



PRAKTIKUM IZ PROGRAMIRANJA 3

VEŽBE 2

Lazar Vasović, Luka Marković,
Andreja Živić, Jovan Janićijević



Napisati funkciju koja vraća listu prvih 100 Fibonačijevih brojeva. Nakon toga filtrirati listu tako da sadrži samo parne elemente.

Izlaz:

[0, 2, 8, 34, 144, 610, ..., 51680708854858323072,
218922995834555169026]



```
def fibonacci():  
    l = list()  
  
    l.append(0)  
    l.append(1)  
  
    for i in range(2, 100):  
        l.append(l[i - 2] + l[i - 1])  
  
    return l  
  
lista = fibonacci()  
  
#Filtriranje - prvi nacin  
novaLista = [x for x in lista if x % 2 == 0]  
  
print(novaLista)
```



```
def paranBroj(x):  
    if(x % 2 == 0):  
        return True  
    return False  
  
lista = fibonacci()  
  
#Filtriranje - drugi nacin  
novaLista = list(filter(paranBroj, lista))  
  
print(novaLista)
```

Napisati program koja učitava dve liste proizvoljne dužine, zatim od njih formira skupove tako da se svaki element liste pojavljuje tačno jednom u skupu. Štampati dobijene skupove. Nakon toga uraditi uniju, presek, razliku i simetričnu razliku ta dva skupa i ispisati dobijen rezultat.

Ulaz:

1 2 3 4 4 3 2 1

7 6 5 4 3 2 7 6 8 9

Izlaz:

{1, 2, 3, 4}

{2, 3, 4, 5, 6, 7, 8, 9}

Presek: {2, 3, 4}

Unija: {1, 2, 3, 4, 5, 6, 7, 8, 9}

Razlika: {1}

Simetricna razlika: {1, 5, 6, 7, 8, 9}



```
lista1 = list(map(int, input().split()))
lista2 = list(map(int, input().split()))

skup1 = set(lista1)
skup2 = set(lista2)

print(skup1)
print(skup2)

print('Presek: ', skup1 & skup2)
print('Unija: ', skup1 | skup2)
print('Razlika: ', skup1 - skup2)
print('Simetricna razlika: ', skup1 ^ skup2)
```



Napisati program koja učitava listu brojeva i ispisuje koliko se puta svaki element te liste pojavljuje u njoj. Zadatak rešiti korišćenjem skupova.

Ulaz:

7 6 5 4 3 4 7 2 2 7 6 8

Izlaz:

Element 2 se u listi pojavljuje 2 puta

Element 3 se u listi pojavljuje 1 puta

Element 4 se u listi pojavljuje 2 puta

Element 5 se u listi pojavljuje 1 puta

Element 6 se u listi pojavljuje 2 puta

Element 7 se u listi pojavljuje 3 puta

Element 8 se u listi pojavljuje 1 puta



```
lista = list(map(int, input().split()))
```

```
skup = set(lista)
```

```
for el in skup:
```

```
    print('Element ', el, ' se u listi pojavljuje ', lista.count(el),  
        ' puta')
```



Tokom jedne košarkaške sezone neki timovi su se sastajali više puta. Ako je poznat rezultat svakog njihovog susreta, napiši program koji određuje ukupan rezultat svih njihovih susreta nakon te sezone.

Prvo se unosi n (broj mečeva), a zatim u sledećih n redova se unose brojevi p , q , r i s (timovi p i q su odigrali meč sa rezultatom $r : s$). Nakon toga unosi se m (broj parova) i m parova p i q .

Za svaki par timova p i q , u posebnoj liniji ispiši koliko ukupno koševa je tim p dao timu q i koliko je tim q dao timu p , razdvojene dvotačkom.



Ulaz:

5

1 2 15 70

2 3 90 88

1 2 23 9

3 1 88 86

2 3 42 50

2

1 2

3 2

Izlaz:

38:79

138:132



```
n = int(input())
mat = [[0 for i in range(4)] for i in range(n)]

for i in range(n):
    mat[i] = list(map(int, input().split())) # unos vise
    promenljivih u istom redu

m = int(input())
rezultati = []
for i in range(m):
    p, q = list(map(int, input().split()))
    poeni1 = 0
    poeni2 = 0
```



```
for i in range(n):
    if (mat[i][0] == p and mat[i][1] == q):
        poeni1 += mat[i][2]
        poeni2 += mat[i][3]
    elif (mat[i][0] == q and mat[i][1] == p):
        poeni2 += mat[i][2]
        poeni1 += mat[i][3]

    rezultati.append(str(poeni1) + ":" + str(poeni2))

for rez in rezultati:
    print(rez)
```



Za dato n ispisati elemente niza 1, 2, 4, 5, 7, 9, 10, 12, 14, 16, 17, ... koji se formira tako što se polazeći od broja 1 – prikazuje jedan neparan prirodni broj, zatim sledeća dva parna – 2, 4; pa sledeća 3 neparna – 5, 7, 9; sledeća 4 parna – 10, 12, 14, 16, ... itd. Poslednja serija sadrži n elemenata.

Na ulazu se zadaje prirodan broj n .

Ulaz:

5

Izlaz:

1 2 4 5 7 9 10 12 14 16 17 19 21 23 25



```
n = int(input())
br = 1
trenutni = 0

while (br <= n):
    trenutni += 1
    for i in range(br - 1):
        print(trenutni)
        trenutni += 2

    print(trenutni)
    br += 1
```



Milica je kupovala razne proizvode i svaki put kada se vratila iz kupovine dopisivala je sve proizvode koje je kupila na spisak. Napiši program koji pomaže Milici da odredi proizvod koji je najčešće kupovala tokom prethodne godine.

Na ulazu se zadaje n (broj proizvoda), a zatim n proizvoda, svaki u novom redu (string).

Na standardni izlaz ispisati jedan red koji sadrži naziv najčešće kupovanog proizvoda, jedan razmak i broj puta koliko je taj proizvod kupljen. Ako je više proizvoda kupljeno isti broj puta, ispisati onaj koji je prvi po abecednom redosledu.



Ulaz:

9

jabuka

hleb

kruska

jabuka

sljiva

hleb

mleko

jabuka

Hleb

Izlaz:

hleb 3



```
recnik = dict()

n = int(input())
for i in range(n):
    proizvod = input()
    if (recnik.get(proizvod) == None):
        recnik[proizvod] = 1
    else:
        recnik[proizvod] += 1

Max = list(recnik.items())[0]

print("-----")
```



```
for r in recnik.items():
    # prvi uslov je za vrednost, drugi ukoliko je isti broj,
    gledamo leksikografski
    if (r[1] > Max[1]) or (r[1] == Max[1] and r[0] < Max[0]):
        Max = r

print(Max[0] + " " + str(Max[1]))
```

```
# recnik.items() - vraca listu parova (kljuc, vrednost)
# recnik.keys() - vraca listu kljuceva
# recnik.values() - vraca listu vrednosti
# nisu klasicne liste (dict_items, dict_keys, dict_values)
# za indeksiranje, moraju da se kastuju sa list, za iteraciju ne
moraju
```

Katarina je odlučila da svojoj drugarici pošalje šifrovanu poruku, koja sadrži samo slova engleske abecede, cifre i interpunkcijske znake. Svako slovo će šifrovati posebno na osnovu narednih pravila. Mala slova se šifruju velikim slovima tako što se slovo a šifruje slovom Z, slovo b šifruje slovom Y, c slovom X itd., sve do slova y koje se šifruje slovom B i z koje se šifruje slovom A. Velika slova se šifruju potpuno analogno - od A koje se šifruje sa z do Z koje se šifruje sa a. Ostali karakteri se ne menjaju.

Sa standardnog ulaza unosi **se jedna linija teksta**.

Na standardni izlaz ispisati **šifrovani tekst**.



Ulaz:

Zdravo svima.

Izlaz:

aWIZEL HERNZ



```
def pretvoriKarakter(c):  
    # ord(c) - dobijanje ASCII vrednosti od karaktera  
    if (c >= 'a' and c <= 'z'):  
        razdaljina = ord(c)-ord('a')  
        noviKar = ord('Z')-razdaljina  
        return chr(noviKar)  
    elif (c >= "A" and c <= "Z"):  
        razdaljina = ord(c)-ord('A')  
        noviKar = ord('z')-razdaljina  
        return chr(noviKar)  
  
    return c # ako nije veliko ili malo slovo samo vraca karakter  
  
s = input()  
for slovo in s:  
    novoSlovo = pretvoriKarakter(slovo)  
    print(novoSlovo, end='')
```