

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Trzyliterowe napisy – LOGIA 21 (2020/21), etap 3

Treść zadania

Tomek jest harcerzem i otrzymał zadanie utworzenia trzyliterowych napisów na podstawie harcerskiego szyfru ułamkowego. Alfabet złożony z małych liter alfabetu łacińskiego jest podzielony na pięć grup. Każda litera jest kodowana ułamkiem. Mianownik określa grupę, w której znajduje się litera, a licznik pozycję litery w grupie.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	2	3	4	5	6	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1						2					3					4					5				

Na przykład: ułamek $\frac{5}{2}$ to zakodowana litera **k**, ułamek $\frac{4}{3}$ to zakodowana litera **o**, ułamek $\frac{4}{4}$ to zakodowana litera **t**.

Pomóż Tomkowi i napisz program znajdujący wszystkie możliwe trzyliterowe napisy, których suma ułamków tworzących zakodowane litery jest podana na wejściu lub słowo **BRAK** w przypadku, gdy nie ma takich napisów.

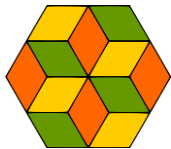
Wejście:

Dwie liczby całkowite z zakresu od **1** do **1000** oddzielone spacją. Pierwsza z nich to licznik, a druga to mianownik ułamka.

Wyjście:

Jedyny wiersz wyjścia zawiera posortowane alfabetycznie wszystkie trzyliterowe napisy oddzielone spacją, których suma ułamków kodujących litery jest równa ułamkowi podanemu na wejściu lub słowo **BRAK**, kiedy nie ma takich napisów.

Przykład 1	Wejście	29 6
	Wyjście	ako aok bio bkl blk boi cgo cil cli cog cor cro dgl dlq dlr drl gco gdl glq goc hko hok ibo icl ijo ilc iob ioj ipp jio jkl jlk joi kao kbl kho kjl klb klj kmp kno koa koh kon kot koz kpm kto kzo lbk lci ldg ldr lqd lic ljk lkb lkj lrd mlp mpk nko nok oak obi ocl ocr ogc ohk oib oij oji oka okh okn okt okz onk orc otk ozk pip pkm pmk ppi rco rdl rld roc tko tok zko zok
Przykład 2	Wejście	3 3
	Wyjście	ggq llq qgg qqr qrr rqq vvx vww vxv www wvv xvq
Przykład 3	Wejście	79 5
	Wyjście	BRAK



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Omówienie rozwiązania

Zastanówmy się jak rozwiązać ten problem. Można wygenerować wszystkie trzyliterowe napisy składające się z małych liter alfabetu łacińskiego i dla każdego z nich sprawdzić, czy po zastosowaniu harcerskiego szyfru ułamkowego suma ułamków tworzących zakodowane litery jest taka, jak podana na wejściu. Na wyjściu wypisane napisy powinny być w kolejności alfabetycznej, dlatego najlepiej, gdy generujemy trzyliterowe słowa alfabetycznie. Wówczas, gdy dane słowo spełnia warunek zadania, można go od razu wypisać. Jeśli żadne z generowanych słów nie spełnia warunku wtedy wypisujemy słowo BRAK.

Do liczenia sumy ułamków zwykłych przyda się funkcja obliczająca największy wspólny dzielnik (**nwd**) dwóch podanych liczb oraz funkcja obliczająca najmniejszą wspólną wielokrotność (**nww**) dwóch liczb. Pierwsza z nich jest realizowana za pomocą algorytmu Euklidesa.

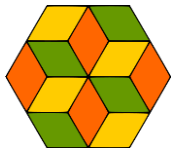
```
1. def nwd(a, b):
2.     while b != 0:
3.         pom = b
4.         b = a % b
5.         a = pom
6.     return a
7.
8. def nww(a, b):
9.     return a * b // nwd(a, b)
```

Funkcja **dodaj** ma 6 parametrów, są to kolejno licznik i mianownik pierwszego ułamka, licznik i mianownik drugiego ułamka, wreszcie licznik i mianownik trzeciego ułamka. Wynikiem tej funkcji jest lista składająca się z dwóch liczb – licznika i mianownika ułamka będącego sumą trzech wejściowych ułamków. Wynikowy ułamek jest w postaci skróconej.

```
1. def dodaj(l1, m1, l2, m2, l3, m3):
2.     m = nww(nww(m1, m2), m3)
3.     l1 = l1 * m // m1
4.     l2 = l2 * m // m2
5.     l3 = l3 * m // m3
6.     licznik = l1 + l2 + l3
7.     pom = nwd(licznik, m)
8.     licznik = licznik // pom
9.     m = m // pom
10.    return [licznik] + [m]
```

Funkcja **szyfr** dla danej małej litery alfabetu łacińskiego zwraca w wyniku listę, która składa się z dwóch liczb – licznika i mianownika ułamka, jaki powstał w wyniku zaszyfrowania tej litery.

```
1. def szyfr(zn):
2.     alfabet = ['abcdef', 'ghijk', 'lmnop', 'qrstu', 'vwxyz']
3.     for i in range(5):
4.         if zn in alfabet[i]:
5.             d = i + 1
6.             for j in range(len(alfabet[i])):
7.                 if zn == alfabet[i][j]:
8.                     g = j + 1
9.     return [g] + [d]
```

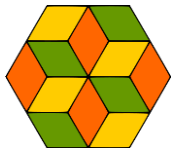


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

Poniższy program generuje alfabetycznie wszystkie trzyliterowe napisy i sprawdza, czy suma ułamków będących kodem każdej z liter jest równa ułamkowi na wejściu. Korzysta z funkcji **szyfr** do szyfrowania każdej z liter w postaci ułamka oraz funkcji **dodaj** do dodania tych ułamków.

```
1. def nwd(a, b):
2.     while b != 0:
3.         pom = b; b = a % b; a = pom
4.     return a
5.
6. def nww(a, b):
7.     return a * b // nwd(a, b)
8.
9. def dodaj(l1, m1, l2, m2, l3, m3):
10.    m = nww(nww(m1, m2), m3)
11.    l1 = l1 * m // m1
12.    l2 = l2 * m // m2
13.    l3 = l3 * m // m3
14.    licznik = l1 + l2 + l3
15.    pom = nwd(licznik, m)
16.    licznik = licznik // pom
17.    m = m // pom
18.    return [licznik] + [m]
19.
20. def szyfr(zn):
21.    alfabet = ['abcdef', 'ghijk', 'lmnop', 'qrstu', 'vwxyz']
22.    for i in range(5):
23.        if zn in alfabet[i]:
24.            d = i + 1
25.            for j in range(len(alfabet[i])):
26.                if zn == alfabet[i][j]:
27.                    g = j + 1
28.    return [g] + [d]
29.
30. dane = input().split()
31. lista = [int(x) for x in dane]
32. # skracanie ułamka wejściowego
33. pom = nwd(lista[0], lista[1])
34. lista[0] //= pom
35. lista[1] //= pom
36.
37. w = False
38. alfabet = 'abcdefghijklmnopqrstuvwxyz'
39. for i in range(len(alfabet)):
40.    s1 = alfabet[i]
41.    u1 = szyfr(s1)
42.    for j in range(len(alfabet)):
43.        s2 = alfabet[j]
44.        u2 = szyfr(s2)
45.        for i in range(len(alfabet)):
46.            s3 = alfabet[i]
47.            u3 = szyfr(s3)
48.            tab = dodaj(u1[0], u1[1], u2[0], u2[1], u3[0], u3[1])
49.            if tab == lista:
50.                print(s1 + s2 + s3, end=' ')
51.                w = True
52. if not w:
53.     print("BRAK")
```

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

wejście → 119 30

wyjście → ipy iyp kmy kym mky myk piy pyi yip ykm ymk ypi

wejście → 238 60

wyjście → ipy iyp kmy kym mky myk piy pyi yip ykm ymk ypi

wejście → 23 12

wyjście → amq aqm gms gsm hmq hqm llu loq lqo lul maq mgs mhq mnq mqa mqh mqn mqt mqz
mrs msg msr mtq mzq nmq nqm olq oql qam qhm qlo qma qmh qmn qmt qmz qnm qol qtm qzm
rms rsm sgm smg smr srm tmq tqm ull zmq zqm

wejście → 888 222

wyjście → aab aaj aba abh abn abt abz agk ahb ahj aii aja ajh ajn ajt ajz akq akr anb anj aop apo ark
atb atj azb azj baa bah ban bat baz bgi bha bhh bhn bht bhz big bir blp bmo bna bnh bnn bnt bnz
bom bpl bri bsu bta bth btn btt btz bus bza bzh bzn bzt bzz cgg cgr clm cml cqs crg crr csq cvy cwx
cxw cyv gak gbi gcg gcr ggc ghk gib gij gji gka gkh gkn gkt gkz gnk grc gtk gzk hab haj hba hbh hbn
hbt hbz hkg hhb hhj hii hja hjh hjn hjt hjz hkg hkr hnb hnj hop hpo hrk htb htj hzb hzj iai ibg ibr igb
igj ihi iia iih iin iit iiz ijj ijr ini irb irj iti iuu izi jaa jah jan jat jaz jgi jha jhh jhn jht jhz jig jir jlp jmo jna
jnh jnn jnt jnz jom jpl jri jsu jta jth jtn jtt jtz jus jza jzh jzn jzt jzz kag kar kga kgh kgn kgt kgz khg khr
kng knr kqu kra krh krn krt krz kss ktg ktr kuq kzg kzt lbp lcn ljp lmc lpb lpj mbo mcl mjo mlc mob
moj mpp nab naj nba nbh nbn nbt nbz ngk nhb nhj nii nja njh njn njt njz nkg nkr nnb nnj nop npo
nrk ntb ntj nzb nzj oap obm ohp ojm omb omj onp ooo opa oph opn opt opz otp ozp pao pbl pho
pjl plb plj pmp pno poa poh pon pot poz ppm pto pzo qcs qku qsc quk rak rbi rcg rcr rgc rhk rib rij
rji rka rkh rkn rkt rkz rnk rrc rtk rzk sbu scq sju sks sqc ssk sub suj tab taj tba tbh tbn tbt tbz tgc thb
thj tii tja tjh tjn tjt tjz tkg tkr tnb tnj top tpo trk ttb ttj tzb tzj uhs uiu ujs ukq uqk usb usj uui vcy vyc
wxc wxc xcw xwc ycv yvc zab zaj zba zbh zbn zbt zbz zgk zhb zhj zii zja zjh zjn zjt zjz zkg zkr znb znj
zop zpo zrk ztb ztj zzb zzj