

3 a) Scrieți o funcție `cifra_control` care primește ca parametru un număr natural `n` și returnează cifra de control a numărului `n`. Cifra de control a unui număr natural se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră.

```
112 def cifra_control(n):  
113       
114     if n<=9:  
115         return n  
116       
117     sum = 0  
118     for i in str(n):  
119         sum += int(i)  
120       
121     return cifra_control(sum)
```

b) Scrieți o funcție `insereaza_cifra_control` care primește ca parametru o listă de numere naturale și o modifică inserând după fiecare element din listă cifra sa de control (!funcția modifică lista primită ca parametru, nu returnează lista modificată). Cifra de control a unui număr se va calcula folosind funcția de la a)

```
125 def insereaza_cifra_control(l):  
126     #global l  
127     for i in range(len(l)):  
128         l.insert(2*i+1,cifra_control(l[2*i]))  
129
```

c) Scrieți o funcție `egale` care primește ca parametri un număr variabil de liste cu elemente numere naturale și întoarce `True` dacă listele primite ca parametru sunt egale și `False` altfel.

```

139 def sunt_egale(*args):
140
141     for l in args:
142         if l!=args[0]:
143             return False
144     return True
145 print(sunt_egale([1,2],[1,2],[1,2],[1,2]))

```

d) Se consideră fișierul numere.in care conține numere naturale separate prin spații. Folosind funcția de la b) afișați pe ecran fiecare număr din fișier urmat de cifra sa de control. După fiecare cifră de control afișată se va trece la linie nouă

numere.in	Iesire pe ecran
10 78 8051 91	10 1 78 6 8051 5 91 1

```

147 fin = open("numere.in","r")
148
149 l = [int(x) for x in fin.read().split()]
150
151 insereaza_cifra_control(l)
152
153 for i in range(0,len(l),2):
154     print(l[i],l[i+1],sep=" ")
155
156 fin.close()

```

e) Se dau două fișiere numere.in și numere2.in conținând numere naturale separate prin spații. Asociem fiecărui fișier un șir de cifre de control astfel: considerăm numerele distincte din fișier în ordine crescătoare și formăm un șir cu cifrele lor de control (în această ordine). Spunem că cele două fișiere sunt similare dacă șirurile asociate celor două fișiere sunt egale. Folosind funcțiile de la punctele b și c să se determine dacă cele două fișiere sunt similare și să se afișeze pe ecran un mesaj corespunzător da/nu

numere.in	numere2.in
100 10 78 8051 10 110	51 100 200 788 100 19 51

Numerele distincte din primul fișier în ordine crescătoare sunt: 10, 78, 100, 110, 8051

Șirul cifrelor de control asociat primului fișier va fi atunci: 1, 6, 1, 2, 5

Numerele distincte din al doilea fișier în ordine crescătoare sunt: 19, 51, 100, 200, 788

Șirul cifrelor de control asociat primului fișier va fi atunci: 1, 6, 1, 2, 5

Cele două șiruri sunt egale, deci se va afișa da.

```

158 fin = open("numere.in", "r")
159 l1 = [int(x) for x in fin.read().split()]
160 fin.close()
161
162 fin = open("numere2.in", "r")
163 l2 = [int(x) for x in fin.read().split()]
164 fin.close()
165
166 set1 = set(l1)
167 set2 = set(l2)
168
169 #print(set1, set2)
170
171 l1 = list(set1)
172 l2 = list(set2)
173
174 l1.sort()
175 l2.sort()
176
177 print(l1, l2)
178
179 cif1 = [cifra_control(x) for x in l1]
180 cif2 = [cifra_control(x) for x in l2]
181
182 if sunt_egale(cif1, cif2):
183     print("Da")
184 else:
185     print("Nu")

```

4) Spiridușii lui Moș Crăciun au fiecare asociat câte un cod format din litere și cifre. Până în luna noiembrie spiridușii au avut la dispoziție un fișier în care să completeze ce jucării pot face până la Crăciun și câte bucăți. O linie din acest fișier conține codul spiridușului, numărul de bucăți (număr

natural) și numele jucăriei (numele este format din cuvinte separate prin câte un spațiu). Un spiriduș poate adăuga de mai multe ori o linie în fișier, chiar și cu aceeași jucărie, dacă se hotărăște că poate face mai multe.

Un exemplu de fișier este: S1 1 papusa S2 1 papusa S3 1 masinuta S1 10 trenulet S2 1 papusa S2 2 masinuta S1 10 ponei S3 15 ponei

a) Memorați datele din fișier astfel încât Moș Crăciun să poată afla cât mai repede informațiile cerute la punctele următoare.

```
17 f = open("input.txt", "r")
18 dict = {}
19 for linie in f.readlines():
20     cuv = [cuvinte for cuvinte in linie.split()]
21     if cuv[0] not in dict:
22         dict[cuv[0]] = {}
23     if cuv[2] not in dict[cuv[0]]:
24         dict[cuv[0]][cuv[2]] = int(cuv[1])
25     else:
26         dict[cuv[0]][cuv[2]] = dict[cuv[0]][cuv[2]] + int(cuv[1])
27 #print(dict)
```

b) Dat codul unui spiriduș, care sunt jucăriile pe care le poate face și ce cantitate din fiecare? Pentru aceasta scrieți o funcție `despre_spiridusi` cu 2 parametri: în primul parametru se transmite structura în care s-au memorat datele la punctul a) iar al doilea este codul unui spiriduș. Funcția returnează o lista cu elementele tupluri de 2 elemente – primul fiind numele jucăriei, iar al doilea cantitatea – ordonată descrescător după cantitate și, în caz de egalitate, crescător după nume. Apelați funcția pentru codul S1 și afișați lista returnată de funcție. Pentru datele din fișierul exemplu se va afișa [('ponei', 10), ('trenulet', 10), ('papusa', 1)]

```
39 def criteriu(e):
40     return -e[1], e[0]
41 def despre_spiridusi(dict, cod):
42     list = []
43     for jucarie in dict[cod]:
44         list.append((jucarie, dict[cod][jucarie]))
45     list.sort(key = criteriu) #Varianta 1
46     #return sorted(list, key = lambda x: (-x[1], x[0])) #Varianta 2
47     return list
48 #print(despre_spiridusi(dict, 'S1'))
```

c) Care este mulțimea jucăriilor pe care spiridușii le pot produce? Pentru aceasta scrieți o funcție `jucarii` care primește ca parametru structura în care s-au memorat datele la punctul a) și returnează o mulțime

cu numele jucăriilor care pot fi produse de spiriduși. Apelați funcția și afișați pe ecran elementele mulțimii returnate (pe o linie, separate prin virgula). Pentru datele din fișierul exemplu o posibilă ieșire este (jucăriile se pot afișa în orice ordine): ponei,trenulet,masinuta,papusa

```
57 def jucarii(dict):
58     list = []
59     for spiridus in dict:
60         for jucarie in dict[spiridus]:
61             if jucarie not in list:
62                 list.append(jucarie)
63     return list
64 #print(jucarii(dict))
65
```

d) Care este lista spiridușilor harnici: ordonați descrescător după numărul de jucării diferite pe care le pot face și, în caz de egalitate, descrescător după cantitatea de jucării pe care o vor produce și, în caz de egalitate, crescător după cod? Pentru aceasta scrieți o funcție spiridusi care primește ca parametru structura în care s-au memorat datele și returnează o lista cu elementele tupluri de 3 elemente – primul fiind codul spiridușului, al doilea numărul de jucării diferite pe care le poate produce, iar al treilea numărul total de bucăți de jucării care le poate produce spiridușul – ordonată după criteriile cerute de Moș Crăciun (precizate anterior). Apelați funcția și afișați pe ecran elementele listei obținute, fiecare tuplu din listă fiind afișat pe o linie separată. Pentru datele din fișierul exemplu se va afișa

('S1', 3, 21)

('S3', 2, 16)

('S2', 2, 4)

```
78 def spiridusi(dict):
79     lista = []
80     for spiridus in dict:
81         nr_tip_juc = len(dict[spiridus])
82         nr_juc = sum([dict[spiridus][x] for x in dict[spiridus]])
83         lista.append((spiridus, nr_tip_juc, nr_juc))
84     return sorted(lista, key=lambda e: (-e[1], -e[2], e[0]))
85 #for x in spiridusi(dict):
86 #    print(x)
```

e) În caz că un spiriduș nu mai poate produce un tip de jucărie, să se actualizeze informațiile asociate spiridușului. Pentru aceasta scrieți o funcție actualizare care primește 3 parametri (în aceasta ordine): structura în care s-au memorat datele, codul spiridușului și numele jucăriei. Dacă spiridușul dat ca parametru produce cel puțin două tipuri de jucării, funcția va șterge din structura de date informațiile despre jucăria cu numele dat ca parametru atașate spiridușului cu codul dat ca parametru și va returna True. Altfel funcția va returna False . Să se apeleze

funcția pentru spiridușul cu cod S1 și jucăria trenuleț, apoi să se afișeze informațiile rămase despre S1 folosind funcția despre\_spiridusi de la punctul b. Pentru datele din fișierul exemplu se va afișa [('ponei', 10), ('papusa', 1)]

```
96     def actualizare(dict, cod, juc):
97         if len(dict[cod]) >= 2:
98             dict[cod].pop(juc)
99             return True
100        else:
101            return False
102        #print(despre_spiridusi(dict, 'S1'))
103        actualizare(dict, 'S1', 'trenuleț')
104        print(despre_spiridusi(dict, 'S1'))
105        f.close()
```