

## ATBASH slaptaraštis

Atsiradus raštui, kurį supranta visi raštingi žmonės, natūralu, kad atsirado poreikis ieškoti būdų, skirtų informacijai paslėpti (užkoduoti). Pirmieji skaitymą „apsunkino“ egiptiečiai, kurie vietoj raidžių naudojo savus hieroglifus, tačiau dar iki šiol nėra aišku, ar taip buvo daroma dėl to, kad informaciją galėtų perskaityti tik tie, kam tas raštas skirtas, ar dėl kitų priežasčių. Tačiau yra aišku, kad slaptaraštį tikrai naudojo žydų raštininkai. Vienas iš jų naudojamų slaptaraščių – ATBASH, kurio esmė, kad vietoj vienu raidžių yra naudojamos kitos. Sukurkite programą, kuri iššifruotų eilutes, užkoduotas ATBASH slaptaraščiu.

ATBASH slaptaraščio esmė tokia, kad pirmoji pateiktos abėcėlės raidė atitinka paskutinąją, antroji - priešpaskutinę ir t.t. Pavyzdžiui, lietuviškai abėcėlei būtų pritaikomi tokie raidžių pakeitimai:

A	Ą	B	C	Č	D	E	Ę	Ė	F	G	H	I	Į	Y	J
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Ž	Z	V	U	Ū	Ų	T	Š	S	R	P	O	N	M	L	K

Tačiau Jūsų užduotis: sukurti programą, kuri naudoja **lotynišką 26 raidžių abėcėlę** ir iššifruoja tekstiniame faile pateiktus sakinius (iššifruojama kiekviena faile pateikta eilutė).

Programos duomenys: duomenų failo **duomenys.txt pirmoje eilutėje** - tekstiniame faile esančių užšifruotų eilučių kiekis (n). Kitose **n** eilučių pateiktos ATBASH šifru užkoduotos eilutės.

Programos rezultatai: rezultatų failo **rezultatai.txt** atskirose eilutėse surašytos iššifruotos duomenų faile buvusios eilutės (kiekvieną duomenų failo eilutę atitinka viena rezultatų failo eilutė).

Pradiniai duomenys	Rezultatai
1 Ozyzh	Labas
2 Ozyzh Svool_Dliow!	Labas Hello_World!
3 Ozyzh Svool_Dliow! OzYzH_oZyZh	Labas Hello_World! LaBaS_lAbAs

### Pastabos:

- Šifruojami tik lotyniškoje abėcėlėje esantys simboliai, jei eilutėje yra kitų simbolių - jie paliekami tokie, kokie yra.
- Atkreipkite dėmesį į tai, kad didžiosios raidės iššifravus lieka didžiosiomis. Atitinkamai ta pati taisyklė galioja ir mažosioms raidėms.

- Naudokite lotynišką abėcėlę (26 raidžių).
- Šiame uždavinyje žodžiai atskiriami apatiniu brūkšneliu.

### Reikalavimai

- Parašyti duomenų skaitymo funkciją **void()**.
- Parašyti iššifruotų eilučių spausdinimo funkciją **void()**.
- Sukurti funkciją **Atbash()**, kuri per savo vardą grąžintų iššifruotą eilutę. Funkcijos tipas - **string()**.
- Realizuoti funkciją, kuri kiekvienai nuskaitytai duomenų failo eilutei pritaikytų funkciją **Atbash()**. Funkcijos tipas - **void()**.

### Atblokavimas

Kadangi technologijos sparčiai tobulėja, telefonų gamintojai stengiasi padaryti kuo patogesnius telefonus. Kad atrakintų telefoną, Alex'ui reikia ištarti dvi raides. Alex'o priešas, Arkadijus, nori atrakinti Alex'o telefoną ir pavogti svarbią informaciją. Deja, Arkadijus gali ištarti tik **n** dviejų raidžių kombinacijų. Ištarimo eilė nesvarbi, taipogi jis gali ištarti iškart keletą raidžių kombinacijų (nebūtinai skirtingų) kad gautų raidžių kombinaciją, kurioje būtų slaptažodis. Parašykite programą, kuri nustatytų, ar Arkadijui pavyks.

Pradinių duomenų faile **duomenys.txt** yra tokie duomenys:

Pirmoje duomenų failo eilutėje dvi raidės - slaptažodis.

Antroje eilutėje **n** - kiek raidžių kombinacijų gali ištarti Arkadijus.

Tolimesnėse **n** eilučių seka raidės, kurias Arkadijus gali ištarti.

Į rezultatų failą **atsakymas.txt**

Išveskite vieną žodį - **Taip** arba **Ne**

duomenys.txt	atsakymas.txt
ya 4 ah oy to ha	Taip

Pavyzdžio paaiškinimas: Arkadijus gali ištarti "oyah" kuris turi dvi slaptažodžio raides - "ya", atrakinimas pavyks.

### Reikalavimai

Kad gautumėte maksimalų įvertinimą, neviršykite 100 eilučių.

## Teksto komentarų analizė

Duotas teksto fragmentas, kuriame tarp eilučių yra įterpta komentarų. Kiekvienas komentaras atskirtas simboliškai "\*\*\*" iš abiejų pusių. Rastus komentarus reikia iškirpti iš teksto ir kartu įsiminti, iš kurios eilutės jis paimtas. Rezultatų faile išspausdinti tekstą be komentarų su sunumeruotomis eilutėmis, už teksto išspausdinti komentarų sąrašą: iš kurios eilutės jis iškirptas ir pats komentaras.

duomenys.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit.\*\*Pirmas komentaras\*\* Nam auctor neque tortor, sit amet lacinia erat eleifend sit amet. Cras nibh arcu, scelerisque non tempor sagittis, suscipit ac orci. Sed sollicitudin ante diam, et porttitor metus vulputate ut. Donec sit amet dui nunc.\*\*Komentaras trecioje eiluteje\*\* Pellentesque diam risus,\*\*Komentaras toje pacioje eiluteje\*\* condimentum elementum diam sit amet, accumsan \*\*Komentaras prasidedantis pradžioje\*\*ullamcorper lectus. Curabitur luctus egestas risus, et vestibulum arcu sollicitudin eget. Proin elementum consequat urna ac luctus. Sed eu pellentesque justo, tempor porttitor dui. Sed sollicitudin sem ut magna aliquet aliquet.\*\*Pabaiga\*\*\*Komentaras einantis iskarto uz kito komentaro\*\*

rezultatai.txt

1: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam auctor neque tortor, sit amet lacinia erat eleifend sit  
2: amet. Cras nibh arcu, scelerisque non tempor sagittis, suscipit ac orci. Sed sollicitudin ante diam, et porttitor metus  
3: vulputate ut. Donec sit amet dui nunc. Pellentesque diam risus, condimentum elementum diam sit amet, accumsan  
4: ullamcorper lectus. Curabitur luctus egestas risus, et vestibulum arcu sollicitudin eget. Proin elementum consequat urna  
5: ac luctus. Sed eu pellentesque justo, tempor porttitor dui. Sed sollicitudin sem ut magna aliquet aliquet.

Komentarai

1: Pirmas komentaras  
3: Komentaras trecioje eiluteje  
3: Komentaras toje pacioje eiluteje  
4: Komentaras prasidedantis pradžioje  
5: Pabaiga  
5: Komentaras einantis iskarto uz kito komentaro

## Reikalavimai

- Sukurti teksto nuskaitymo į **string** masyvą funkciją.
- Sukurti teksto apdorojimo funkciją, kuri atskiria komentarus nuo teksto.
- Sunumeruoti rezultatuose teksto eilutes.
- Išspausdinti, iš kurios eilutės paimtas komentaras.

## IP adresai

Sukurkite programą, kuri nustatytų IP adreso validumą (tinkamumą) pagal pateiktas **taisykles**:

- IP adresas turi būti sudarytas iš 4 dalių.
- IP adresas turi būti vien iš skaičių ir taško simbolių.
- IP adreso dalys gali būti nuo 0 iki 255

Duomenų faile (Duomenys.txt) pateikiamas nežinomo dydžio (bet ne daugiau nei 10) IP adresų sąrašas, po vieną eilutėje.

Rezultatų faile turite išspausdinti IP adresą ir atsakymą, teisingas jis ar neteisingas.

Duomenys.txt	Rezultatai.txt
199.145.148.97	199.145.148.97 Teisingas
83.33.28.184	83.33.28.184 Teisingas
144.174.24	144.174.24 Neteisingas
219.135.ABC.202	219.135.ABC.202 Neteisingas
256.0.15.1	256.0.15.1 Neteisingas

## Reikalavimai

- Skaitymo funkcija.
- Spausdinimo funkcija.
- Vieno IP adreso tikrinimo funkcija, grąžinanti bool tipo reikšmę.

## Straipsnių pavadinimai

Įsivaizduojamoje planetoje Gyvūnų žemė yra leidžiamas laikraštis. Laikraščio kiekvieno straipsnio pavadinimas  $s_1$  sudarytas iš mažųjų 26 lotyniškos abėcėlės raidžių. Planetos gyventojas, ruoniukas Jonas, nori nusipirkti keletą tokių laikraščių, iškirpti antraštes bei suklijuoti jas į vieną didelį žodį. Suklijavęs antraštes, Jonas iškerpa tam tikras raides iš gauto žodžio, kad būtų gautas naujas žodis  $s_2$ . Laikoma, kad iškirpus raidę iš žodžio, nepaliekamas tarpas, t.y., žodis išlieka vientisas tik be iškirptų raidžių. Be to, iškirptų raidžių keletą kartų naudoti negalima.

Pavyzdžiui, jeigu straipsnio pavadinimas yra "ada", suklijavus du tokius pačius pavadinimus, gauname "adaada". Jeigu pašalinamos 1 ir 6 raidės, gaunamas naujas žodis "daad".

Kiek mažiausiai reikia laikraščio antraščių  $s_1$ , kad Jonas jas suklijavęs bei pašalinęs tam tikras raides gautų žodį  $s_2$ ?

Duomenų faile **in.txt** yra dvi teksto eilutės. Pirmojoje eilutėje yra straipsnio pavadinimas  $s_1$ , antrojoje žodis  $s_2$ . Žodžiai gali būti sudaryti iš mažųjų 26 lotynų abėcėlės raidžių ( $1 \leq |s_1| \leq 10^4$ ,  $1 \leq |s_2| \leq 10^6$ ).

Rezultatų faile **out.txt**, jeigu neįmanoma suklijuoti žodžio  $s_2$  pagal nurodytas sąlygas, išvedamas "-1" (be kabučių). Kitu atveju turi išvestas skaičius, kuris apibrėžia **mažiausią reikalingą laikraščio antraščių  $s_1$  kiekį**, kuris reikalingas Jonui suklijuoti žodį  $s_2$ .

**Pastabos**

- Atkreipkite dėmesį į tai, kad žinomas maksimalus  $s_1$ .
- Simbolius galima užrašyti sveikaisiais skaičiais. ASCII lentelėje galima rasti, jog a-z mažosios lotyniškos raidės atitinka 97-122 sveikuosius skaičius.

in.txt	out.txt	Detalus pavyzdys
jonas qwerty	-1	<p>3 pavyzdžio paaiškinimas, kaip gautas rezultatas.</p>
jonas jonas	1	
qwerty qwqyyyq	4	
ab baaabba	6	

## Reikalavimai

- Parašyti duomenų skaitymo funkciją **void()**.
- Uždavinį išspręsti optimizuotai. Kad tai būtų pasiekta, per  $s_1$  ir  $s_2$  galima iteruoti maksimaliai vieną kartą. Be to, turi būti sukurta funkcija, kurios vienas iš parametru dvimatis masyvas **int[][26]**, kurį funkcija turi užpildyti. Šiame dvimačiame masyve saugoma kiekvienai  $s_1$  pozicijai arčiausia kitų abėcėlės raidžių pozicija žodyje.
- Sukurti funkciją, kuri per savo vardą grąžintų reikalingų  $s_1$  žodžių kiekį, o vienas iš funkcijos parametru būtinai būtų apskaičiuotas dvimatis masyvas **int[][26]**. Funkcijos tipas **int()**.

## Pasaka

Sukurkite programą, kuri padėtų autoriui pakeisti tekste esančius vardus. Turite vieną duomenų failą - **pasaka.txt**, kuriame surašytas tekstas, tada yra **viena eilutė kurioje padėtas** ; (kabliataškis) ir po jos **kiekvienoje naujoje eilutėje** surašyti vardų rinkiniai: **senas\_vardas naujas\_vardas**. Nuskaitytą tekstą su juo atlikite veiksmus taip, kad tekste senus vardus pakeistų nauji. Tekstą su pakeistais vardais spausdinkite į

failą **pasakarez.txt**. Turi būti pakeisti tik vardai, kiti simboliai ir žodžiai turi išlikti tokie kokie buvo. **Į linksnius nekreipkite dėmesio**. Keiskite tik tuos vardus, kurių linksniai sutampa.

pasaka.txt	pasakarez.txt
<p>Tomas_nusipirko_dviratį.            Jam_nepatiko_jo_naujas_dviratis            Todėl_Tomas_jį_pardavė_ir_nusipirko            paspirtuką.            ;            Tomas Marius</p>	<p>Marius_nusipirko_dviratį.            Jam_nepatiko_jo_naujas_dviratis            Todėl_Marius_jį_pardavė_ir_nusipirko            paspirtuką.</p>
<p>Lukas_yra_ledų_gurmanas.            Lukas_labai_mėgsta_vanilinius.            Todėl_jam_Karolis_davė_du_eurus.            Lukas_padėjo_draugui_už_paskolintus_pinigis            ledams.            ;            Lukas Darius            Karolis Tomas</p>	<p>Darius_yra_ledų_gurmanas.            Darius_labai_mėgsta_vanilinius.            Todėl_jam_Tomas_davė_du_eurus.            Darius_padėjo_draugui_už_paskolintus_pinigis            ledams.</p>
<p>Paulius_mėgsta_sportą.            Jis_į_komandą_kviečia_draugus.            Darius_Andrius_ir_Lukas            sutiko_būti_su_juo_komandoje.            ;            Paulius Petras            Darius Domas            Andrius Algis            Lukas Laurynas</p>	<p>Petras_mėgsta_sportą.            Jis_į_komandą_kviečia_draugus.            Domas_Algis_ir_Laurynas            sutiko_būti_su_juo_komandoje</p>

### Reikalavimai

- Pradinių duomenų skaitymo funkcija **void()**.
- Funkcija **void()**, pertvarkanti tekstą .
- Duomenų rašymo į failą funkcija **void()**.

### Kiti formatai

Programai pateikiamas tekstas be tarpų vienoje eilutėje faile **nesuprantamas.txt**, tame tekste visos š raidės pakeistos dviem simboliais: &š;. Programa turi pakeisti visus užrašus &š; į š raides ir išsaugoti rezultatą faile **suprantamas.txt**.

nesuprantamas.txt    suprantamas.txt	
abcabc&š;	abcabcš
abc&š;abc	abcšabc
abc&š;&š;abc	abc&šš;abc

### Reikalavimai

- Funkcija, kuri pakeistų &š; į š raides nepašalinant tvarkos ir kitų raidžių.
- Funkcija, kuri išsaugotų rezultatus faile.