znakom `)`. Režisér je spokojný vtedy a len vtedy, ak rad zodpovedá dobre uzátvorkovanému výrazu. Dobre uzátvorkovaný výraz môžeme definovať rekurzívne:

- Prázdny reťazec je dobre uzátvorkovaný výraz.
- Ak `R` je dobre uzátvorkovaný výraz, potom aj `(R)` je.
- Ak `R` a `S` sú dobre uzátvorkované výrazy, potom aj `RS` je.
- Ak sa reťazec nedá skonštruovať použitím konečného počtu týchto troch pravidiel, potom nie je dobre uzátvorkovaný výraz.

Na vstupe máte aktuálny rad tanečníkov zapísaných ako reťazec znakov `)` a `(`. Môžete predpokladať, že tento reťazec neobsahuje iné znaky ako tieto dva, že je neprázdny a že nie je dlhší ako 500,000 znakov. Na výstup vypíšte koľko najmenej tanečníkov treba pridať, aby výsledný rad zodpovedal dobre uzátvorkovanému výrazu. Môžete predpokladať, že toto číslo je medzi 0 a 2^{30} vrátane.

Príklady:

vstup

```
()())((
```

výstup

```
3
```

vstup

```
))()
```

výstup

```
2
```

vstup

```
()()
```

výstup

```
0
```

G: Trojuholník

30 bodov — Časový limit: 500 ms

Nácvik videa z predchádzajúcej úlohy pokračoval záberom zhora. Režisér si rozostavil tanečníkov v hale, ktorá má obdĺžnikovú podstavu a jej šírka a dĺžka v metroch je kladné celé číslo. Halu môžeme rozdeliť na pomyselné štvorčeky s veľkosťou 1x1 meter. Každý takýto štvorec je prázdny alebo v ňom stojí jeden tanečník. Keď sa na situáciu pozrieme zhora, môžeme si ju nakresliť ako mriežku, kde `.` označuje prázdne miesto a `#` obsadené.

Hlavným sponzorom podujatia je firma `Trojuholníky s.r.o`, ktorá sa venuje výrobe predmetov trojuholníkového tvaru. Rozostavenie je preto uspokojivé, ak obsadené políčka formujú trojuholník. Trojuholník s veľkosťou d je tvorený vodorovnou podstavou dĺžky $2d + 1$ a dvoma šikmými stranami dĺžky $d + 1$. Príklady trojuholníkov s veľkosťou 1,2 a 3:

```
.......................
...#.....#.......#.....
..###...#.#.....#.#....
.......#####...#...#...
..............#######..
.......................
```

Trojuholník pozostáva len z obvodu (nemusí byť vyplnený). Napíšte program, ktorý prečíta popis aktuálneho stavu miestnosti a zistí, či obsadené políčka formujú nejaký trojuholník. Trojuholník musí byť orientovaný ako v príklade. Otočenia dole hlavou, o 90 alebo 270 stupňov vyhovujúce nie sú. V sále môžu byť aj iní tanečníci. Stačí, že sa trojuholník vyskytne niekde medzi nimi (viď príklady vstupu).

V prvom riadku vstupu sú dve celé čísla R a C ($1 \leq R, C \leq 50$) – rozmery sály. Na ďalších R riadkoch je samotný popis. Na každom z týchto riadkov je reťazec dlhý C, ktorý pozostáva len zo znakov `.` a `#`. Na jediný riadok výstupu vypíšte `ANO`, ak znaky `#` formujú trojuholník s veľkosťou d pre nejaké $d \geq 1$. V opačnom prípade vypíšte `NIE`.

Príklady:

vstup

```
4 4
.#..
##..
###.
...#
```

výstup

```
ANO
```

vstup

```
4 4
..#.
.#.#
#.##
...#
```

výstup

```
NIE
```

vstup

```
6 7
..##..#
.#.#.##
..###.#
.#..##.
#######
#.##..#
```

výstup

```
ANO
```

H: Generátor bludísk

40 bodov — Časový limit: 500 ms

Je obedová prestávka a Lukáš si potrebuje oddýchnuť a vyventilovať hlavu. Preto si kúpil svoj obľúbený denník. Možno neviete, ale keď si Lukáš kúpi svoj obľúbený denník, nehľadá v ňom ani správy z domova, ani zo športu a ani z algoritmických súťaží. Nerieši ani trápne sudoku a nečíta si ani komix s Garfieldom. Áno,

uhádli ste. Lukáš si ako prvé vždy pozrie bludisko a perom vyznačí cestu. V tejto úlohe by ste mali pomôcť redakcii týchto novín a naprogramovať jej generátor labyrintov, aby sme zabezpečili stabilný prínos inštancií tejto zaujímavej logickej disciplíny. Bludisko vyzerá napríklad takto:

```
#########
#.#S....#
#.###.#.#
#.....#.#
###.###.#
#C..#...#
#########
```

Bodka označuje chodbu, mriežka stenu. Krok sa dá spraviť medzi políčkami len vodorovne a zvislo (tzn. každé políčko má najviac štyroch susedov). Dĺžka cesty zo štartu (`S`) do cieľa (`C`) je v uvedenom prípade 10 krokov.

Na vstupe sú tri čísla: R, C a K ($4 \leq R, C \leq 100$, $1 \leq K \leq R \cdot C$). Výstupom vášho programu je bludisko, ktoré má rozmery R riadkov, C stĺpcov a najkratšia cesta zo štartu do cieľa má presne K krokov. Ak také bludisko neexistuje, máte vypísať `NEDA SA`. Presnejšie, pre bludisko na výstupe musí platiť:

- Všetky políčka na obvode bludiska musia byť steny.
- Nesmie existovať cesta zo štartu do cieľa, ktorá by bola kratšia ako K krokov.
- Musí existovať cesta zo štartu do cieľa, ktorá má K krokov.
- Nesmie obsahovať žiadny znak okrem spomenutých štyroch a znaky `S` a `C` musí obsahovať práve raz.
- Jeho plán musí pozostávať z práve R riadkov a C stĺpcov.

Vypíšte ľubovoľné bludisko spĺňajúce tieto požiadavky.

Príklady:

vstup

```
5 5 5
```

výstup

```
#####
#S..#
###.#
##C.#
#####
```

vstup

```
10 8 22
```

výstup

```
########
#......#
##.###.#
#C.#...#
#####.##
#......#
#.####.#
#.#S#..#
#...#.##
########
```

vstup

```
10 8 44
```

výstup

```
NEDA SA
```

I: Párty

40 bodov — Časový limit: 1500 ms

Po skončení súťažnej časti programu sa tanečníci tešia na veľkú párty s jahodami a šampanským. Mnohí účastníci sú ale introvertní a hoci chuť majú, na párty sa hanbia ísť, ak sa na nej nezúčastní určitý počet kamarátov.

Na prvom riadku vstupu sú tri celé čísla: počet súťažiacich N ($1 \leq N \leq 200,000$), počet dvojíc kto sa s kým pozná M ($0 \leq M \leq 600,000$) a číslo K ($0 \leq K \leq N-1$). Súťažiacich pre jednoduchosť označíme číslami 1 až N. Zámerne neprezradíme, ktoré číslo má Lukáš, aby ste nemohli získať žiadnu informáciu o skutočnom počte jeho kamarátov. Nasleduje M riadkov, na každom sú dve čísla popisujúce dvojicu účastníkov, ktorí sa kamarátia. Môžete predpokladať, že tieto dve čísla sú z rozsahu 1 až N, sú rôzne a tiež že na každom z týchto riadkov je rôzna dvojica čísel.

Každý z účastníkov je schopný ísť na párty len vtedy, ak sa na nej zúčastní aspoň K jeho kamarátov. Podmnožina ľudí je vyhovujúca vtedy a len vtedy, ak pre každého jej člena platí, že je v podmnožine aj K alebo viac jeho kamarátov. Niektoré podmnožiny účastníkov túto podmienku spĺňajú a niektoré nespĺňajú. Napríklad prázdna množina vyhovuje vždy. Nájdite najväčšiu vyhovujúcu množinu. Ak je takých viacero, vypíšte ľubovoľnú z nich. Na prvom riadku výstupu je počet účastníkov párty. Ak je tento počet kladný, tak výstup obsahuje aj druhý riadok. Na ňom je zoznam účastníkov. Títo majú byť vypísaní v rastúcom poradí a jednotlivé čísla majú byť oddelené práve jednou medzerou.

Táto úloha sa dá riešiť veľa rôznymi spôsobmi a získaný počet bodov bude spravodlivo odvodený od časovej zložitosti vášho riešenia.

Príklady:

vstup

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

výstup

```
3
1 2 3
```

vstup

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

výstup

```
0
```

vstup

```
9 8 2
1 7
1 5
2 8
2 5
4 9
6 4
2 7
2 1
```

výstup

```
4
1 2 5 7
```

J: Deľba cukríkov

35 bodov — Časový limit: 500 ms

Keďže Lukášovi sa po celom tom dni plnom súťaží už nechcelo na tanec ani len pomyslieť, na párty sa väčšinou venoval konverzácii s inými účastníkmi a konzumácii zaujímavých dobrôt. Najviac ho oslovili mištičky so zvláštnymi cukríkmi. Mištičiek je veľa a v každej je nejaký kladný počet chutných cukríkov. Lukáš by najradšej

zjedol všetky. Bohužiaľ, podobnú ambíciu má aj chamtivý Maroško. Napokon sa Lukášovi podarilo dohodnúť, že sa rozdelia férovo – tak, aby mali obaja rovnako cukríkov. Keďže nechcú presýpať cukríky medzi mištičkami, môžu si deliť len samotné mištičky.

Maroško trénoval tanec tak veľa, že zabudol všetko ostatné. Okrem iného aj počítať. V minulosti si ale ako slávny programátor naprogramoval kalkulačku na mobil a tú teraz používa. Lukáš si všimol, že táto kalkulačka má dosť závažnú chybu: pri sčítaní zabúda na prenos do vyššieho rádu. Napríklad sčítanie čísel 13 a 27 vyzerá nasledovne. Najprv si ich kalkulačka zapíše binárne ako `1101` a `11011`. Potom sčíta nasledovne:

```
   01101
   11011
-------
   10110
```

Pri sčítaní sa postupuje klasicky sprava doľava. Prvý stĺpec bol sčítaný správne ako 1+1=2. To znamená, že daná cifra bude 0. Pri správnom postupe by sme si mali zapamätať prenos +1 do vedľajšieho stĺpca, čo ale Maroškova kalkulačka nerobí. Takto Maroško dostane výsledok `10110`, čo je v desiatkovej sústave 22. Správny výsledok má byť samozrejme `101000` (40). Lukáš vie, že Maroško sa spolieha na svoju kalkulačku a že mu vôbec nepripadá zvláštne, ak je výsledok sčítania kladných čísel menší ako jeden zo sčítancov.

Ak sa Lukášovi podarí rozdeliť mištičky na dve hromádky tak, že výsledok sčítania počtu cukríkov na Maroškovej kalkulačke bude rovnaký, potom bude Maroško s daným rozdelením súhlasiť. Počet mištičiek nemusí byť rovnaký a Maroško musí dostať aspoň jednu. Pomôžte Lukášovi a zistite, či také rozdelenie existuje a ak áno, koľko najmenej cukríkov môže byť na Maroškovej hromádke. Maroško počíta súčet postupne: pamätá si medzivýsledok a k nemu po jednej pripočítava mištičky (zľava doprava v poradí ako na vstupe). Napríklad, ak je mištičiek šesť a Lukáš si chce nechať druhú a piatu, potom Maroško najprv spočíta súčet prvej a tretej mištičky. K tomuto súčtu Maroško pripočíta počet cukríkov v štvrtej mištičke a napokon k súčtu týchto troch mištičiek pripočíta počet cukríkov v šiestej mištičke.

Na prvom riadku vstupu je N ($1 \leq N \leq 100,000$) – počet mištičiek. Na ďalšom riadku je N medzerou oddelených čísel. Tieto čísla sú celé, kladné a nepresahujú miliardu. Na jediný riadok výstupu vypíšte `NEDA SA` alebo najmenší počet cukríkov, ktoré dostane Maroško.

Príklady:

vstup
```
3
4 1 5
```
výstup
```
1
```

vstup
```
4
3 6 7 1
```
výstup
```
NEDA SA
```

vstup
```
9
8 6 12 8 8 4 4 4 6
```
výstup
```
4
```

K: Rozdeľovanie sociálneho grafu

50 bodov — Časový limit: 1500 ms

Keďže autorom zadaní ZENITu bolo vyčítané, že píšu dlhé texty, rozhodli sme sa príbeh vedúci k tejto úlohe vypustiť. Takže o divokých konfliktoch, chlpatých opiciach, predaji zubných kefiek rôznych farieb a samozrejme Lukášovi sa dočítate možno nabudúce.

Kto sa dočítal až sem, zrejme pozná pojem grafu (je to to isté, čo v úlohe o párty). Graf pozostáva z vrcholov a hrán. V našom prípade sa budeme baviť o neorientovanom ohodnotenom grafe, pričom hodnoty hrán budú nezáporné čísla.

Chceme ofarbiť vrcholy grafu dvoma farbami: čiernou a červenou. Skóre ofarbenia je súčet hodnôt hrán, ktoré vedú medzi vrcholmi rôznych farieb. Prečítajte zo vstupu graf a vypíšte najvyššie možné dosiahnuteľné skóre na tomto grafe.

Na prvom riadku vstupu je počet vrcholov N ($1 \leq N \leq 24$) a počet hrán M ($0 \leq M \leq 50,000$). Na ďalších M riadkoch sú trojice čísel: popisy hrán. Prvé dve čísla sú vrcholy (v rozmedzí 1 a N), posledné číslo je ohodnotenie hrany (celé, nezáporné, nepresahujúce 1,000,000). Môžete predpokladať, že každé dva vrcholy sú spojené najviac jednou hranou a že každá hrana spája dva rôzne vrcholy.

Poznámka: pre polovicu vstupov bude platiť $N \leq 20$. Za túto úlohu je relatívne ľahké získať približne polovicu bodov, ale relatívne ťažké získať všetky. Na testovači máte okrem nasledovných príkladov ešte jeden, kde $N = 24$.

Príklady:

vstup

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

výstup

```
11
```

vstup

```
4 5
1 2 7
4 2 6
3 1 9
4 1 5
3 4 9
```

výstup

```
31
```

vstup

```
6 9
1 2 15
1 3 3
2 3 1
2 4 4
4 3 3
3 6 2
6 5 7
5 3 7
4 5 7
```

výstup

```
41
```

Jedným z optimálnych riešení je ofarbiť vrcholy číslo 2 a 5 na červeno a zvyšné vrcholy na čierno. Hrany, spájajúce vrcholy rôznych farieb, sú (1,2),(2,3),(2,4),(3,5),(4,5) a (5,6). Skóre je teda 15+1+4+7+7+7 = 41. Inou možnosťou na dosiahnutie optimálneho skóre je zafarbiť na červeno vrcholy 1,4 a 6 (a zvyšné na čierno).