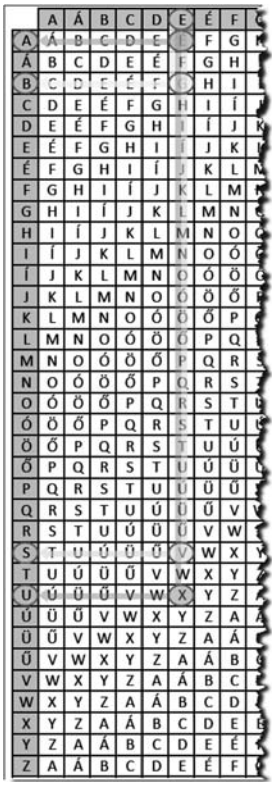


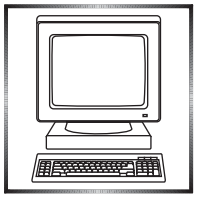
Ha ezek vannak az egyes csoportok gyakorisági táblázatának élén, akkor ezekké kódolódott a magyarban leggyakoribb betű, az E. Nézzük meg a Vigenére-tábla E oszlopát és keressük meg, melyik sorokban találjuk ezeket a betűket:

a G-t	a B oszlopban;
az X-et	az U oszlopban;
a V-t	az S oszlopban, végül
az É-t	az A oszlopban.

A kapott betűket a fenti sorrendben már csak össze kell olvasnunk és meg is van a kulcs: BUSA. A kulcs ismeretében alig valamivel nehezebb kihámozni a titkos üzenet értelmét, mintha a kódolatlan üzenetet nyomták volna a markunkba.

Végkövetkeztetésül: Ha elég hosszú a szöveg, továbbá ismerjük az adott nyelv leggyakrabban használt betűjét, akkor bizony a Vigenère-kódolású üzenet is megfejthető a kulcs ismerete nélkül. Babbage munkája új kihívás elé állította a rejtjelezőket, így született meg az ENIGMA, az RSA és az MD5 kódolás, de ez már egy másik történet. Hiábavaló volt tehát a sok fáradság, amivel kitalálták e furfangos titkosítási módszert, és a kódolásra fordított energia is kárba veszett a zseniális elmével szemben, és ez így volt törvényszerű. Hiszen kell lennie valamilyen értelmes rendszernek a kódoláskor, mert ha például a Bibliát szeretnénk titkosítani, mondjuk úgy, hogy ábécébe rendezzük a szavait, akkor sem lehetnénk biztosak benne, hogy némi próbálkozás után az eredeti szöveget kapjuk vissza. A Vigenère-módszer utódai is megfejtve végezték, vagy végzik majd.

Tóth Tamás



Informatikából kitűzött feladatok

I. 547. A morzekód (Samuel Morse találmánya) olyan kommunikációs kód, amely pontok és vonalak kombinációjából áll. Szöveges üzenet átvitelére alkalmas vezetékes, vagy vezeték nélküli kommunikációs csatornán.

A kódrendszer rövid és hosszú jelekből, valamint szünetekből áll. A morzekóddal a magyar ábécé ékezetmentes betűi egyértelműen kódolhatók a következő táblázat alapján:

Betű	**Jel**	**Betű**	**Jel**
A	· —	N	— ·
B	— · · ·	O	— — —
C	— · — ·	P	· — — ·
D	— · ·	Q	— — · —
E	·	R	· — ·
F	· · — ·	S	· · ·
G	— — ·	T	—
H	· · · ·	U	· · —
I	· ·	V	· · · —
J	· — — —	W	· — —
K	— · —	X	— · · —
L	· — · ·	Y	— · — —
M	— —	Z	— — · ·

Készítsünk programot `i547` néven, amely morzekóddal megadott üzenetet dekódol és a képernyőn megjelenít. Az üzenet legfeljebb 1000 jelből áll.

A jeleket, betűket, szavakat és mondatokat megadott számú szóközök választják el:

Jel	**Kód**
Rövid jel	· (pont)
Hosszú jel	- (kötőjel)
Jelköz	_ (1 szóköz)
Betűköz	___ (3 szóköz)
Szó és mondatköz	_____ (7 szóköz)

A program olvassa be a standard input egyetlen sorából az üzenet morzekódját. Írja a standard output egyetlen sorába a dekódolt, megfelelően tagolt üzenetet.

Példa a bemenetre (nem tartalmaz sortörést):	**Példa a kimenetre:**
·_·_-___-_-_·___-_·_-_-___·___·_·_·_____·_·_·_-___·_-___-_-_·___-_·_-_-_	`UGYES VAGY`

Beküldendő egy tömörített `i547.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 548 (É). Egy webáruház termékeit a vásárlók az ország különböző pontjain elhelyezkedő csomagátvevő helyeken kapják meg. A vállalat több raktárból szállítja ki a csomagátvevő helyekre a termékeket. A csomagok eljuttatását a raktárból a csomagátvevő helyekre kisebb szállítóvállalatok segítségével oldják meg.

A `szallit.txt` állományban rendelkezésre állnak az elmúlt hónapban történt kiszállítások adatai: a raktár és a csomagátvevő hely sorszáma, a szállító cég neve és a szállítási díj. Válaszoljunk a kérdésekre és oldjuk meg a feladatokat táblázatkezelő program segítségével. A feladat mintája a borítón található.

1. Töltsük be a tabulátorokkal tagolt, UTF-8 kódolású `szallit.txt` szövegfájlt a táblázatkezelő egy munkalapjára az A1-es cellától kezdődően. Mentsük a táblázatot i548 néven a táblázatkezelő alapértelmezett formátumában.
2. Gyűjtsük ki a szállítók nevét a G oszlopban a mintának megfelelően. Minden szállító pontosan egyszer szerepeljen a táblázatrészben.
3. Adjuk meg a H oszlopban a szállítók nevei mellett, hogy hány alkalommal végeztek szállítást az elmúlt hónapban. Egy másolható képlettel dolgozzunk minden szállító esetében.
4. Adjuk meg az I oszlopban, hogy az egyes szállítóknak összesen mekkora díjat kellett fizetni a szállítások után.
5. Hozzunk létre az F8:K23 cellák fölött és formázzuk a minta szerint a szállítók teljes díját szemléltető diagramot. A diagram ne rendelkezzen jelmagyarázattal, de adjuk meg minden körcikkhez a cég nevét és az összes díjat tartalmazó feliratot a mintának megfelelően.
6. A G26:J30 tartományban a mintának megfelelően vizsgáljuk meg, hogy egy raktár és egy csomagátvevő hely közötti szállításnál melyik szállító és milyen szállítási díjért vállalja legolcsóbban a termékek fuvarozását. A táblázatrészt a mintának megfelelően formázzuk. Amennyiben a G27 és I27 cellákban megadjuk egy raktár és egy szállító sorszámát, akkor a G30 cellában jelenjen meg a legkisebb ár, vagy „---” három vonal, illetve az I30-as cellában jelenjen meg a szállító neve, vagy a „--- nincs ---” felirat.
7. Alakítsuk ki a mintán látható elrendezést. A feladat megoldása során segédcellákat használhatunk az N oszloptól jobbra.

A feladathoz tartozó minta megtalálható a borítón.

Beküldendő egy tömörített `i548.zip` állományban a megoldást adó munkafüzet és egy rövid dokumentáció, amely leírja, hogy a munkafüzet melyik program segítségével készült.

I. 549. A Vigenère-féle kódolásról októberi számunkban, a visszafejtés rejtelmeiről pedig mostani számunkban olvashatunk egy-egy cikkben. A feladat ezek alapján a kódolást és visszafejtést végző programok elkészítése lesz.

Készítsünk programot `vcode` néven, amely a bemenet első argumentumaként megadott szöveges állományt kódolja a második argumentumként megadott kulcsszó segítségével, és az eredményt egy szöveges állományba írja. A kimeneti állomány csak a bemeneti állomány magyar ABC szerinti betűinek kódját tartalmazza, tehát

a szóközöket és írásjeleket hagyjuk el. A könnyebb olvashatóság kedvéért a sortörések maradjanak meg. A kimeneti állomány nevét egy „`_vc`” szórészlettel egészítsük ki, az állomány kiterjesztése ne változzon.

Példa: a `vcode vers.txt KÖMALINFORMATIKA` parancs futtatása esetén jöjjön létre a `vers_vc.txt` szöveges állomány.

a `vers.txt` első négy sora	a `vers_vc.txt` első négy sora
Titkos üzenet száll a széllel, Hozzád is elér még az éjjel. Most megáll az ész, úgy száll a képzelet, És megtalállak téged.	DVÉLXAIFREPUMIMMÜŐEAPTWKX RBMANMÚXRCQSÉOQÁKUÜKŐT ŰBEUVNTHXCNAXAKÜQÓEANTWGWVBAWTÖU ŐHWÉRÁOQÖCVÁDÁŐHÖS

Készítsünk programot `vdecode` néven, amely a bemenet első argumentumaként megadott szöveges állományt visszaalakítja a második argumentumként megadott kulcsszó segítségével, és az eredményt egy szöveges állományba írja. A bemeneti állomány a magyar ABC nagybetűit tartalmazza, szóközöket és írásjeleket nem. Az áttekinthetőség érdekében az állomány sorokra tagolt. A kimeneti állomány nevét egy „`_de`” szórészlettel egészítsük ki, az állomány kiterjesztése ne változzon.

Példa: a `vdecode szoveg.txt KÖMALINFORMATIKA` parancs futtatása esetén jöjjön létre a `szoveg_de.txt` szöveges állomány.

a `szoveg.txt` első négy sora	a `szoveg_de.txt` első négy sora
OŰNSŰNÉIÓUÓÁZNLTKPWJEŰTLDLFEGUMOJ ÖÜLÍÜSZWÓZXÁZHÚFAMUTŐTBŐR LSUGQNEKZÚUBEÖŐQŐHNA LÁNMTÁÚPÓZSFJSSULZOMYRO	CHARLESBABBAGEASZÁMÍTÓGÉPTUDOMÁNY EGYIKKORAINAGYKÉPVISELŐJE ADIFFERENCIÁLGÉPÉSAZ ANALITIKAIGÉPKITALÁLÓJA

Beküldendő egy tömörített `i549.zip` állományban a két program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállományok melyik fejlesztői környezetben fordíthatók.

I/S. 57. Adott egy N elemű T tömb, amelynek az i-edik elemét $T[i]$-vel jelöljük ($1 \leqslant i \leqslant N$). A tömb összes eleme pozitív, N-nél nem nagyobb egész szám. A tömb leggyakoribb elemei közé akkor tartozik egy x szám ($1 \leqslant x \leqslant N$), ha nincs olyan másik y szám ($1 \leqslant y \leqslant N$), ami többször fordul elő T-ben, mint x.

Módosításnak nevezzük azt, ha az **eredeti** T tömb egy elemét megváltoztatjuk egy tetszőleges pozitív, N-nél nem nagyobb egész számra. Két módosítás különböző, ha különböző elemét módosítjuk T-nek, vagy ugyanazt az elemét módosítjuk, de más értékre.

Adjuk meg minden x számra ($1 \leqslant x \leqslant N$), hogy hány olyan módosítás van, aminek a végrehajtása után a kapott tömbnek x egy leggyakoribb eleme lesz.

Bemenet: az első sorban az N szám található. A következő sorban N darab szám található: a T tömb elemei.

A kimenet egyetlen sorában adjunk meg N darab számot: az i-edik szám azon módosítások száma legyen, amikor i egy leggyakoribb elem.

Példa:

Bemenet	Kimenet
4 1 1 2 2	6 6 0 0

Magyarázat: az 1 egy leggyakoribb elem lesz, a 3. vagy 4. elemet bármilyen lehetséges értékre módosítjuk. Ez összesen $3+3=6$ eset. Ehhez hasonlóan lehet a 2 egy leggyakoribb elem. A 3 vagy a 4 nem lehet egy leggyakoribb elem.

Korlátok: $1 \leqslant N \leqslant 1000$, $1 \leqslant T[i] \leqslant 1000$. Időlimit: 0,3 mp.

Értékelés: a pontok 50%-a kapható, ha a program az $1 \leqslant N \leqslant 10$ elemszámú tesztesetekre helyes megoldást ad.

Beküldendő egy is57.zip tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

S. 156. Piripócson N utca van, melyek N csomópontban találkoznak. A csomópontok 1-től N-ig számozottak. Minden utca két csomópontot köt össze. Az utcákon két irányban lehet közlekedni és bármely csomópontból el lehet jutni bármelyik másikba. Egy turisztikai cég olyan túrát tervez, mely pontosan K utcán megy keresztül. Minden utcán legfeljebb egyszer haladnak át, és a túrán az egymás után következő utcák egyik végpontja közös.

Készítsünk programot, amely megadja, hogy hányféle túrát szervezhetnek. Két túra csak akkor különböző, ha van olyan utca, amit az egyik túra érint, de a másik nem.

Bemenet: az első sor tartalmazza a csomópontok N számát és a túra K hosszát. A következő N sor mindegyike egy-egy utca két végét írja le.

Kimenet: a kimenet első és egyetlen sorába a K hosszú túrák számát kell kiírni.

Példa:

Bemenet (a / jel sortörést helyettesít)	**Kimenet**
4 2 / 1 2 / 1 3 / 2 3 / 2 4	5

Korlátok: $3 \leqslant N \leqslant 500$, $1 \leqslant K \leqslant 20$. Időlimit: 0,5 mp.

Értékelés: a pontok 50%-a kapható, ha a program az $N \leqslant 50$ elemszámú tesztesetekre helyes megoldást ad.

Beküldendő egy s156.zip tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

❄

A feladatok megoldásai regisztráció után a következő címen tölthetők fel:
https://www.komal.hu/munkafuzet
Beküldési határidő: 2021. december 15.

❄