

# 9x tabela in 9x niz

## Tabela

### 1. Ocene študentov

Študent mora v 1. letniku opraviti izpite iz 15 predmetov. Njegove ocene hranimo v dveh tabelah: tabela `izpit` hrani ocene izpitov, tabela `vaje` pa ocene vaj. Če študent še ni polagal izpita pri določenem predmetu, je vpisana ocena 0 (pri izpiti in vajah), v nasprotnem primeru pa ocena zadnjega polaganja.

```
const int MAX_IZPITOV = 15;
int[] izpit = new int[MAX_IZPITOV];
int[] vaje = new int[MAX_IZPITOV];
```

- Najprej z uporabo generatorja naključnih števil napolni obe tabeli z vrednostmi med 0 in 10. Pri polnjenju bodite pozorni na medsebojno skladnost podatkov v obeh tabelah:
  - a) če ima študent oceno izpita 0, mora biti tudi ocena vaj 0 (študent še ni polagal izpita);
  - b) če ima študent negativno oceno izpita, mora biti ocena vaj 0 (na naši fakulteti velja pravilo, da se ocena vaj vpiše šele takrat, ko študent opravi izpit);
  - c) če ima študent pozitivno oceno izpita, mora biti tudi ocena vaj pozitivna.
- Nato :
  - a) izračuna povprečno oceno izpitov in povprečno oceno vaj, pri čemer upošteva samo pozitivne ocene
  - b) izračuna indeks predmeta, kjer je bila dosežena najvišja ocena izpita, najvišja ocena vaj in najvišja skupna ocena izpita in vaj
  - c) izračuna standardno deviacijo ocen izpitov in vaj, upoštevajoč samo pozitivne ocene. Formula za izračun standardne deviacije je naslednja:

$$\bar{x} = \frac{\sum x_i}{n} \quad i=0,1,\dots,n-1$$
$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \quad i=0,1,\dots,n-1$$

Izračunane podatke vpišite v tabele `povOc`, `maxOc` in `std` tako, da velja naslednje:

- `povOc[0]` vsebuje povprečno oceno izpitov;
  - `povOc[1]` vsebuje povprečno oceno vaj;
  - `maxOc[0]` vsebuje indeks predmeta, kjer je študent dosegel najvišjo oceno izpita;
  - `maxOc[1]` vsebuje indeks predmeta, kjer je študent dosegel najvišjo oceno vaj;
  - `maxOc[2]` vsebuje indeks predmeta, kjer je študent dosegel najvišjo skupno oceno izpita in vaj;
  - `std[0]` vsebuje standardno deviacijo za ocene izpitov;
  - `std[1]` vsebuje standardno deviacijo za ocene vaj;
- Na koncu pregledno izpišite vrednosti tabel `izpit`, `vaje`, `povOc`, `maxOc` in `std`.

### 2. Kozarci marmelade

V neki tovarni polnijo marmelado v 700-gramske kozarce. Delovanje polnilne naprave preverjajo tako, da v vsaki izmeni (tovarna dela v treh izmenah) naključno izberejo `STvZORCEV` kozarcev in jih stehtajo. Podatke, ki jih dobijo pri tehtanju, zabeležijo v dvodimenzionalno tabelo `vzorec`, ki ima 3 vrstice (vsaka vrstica predstavlja eno izmeno) in `STvZORCEV` stolpcev (vsak stolpec ustreza enemu vzorcu).

```
const int STvZORCEV = 20;
int[,] vzorec = new int[3,STvZORCEV];
```

- Z uporabo generatorja naključnih števil napolnite tabelo `vzorec` z vrednostmi med 670 in 730.
- Za vsako izmeno posebej:
  - a) izračunajte povprečno maso vseh `STVZORCEV` naključno izbranih kozarcev
  - b) poišče indeks tistega kozarca, ki najbolj odstopa od predpisane mase 700 gramov.

Izračunane podatke vpišite v tabeli `pv` in `ik` tako, da velja naslednje:

- `pv[0]` vsebuje povprečno maso kozarcev, odvzetih v prvi izmeni;
  - `pv[1]` vsebuje povprečno maso kozarcev, odvzetih v drugi izmeni;
  - `pv[2]` vsebuje povprečno maso kozarcev, odvzetih v tretji izmeni.
  - `ik[0]` vsebuje indeks tistega kozarca iz prve izmene, ki najbolj odstopa od predpisane mase 700 g;
  - `ik[1]` vsebuje indeks tistega kozarca iz druge izmene, ki najbolj odstopa od predpisane mase 700 g;
  - `ik[2]` vsebuje indeks tistega kozarca iz tretje izmene, ki najbolj odstopa od predpisane mase 700 g;
- Pregledno izpišite vrednosti tabel `pv` in `ik`.

### 3. Trgovine

V podjetju, ki ima `StTRGOVIN` trgovin, spremljajo prodajo po posameznih trgovinah za obdobje `StLET` let. Podatki o prodaji (tj. vrednost vseh prodanih izdelkov v tisočih €) so shranjeni v dvodimenzionalni tabeli `prodaja`, ki ima `StTRGOVIN` vrstic (vsaka vrstica ustreza eni trgovini) in `StLET` stolpcev (vsak stolpec ustreza enemu letu).

Tabela prodaja je deklarirana kot:

```
int[,] prodaja = new int[StTRGOVIN, StLET];
```

- Z uporabo generatorja naključnih števil napolni tabelo z vrednostmi med 10 in 1000.
- Za vsako trgovino posebej izpišite: vrednost prodanih izdelkov v vsakem letu, nato pa še vsoto vrednosti prodaje trgovine v vseh letih skupaj.
- Za vsako leto izpišite vsoto in povprečno vrednost prodaje vseh trgovin skupaj.
- Poiščite indeks trgovine, ki je v letu z indeksom `let0` (podatke, ki ga prebrete – zagotovite, da je smislen!) imela največjo vrednost prodaje. Če je takih trgovin več, izpišite indeks prve izmed njih.
- Naredite celoštevilsko tabelo z dvema elementoma: prvi predstavlja indeks trgovine, drugi pa indeks leta, v katerem je bila dosežena največja prodaja nasploh. Če je rešitev več, vrnite indeks zadnjega leta.
- Pregledno izpišite to tabelo.

### 4. Trgovsko podjetje

Neko trgovsko podjetje ima `StTRGOVIN` trgovin. V tabeli `promet` so shranjeni podatki o vrednosti prodanega blaga za vsak dan preteklega leta. Tabela je tridimenzionalna: prva dimenzija predstavlja `StTRGOVIN` trgovin, druga 12 mesecev in tretja 31 dni v vsakem mesecu (če ima mesec manj kot 31 dni, imajo odvečni elementi tabele vrednost 0).

- Z uporabo generatorja naključnih števil napolni tabelo `promet` z vrednostmi med 10 000 in 10 000 000. Bodite pozorni na elemente tabele, ki morajo biti 0. Prsetopna leta zanemarite!
- Za vsako trgovino posebej izračunajte, koliko znaša vrednost prodanega blaga v celem letu, tako da sešteje podatke po dnevih in mesecih ter dobljeno vsoto vpiše v tabelo `letPrometTrg`.
- Izpišite vrednosti tabele `letPrometTrg`.

### 5. Temperature

Napišite razred `Temperature`, ki omogoča statistično obdelavo podatkov o povprečnih mesečnih temperaturah zadnjih desetih let v `KOLIKOKRAJEV` slovenskih krajih. Podatke hranimo v tridimenzionalni tabeli, kjer prva

dimenzija predstavlja KOLIKOkRAJEV krajev, druga 10 let in tretja 12 mesecev vsakega leta. Imena krajev so v tabeli nizov.

- Z uporabo generatorja naključnih števil napolni tabelo z ustreznimi decimalnimi števili. Da bi bile generirane vrednosti kar najbolj realne, upoštevajte, da so povprečne temperature:
  - pozimi (december, januar in februar) med -20.0 in +5.0 C;
  - spomladi (marec, april in maj) med 0.0 in 25.0 C in naraščajo – marec ima najmanjše, maj največje;
  - poleti (junij, julij in avgust) med 20.0 in 35.0 C;
  - jeseni (september, oktober in november) med 0.0 in 25.0 C in padajo – september ima največje, november najmanjše;
- Za kraj z imenom *ime* izpišite za leto *l* vse mesečne temperature in celoletno povprečje (celoletno povprečje dobimo tako, da seštejemo povprečne temperature po posameznih mesecih in jih delimo z 12).
- Za vsako leto izpišite celoletno povprečje za vsak kraj posebej in za vse kraje skupaj (povprečna vrednost vseh temperatur v vseh mesecih in krajih določenega leta).
- Izpišite tri kombinacije indeksov kraja, leta in meseca, kjer so bile dosežene tri najnižje mesečne temperature nasploh. V primeru, da je več enakovrednih rezultatov, izpišite vse.

## 6. Znanstvena konferenca

Na znanstveni konferenci bodo avtorji iz 25 držav EU predstavili tri vrste prispevkov (referate, kratke referate in posterje), ki sodijo v pet različnih področij ("kako postati bogart", "zapravljanje časa", "učinkovito lenarjenje", "mlatenje prazne slame" in "pomembno je biti politik"). V tridimenzionalni tabeli prispevki so shranjeni podatki o številu prispevkov posamezne vrste po državah in posameznih področjih: prva dimenzija predstavlja 25 držav, druga tri vrste prispevkov in tretja pet področij. Poleg tega imamo na voljo še tri tabele v katerih so shranjena imena držav, vrsta prispevkov in področje.

- Z uporabo generatorja naključnih števil napolni tabelo prispevki z vrednostmi med 0 in 20.
- Izpišite skupno število vseh prispevkov na konferenci
- Izpišite skupno število prispevkov vsake vrste.
- Izpišite ime tistega področja, s katerega bo predstavljenih največ prispevkov. Če je takšnih področij več, izpišite imena vseh področij.

## 7. Kdo je izdal skrivnost?

Telefonski operaterji zbirajo tako imenovane prometne podatke o telefonskih pogovorih: kdo je koga klical in kdaj. Čeprav pri tem zbiranju še ne gre za prisluškovanje pogovorom, so tudi prometni podatki strogo zasebni podatki, saj se da na njihovi podlagi marsikaj sklepati o klicocih in jih lahko operaterji posredujejo preiskovalnim organom samo na izrecno zahtevo sodišča ob utemeljenem sumu kaznivih dejanj (ko lahko sodišče odredi tudi prisluškovanje imetnikom izbranih telefonskih števil).

Pa recimo, da teh moralnih dilem nimamo. Dobili smo zaporedni seznam prometnih podatkov nekega operaterja o vseh telefonskih klicih v nekem časovnem obdobju. Elementi seznama so pari telefonskih števil, ki sta uspešno vzpostavili telefonsko zvezo, torej takole (zaradi enostavnosti bomo pri tej nalogi predpostavili, da so vse telefonske številke štirimestne, npr. interne številke nekega podjetja):

1705 2312  
1326 1705  
3789 1230  
2312 2372  
0137 1705

in tako dalje. Zaradi enostavnosti v seznamu niso zapisani casi klicev, vendar je seznam urejen po časovnem zaporedju, torej prvi element seznama je prvi klic v opazovanem časovnem obdobju in tako naprej.

Zanima nas, ali je neka zaupna informacija, ki jo je imel lastnik telefonske številke *a*, lahko prek telefonskih pogovorov prišla do osebe, ki je lastnik telefonske številke *b*? Mogoče se je oseba *a* z *b*-jem pogovarjala neposredno (potem je odgovor preprost), lahko pa se je *a* najprej pogovarjal z nekim *c*, potem je *d* poklical *c*-ja in nazadnje je *d* poklical *b*-ja. Informacija od *a* do *b* bi lahko prišla tudi še po kakšni drugacni poti, npr. tako, da *e* najprej pokliče *a*-ja in potem *b* pokliče *e*-ja.

Denimo, da je seznam klicev dan v tabeli `klici`. Sestavite program, ki pregleda seznam vzpostavljenih zvez (tabela `klici`) in za dani dve telefonski številki iz seznama, npr. `a` in `b`, ugotovi, ali je lastnik telefonske številke `b` lahko prišel do zaupne informacije, ki jo je imel lastnik številke `a`.

**Namig:**

Ker so telefonske številke pri tej nalogi le štirimestne, je možnih samo 10 000 telefonskih števil. V neki tabeli (denimo `obvescen`) si bomo za vsako številko zapomnili, ali obstaja primerno zaporedje klicev, s katerim bi lahko lastnik te številke dobil informacijo od lastnika številke `a`. Telefonske pogovore pregledujemo po naraščajočem času; pri pogovoru med `u` in `v` upoštevamo, da ce je pred tem pogovorom eden od njiju že imel a-jevo informacijo (kar vidimo v tabeli `obvescen`), jo bo imel po pogovoru tudi drugi (in tudi to vpišemo v tabelo `obvescen`). Na koncu tega postopka imamo v `obvescen[b]` podatek o tem, ali je a-jeva informacija prišla tudi do b-ja ali ne.

Seveda čim med pregledovanjem seznama pogovorov opazimo, da je informacija dejansko lahko prišla do b-ja, prenehamo z delom, saj smo že dobili odgovor, ki nas zanima.

**8. Suduku**

Sudoku je številčna križanka. Igralno polje velikosti  $9 \times 9$  kvadratkov je dodatno razdeljeno na devet manjših kvadratov velikosti  $3 \times 3$  (na spodnji sliki so predstavljeni z debelejšimi črtami), vanj pa je vpisanih nekaj števil (od 1 do 9). Primer:

							7	
8								
			7					
			4					
						8		
	6							

V igralno polje igralec vpisuje števila od 1 do 9 in to tako, da so na koncu izpolnjeni naslednji trije pogoji:

- (1) v vsakem stolpcu se mora vsako število od 1 do 9 pojavljati natanko enkrat;
- (2) v vsaki vrstici se mora vsako število od 1 do 9 pojavljati natanko enkrat;
- (3) v vsakem od devetih malih kvadratov velikosti  $3 \times 3$  se mora vsako število od 1 do 9 pojavljati natanko enkrat.

Na spodnji sliki levo vidimo primer pravilno izpolnjenega polja. Desno polje pa je izpolnjeno napačno (med drugim zato, ker v zadnjem stolpcu ni števila 6, se pa število 2 v njem pojavlja kar dvakrat).

7	6	9	3	1	4	5	8	2
1	4	2	6	5	8	3	7	9
3	8	5	7	2	9	1	6	4
6	9	3	1	7	5	2	4	8
8	1	4	2	9	3	6	5	7
5	2	7	4	8	6	9	3	1
2	7	8	5	6	1	4	9	3
4	5	1	9	3	7	8	2	6
9	3	6	8	4	2	7	1	5

7	6	9	3	1	4	5	8	2
1	4	2	6	5	8	3	7	9
3	8	5	7	2	9	1	6	4
6	9	3	1	7	5	2	4	8
8	1	4	2	9	3	6	5	7
5	2	7	4	8	6	9	3	1
2	7	8	5	6	1	4	9	3
4	5	1	9	3	7	8	6	2
9	3	6	8	4	2	7	1	5

Napiši program, ki bo prebral rešitev in izpisal niz PRAVILNA, če je rešitev pravilna, in NAPAČNA, če ni pravilna.

**9. Križanka**

Napiši podprogram, ki dobi kot vhod križanko, na izhod pa izpiše besede, ki se pojavljajo v križanki, skupaj z njihovimi položaji. Križanka je sestavljena iz `m` vrstic in `n` stolpcev. V vsakem polju je ali velika tiskana črka angleške abecede ali pa znak '\*', ki pomeni, da to polje razmejuje besede. Za besedo šteje strnjeno zaporedje vsaj dveh črk, ki je napisano navpično ali vodoravno in je na začetku in koncu ločeno od ostalih črk v križanki z znakom '\*' ali z robom križanke.

Vsa polja, na katerih se začnejo besede, oštevilčimo od 1 naprej. Štejemo po vrsticah od zgoraj navzdol, v vsaki vrstici pa od leve proti desni (glej primer).

Primer: recimo, da imamo podano naslednjo križanko.

SOLAR

\*LOK\*

SIPON

IMATO

RPR\*V

Preberemo jo kot 5 nizov (SOLAR, \*LOK\*, SIPON, IMATO, RPR\*V)

Najprej oštevilčimo vsa polja (od zgoraj navzdol, v vsaki vrstici od leve proti desni):

1234.

\*5..\*

6...7

8....

9....

Program mora v tem primeru izpisati:

1, vodoravno: SOLAR

1, navpično : -

2, vodoravno: -

2, navpično : OLIMP

3, vodoravno: -

3, navpično : LOPAR

4, vodoravno: -

4, navpično : AKOT

5, vodoravno: LOK

5, navpično : -

6, vodoravno: SIPON

6, navpično : SIR

7, vodoravno: -

7, navpično : NOV

8, vodoravno: IMATO

8, navpično : -

9, vodoravno: RPR

9, navpično : -

## Nizi

### 1. Analiza nizov

Najprej preberite nekaj stavkov, ki jih sestavlja več besed. Vsak stavek shranite v tabelo nizov in potem za vsak stavek

- izpišite število malih črk, ki jih vsebuje.
- izpišite število besed, ki jih vsebuje stavek. Pri tem predpostavite, da se lahko vsaka beseda začne s črko ali številko (in vsebuje poljubno število črk in številke), ločilo med besedami pa je '!', ', ' in/ali en ali več presledkov.
- Izpišite m-to besedo stavka. Če je  $m$  večji od števila besed, izpišite prazen niz.

### 2. Kodiranje nizov

Napisali bomo postopka, ki omogočata kodiranje niza tipa `string` v število tipa `long` (z njim delamo enako, kot s tipom `int`, le da omogoča uporabo števil do 9,223,372,036,854,775,807) in obratno. Pri tem naj velja:

- niz sestavljajo samo črke iz angleške abecede (26 znakov) in številke (10 znakov)
- v nizu ne razlikujemo med velikimi in malimi črkami
- nizi so lahko dolgi največ 10 znakov

Pri kodiranju si pomagamo s kodirno tabelo, kjer vsakemu izmed 36 znakov pripada ena vrednost (recimo a-ju pripada vrednost 1, b-ju vrednost 2, itd.). Vrednost niza izračunamo tako, da vrednost prvega znaka v nizu pomnožimo s 37, temu prištejemo vrednost drugega znaka in vse skupaj pomnožimo s 37, nato dodamo vrednost tretjega znaka in tako naprej do zadnjega znaka.

**Primer:**

A, a	B, b	C, c	D, d	...	N, n	...	R, r	...	O
1	2	3	4		14		18		27

$$\mathbf{ABRAN0} = 1 | 2 | 18 | 1 | 14 | 27 = (((1*37+2)*37+18)*37+1)*37+14)*37+27 = \mathbf{74005947}$$

Pri dekodiranju ravnamo obratno: ostanek pri deljenju s 37 nam pove kodo zadnjega znaka, nato pa kodo postopoma celoštevilsko delimo in določamo ostanke. S postopkom končamo takrat, ko je rezultat deljenja s 37 enak 0.

**Primer:**

$$\begin{aligned} 74005947 \% 37 &= 27 && \gg \mathbf{0} \ll \\ 74005947 / 37 &= 2000160, 2000160 \% 37 = 14 && \gg \mathbf{N} \ll \\ 2000160 / 37 &= 54058, 54058 \% 37 = 1 && \gg \mathbf{A} \ll \\ 54058 / 37 &= 1461, 1461 \% 37 = 18 && \gg \mathbf{R} \ll \\ 1461 / 37 &= 39, 39 \% 37 = 2 && \gg \mathbf{B} \ll \\ 39 / 37 &= 1, 1 \% 37 = 1 && \gg \mathbf{A} \ll \\ 1 / 37 &= 0 && \end{aligned}$$

Sestavi program, ki najprej prebere niz, ga zakodira, izpiše njegovo kodo, nato pa ga še odkodira in izpiše odkodiranega.

### 4. Popačena govorica

V tabeli `nizi`, ki jo morate kreirate sami, naj bo shranjenih KOLIKO nizov tipa `string`, ki jih želimo popačiti. To naredimo tako, da posamezen niz preoblikuje tako, da v vsaki besedi (besede so med seboj ločene z enim ali več presledki):

- odvzame prvo črko in jo prestavi na konec besede
- na koncu tako preoblikovane besede doda še črko 'a'.

**Primer:** Niz z besedilom "CSharp je objektno usmerjen programski jezik" se pretvori v "SharpCa eja bjektnooa smerjenua rogramskipa ezikja"

Pazite, da so nizi taki, da je med posameznimi besedami včasih tudi več kot en presledek. Vsaj en niz naj bo tak, da se konča z enim presledkom, vsaj en s tremi presledki, vsaj en niz pa tak, da so v njem le znaki (brez presledkov)

## 5. Šifriranje nizov

Niz  $n$  zašifrirajte na naslednji način:

- če je prvi znak niza črka 'o' ali 'O', mora preostanek niza "obrniti" tako, da se bo pravilno bral z desne proti levi
- če je prvi znak niza črka 'n' ali 'N', mora preostale znake v nizu nadomestiti z njihovimi nasledniki (naslednik je znak, katerega numerična koda je za 1 večja)
- če je prvi znak niza črka 'p' ali 'P', mora preostale znake v nizu nadomestiti z njihovimi predhodniki (predhodnik je znak, katerega numerična koda je za 1 manjša)

Poleg tega morate še nastaviti prvi znak vsakega zašifriranega niza tako, da bo možno ponovno dešifriranje (tj. pretvorba niza v prvotno obliko).

**Primer:**

- niz "oobrni" se pretvori v "oinrbo"
- niz "nnaslednji" se pretvori v "pobtmfeokj"
- niz "ppredhodni" se pretvori v "noqdcgncmh"

## 6. Kodiranje

Sestavite postopka za kodiranje in dekodiranje. Prvi niz  $n$  pretvori v drug niz tako, da vsako črko angleške abecede nadomesti z nizom, ki predstavlja zaporedno številko velike črke v angleški abecedi, vse ostale znake (tudi števke) pa naj ohrani nespremenjene. Da bi omogočili enolično dekodiranje črk, je treba 'A' in 'a' nadomestiti z "01" ter 'B' in 'b' z "02", medtem ko ostale črke lahko nadomestimo z eno ali dvema ciframa ('C' in 'c' s "3", 'D' in 'd' s "4", ... ter 'Z' in 'z' s "26").

Drugi postopek naj tako zakodiran niz pretvori spet v prvotno obliko. Če ga uporabimo s pokvarjenim zakodiranim nizom, naj niz pretvori v "To pa ne gre. Napaka na mestu: " + mestoNapake v kodiranem nizu (primer pokvarjenega niza je recimo niz "03").

**Primer:** Tako prvi postopek niz (podčrtaji so le zaradi lepše vidnosti) "RACUNALNIK JE ZAKON !" pretvori v "180132114011214911 105 2601111514 !", drugi pa ta niz pretvori ponovno v "RACUNALNIK JE ZAKON !".

## 7. Stiskanje nizov

Nize, v katerih se neka črka večkrat ponovi, lahko zapišemo krajše na naslednji način:

AAA	zapišemo kot	A3
ABBCCD	zapišemo kot	AB2CD
AAAAEEEEEEEEEEEEUU	zapišemo kot	A4E11U

Kot je razvidno iz primera, stisnjena beseda ni nikoli daljša od originala, saj število zaporednih nastopanj vsake črke dodamo samo v primeru, ko se črka dejansko ponavlja.

Napišite program, ki bo omogočal stiskanje in ponovno razširjanje besed. Program naj bo omejen na besede, ki so sestavljene samo iz črk (a-z in A-Z), upoštevajte pa, da je število pojavitev lahko tudi večmestno število.

V njem najprej preberemo tabelo  $t_1$  z 10 besedami, besede stisnemo in zapišemo v tabelo  $t_2$  ter stisnjene besede zopet razširimo in zapišemo v tabelo  $t_3$ . Program naj na koncu izpiše vse tri tabele.

## 8. Naraščajoče besede

V nekaterih besedah so črke že urejene naraščajoče po abecedi: vsaka črka take besede pride v abecedi kasneje kot prejšnja črka te besede. Takšnim besedam pravimo naraščajoče besede. Primer naraščajoče besede je AGILNOST — G je v abecedi kasneje kot A, I je kasneje kot G in tako naprej.

Napiši program, ki prebere n besed in izpiše vse najdaljše naraščajoče besede v njem. Predpostaviš lahko, da je v besedah nastopajo samo velike črke angleške abecede.

## 9. "Pet čevljev merim, palcev pet,"

. . . je tarna mlad trgovec, ko je čez lužo odprl trgovino z blagom. Sprva se mu niti sanjalo ni, da mu utegne pretvarjanje enot povzročati toliko preglavic.

"Kako že gre? 1 liga so 3 milje, 1 milja je 8 furlongov, 1 furlong je 220 jardov, 1 jard so 3 čevlji, 1 čevlj so 3 dlani, 1 dlan so 4 palci," je drdral v mislih. Uh, saj to je še šlo, a ko je nekaj kupcev naročilo blago, se je pri seštevanju in naročanju blaga v tovarni krepko uštel.

Pomagaj mu sestaviti program, ki prebere deset vrstic s po sedmimi števili (ki pomenijo zaporedoma število lig, milj, furlongov, jardov, čevljev, dlani in palcev blaga, ki so jih posamezne stranke naročile). Med števili je po en presledek. Tako je npr. primer vrstice

```
0 2 6 160 0 0 0
```

Nato preberi še vrstico sedmih števil, ki povedo količino blaga, naročenega v tovarni. Nato izpiše eno od naslednjih sporočil:

```
Naročil si *** preveč blaga.  
Naročil si *** premalo blaga.  
Naročil si ravno prav blaga.
```

Namesto zvezdic naj se izpiše količina preveč ali premalo naročenega blaga. Ta naj bo izražena tako, da uporabiš čim večje enote. Dolžino 5 000 jardov bi moral na primer napisati kot 2 milji, 6 furlongov in 160 jardov (ne pa kot 5 000 jardov ali pa kot 2 milji in 1 320 jardov ali pa kot 22 furlongov in 160 jardov ali pa celo kot 15 000 čevljev ali kaj podobnega). Količino zapiši kar s sedmimi števili, ločenimi s presledki, torej v enaki obliki, v kakršni so zapisane posamezne količine tudi v vhodnih podatkih. Dolžino 5 000 jardov bi tako zapisal kot 0 2 6 160 0 0 0.