Subiecte examen
(anii anteriori)

Soutiile vor fi incarcate colorate cu rosu (inseamna ca deocamdata nu au fost verificate de profesor).
Solutiile verificate si care sunt corecte, dar pot fi imbunatatite vor fi colorate de catre profesor cu verde.
O posibila solutie optima (din punct de vedere al modului de codare - cod compact, etc.) va fi colorata cu albastru.
Greselile vor fi marcate cu portocaliu.

Observație: Vor fi adaugate subiecte saptamanal, pe masura ce acestea se pot rezolva cu noțiunile predate la curs. Puteti scrie variante de rezolvari (corectam si completam impreuna).

1. Fara a folosi structuri repetitive, determinati numarul de valori de 1 din scrierea binara a unui numar citit de la tastatura.

   Exemplu: pentru 147 se va afișa 4.

```
print(sum(map(lambda x : 1 if (x == '1') else 0, list(bin(int(input()))[2::]))))
print(list(bin(int(input()))).count("1"))
```

```
n=int(input())
l=list(bin(n))
print(l.count("1"))
```

2. Fara a folosi structuri repetitive, afisati o singură dată, în ordine lexicografică, vocalele folosite intr-un text citit din fisierul *text.txt*.

   Input:

   Fisierul *text.txt* conține

   Ana are mere.

Output:

A, a, e

```
print(sorted(set("Ana are mere.")  & set("aeiouAEIOU")))
```

```
print(*sorted(set("Ana are mere.")  & set("aeiouAEIOU")), sep = ", ")
```

```
f = open("date.txt", "r")
```

```
l=set(f.read())
```

```
setv={'a','e','i','o','u','A','E','I','O','U'}
```

```
voc=l.intersection(setv)
```

```
print(*sorted(voc),sep=", ")
```

3. Se citește de la tastatură un număr natural *n*, apoi *n* numere întregi ce se vor memora într-o colecție A. Se citește de la tastatură un număr natural *m*, apoi *m* întregi ce se vor memora într-o colecție B.

Ex.: n=6, A: 113, -4, 7, -4, -5, -109 și m=7, B: 120, 178, -4, -4, -4, 120, -109

Să se afișeze o singură dată elementele negative din A care apar și în B. *Ex.: pentru colecțiile de mai sus, se va afișa șirul -4, -109.*

```
n = int(input("n = "))
A = set()
print("A: ")
for i in range(0, n):
    A.add(int(input()))
m = int(input("m = "))
B = set()
print("B: ")
for i in range(0, m):
    B.add(int(input()))
print(list(filter(lambda x: x < 0, list(A & B))))    # print(list(filter(lambda x: x < 0, A & B)))
```

da eroare

Scrie aici setul de date pentru care da eroare (eventual print screen)

Cum am citi din fisier aici?

R.:

Daca date.txt contine:

6 113 -4 7 -4 -5 -109 7 120 178 -4 -4 -4 120 -109

atunci

```
f = open("date.txt")
l_numere = list(map(int, f.read().split()))
n = l_numere[0]
A = l_numere[1 : n + 1]
m = l_numere[n + 1]
B = l_numere[n + 2: ]
A = set(A); B = set(B)
```

```
n=int(input("n="))
A=[]
B=[]
for i in range(n):
        A.append(int(input()))
m=int(input("m="))
for j in range(m):
        B.append(int(input()))
AA=set(A)
BB=set(B)
la_fel=AA.intersection(BB)
negative=list(la_fel)
for k in negative:
```

```
if k<0:

print(k)
```

nu merge # Da-mi exemplu de set de date de intrare pentru care nu iti merge codul.

4. Se da un text literar in limba romana, fara diacritice, care respecta toate regulile ortografice ale limbii romane (fara spatii nepotrivite, semne de punctuatie gresite, etc). Să se afișeze in consola, pe o singura linie, separate prin virgule, cuvintele din text în ordinea crescătoare a numărului de caractere, iar pentru cele de aceleași lungime, ordonarea se va face alfabetic (vom considera majusculele înaintea literelor mici – conform ordinii codurilor ASCII).

Exemplu:

Input:

s = "Astazi este sambata ! Avem examen la PP... Tu ai invatat? Ai lucrat laboratoarele? . . ."

Output:

Ai, PP, Tu, ai, la, Avem, este, Astazi, examen, lucrat, invatat, sambata, laboratoarele

---

```
import re

import functools as ft

s = input("s = ")

print(', '.join(sorted(list(filter(lambda x: x != '', re.split('[\W*]', s))), key=ft.cmp_to_key(lambda a, b: len(a) - len(b) if len(a) - len(b) != 0 else -1 if a <= b else 1))))

'''
```

SAU:

```
print(sorted(list(filter(lambda x: x != '', re.split('[\W*]', s))), key=ft.cmp_to_key(lambda a, b: len(a) - len(b) if len(a) - len(b) != 0 else -1 if a <= b else 1)), sep = ", ")

'''
```

```
#varianta 2

import re

s=input("s= ")
```

```
print(sorted(re.findall('\w+',s), key=lambda x: (len(x), x)))    # nu va detecta corect cuvintele cu cratima

# DAR:

# print(sorted(re.findall('[\w-]+',s), key=lambda x: (len(x), x)))
```

```
import re

import functools


def crit(x,y):
    if len(x)<len(y):
        return -1
    elif len(x)>len(y):
        return 1
    else:
        if x<y:
            return -1
        else:
            return 1


text=input()

text=re.sub('[\.\?\!\,]','', text)

text=re.split(' ',text)

'''

# SAU:

text = re.findall('[\w-]+', text)

'''

text.sort(key=functools.cmp_to_key(crit))

print(*text)   # nu avem virgula intre cuvinte
```

```
# DAR:

# print(*text, sep = ", ")
```

5. Fie o listă ce conține liste sau tupluri încuibate (ex.: l= [1, [2,3], [4, 5, (6, (7, 8, 9))], 10]. Să se transforme într-o listă simplă, fără a folosi liste suplimentare, păstrând ordinea elementelor.

```
def is_not_flat(l):
    sw = False
    for i in l:
        if(type(i) == list or type(i) == tuple):
            sw = True
            break
    return sw


def type_check(i):
    return type(i) != tuple and type(i) != list


l = [1, [2,3], [4, 5, (6, (7, 8, 9))], 10]


while is_not_flat(l):
      l = [y for x in l for y in ([x] if type_check(x) else x)]
  # aici ai folosit o lista suplimentara:   verifica id-ul listei
print(l)
```

```
# SOLUTIA NU RESPECTA CERINTA: "fără a folosi liste suplimentare"

lista_noua=[]

def dezincuibare(l):
    for i in l:
```

```
        if type(i)==int:
            lista_noua.append(i)
        else:
            dezincuibare(i)
    return lista_noua


l=[1, [2,3], [4, 5, (6, (7, 8, 9))], 10]
print(dezincuibare(l))


# O varianta posibila:
ok = 0
while ok == 0:
    ok = 1
    for el in l:
        if type(el) in [list, tuple]:
            ok = 0
            i = l.index(el)
            l = l[:i] + list(el) + l[i + 1 : ]
print(l)
```

6. Intr-o companie, in ultima vineri din fiecare luna se tine o mica petrecere, pentru cei care sunt nascuti in respectiva luna. Din fisierul *date.in* se citesc, de pe cate o linie, numele unei persoane si data nasterii, in formatul *zz.ll.aaaa* (ca in exemplu).

In fisierul *date.out* sa se afiseze pentru fiecare luna in parte data la care are loc petrecerea si numele sarbatoritilor (ca in exemplu).

Exemplu

INPUT

date.in

Popa Daniela 10.03.1986

Ion Laurentiu 18.07.1990

Tatu Elena 11.03.1986

Grigore Denisa 27.08.2001

Georgescu Florin 18.07.2000

Dinu Andrei 10.03.1997

Pop Floarea 20.04.1970

OUTPUT

- pentru anul 2021 se va afisa:

26 martie 2021: Popa Daniela, Tatu Elena, Dinu Andrei

30 aprilie 2021: Pop Floarea

30 iulie 2021: Ion Laurentiu, Georgescu Florin

27 august 2021: Grigore Denisa

```
import datetime

import calendar

import collections


f = open('date4.txt', 'r' )

t= f.readline()

d = {}

# SAU    d = collections.defaultdict(set)

l_n = []

l_luna = []

final = {}

azi = datetime.date.today()

lista_luna = {1:"Ianuarie", 2:"Februarie", 3:"Martie", 4:"Aprilie", 5:"Mai", 6:"Iunie", 7:"Iulie", 8:"August", 9:"Septembrie", 10:"Octombrie", 11:"Noiembrie", 12:"Decembrie"}

zile = {0: 'luni', 1: 'marti', 2: 'miercuri', 3: 'joi', 4:'vineri', 5: 'sambata', 6 : 'duminica'}

while t:
```

```
    l=t.split()
    l2 = l[2].split('.')
    data_nastere = datetime.date(int(l2[2]), int(l2[1]), int(l2[0]))
    nume = l[0]+ " " + l[1]
    l_n.append(str(nume))
    l_luna.append(data_nastere.month)
    data_nastere = data_nastere.replace(year = azi.year)
    for i in range(len(l_luna)):
        if l_luna[i] in d:
            d[l_luna[i]].add(l_n[i])
        else:
            d[l_luna[i]] = {l_n[i]}
# SAU: d[l_luna[i]].add(l_n[i]) , daca d este defaultdict
    t= f.readline()



for i in l_luna:
    for j in range(15,calendar.monthrange(azi.year, i)[1] + 1):
        data = datetime.date(azi.year, i , j )
        if data.weekday() == 4:
            final[i] = j
'''
# DAT FIIND CA IN CURS NU GASITI calendar.monthrange (SI NU VA CER CEVA NEPREDAT) AVETI SOLUTIE ALTERNATIVA:
zile_februarie = 28
if calendar.isleap(azi.year):
 zile_februarie = 29
```

```
ultima_zi_luna = ["luna 0 nu exista", 31, zile_februarie, 31, 30, 31,
30, 31, 31, 30, 31, 30, 31]

for i in l_luna:

   for j in range(15,ultima_zi_luna[i] + 1):

       data = datetime.date(azi.year, i , j )

       if data.weekday() == 4:

           final[i] = j


'''

for i in lista_luna:

   if i in d:

      print(str(final[i]), lista_luna[i], azi.year, ":" , *d[i])
```

```
import time

import datetime

import collections as c

import calendar


d = c.defaultdict(list)

luni = {1:'ianuarie', 2:'februarie', 3:'martie', 4:'aprilie', 5:'mai', 6:'iunie', 7:'iulie', 8:'august',
9:'septembrie', 10:'octombrie', 11:'noiembrie', 12:'decembrie'}

sapt = {1:"luni",2:"marti",3:"miercuri",4:"joi",5:"vineri",6:"sambata",7:"duminica"}

afisare=c.defaultdict(list)

x = time.localtime()

an_curent=x[0]

zile_devineri=c.defaultdict(list)

vinerif=c.defaultdict(list)
```

```
for i in range(1,13):
  if i in [1,3,5,7,8,10,12]:
    for j in range(1,32):
      data_mea=datetime.date(an_curent, i, j)
      if calendar.weekday(an_curent,i,j)==4:
        zile_devineri[i].append(j)
  if i in [4,6,9,11]:
    for j in range(1,31):
      data_mea=datetime.date(an_curent, i, j)
      if calendar.weekday(an_curent,i,j)==4:
        zile_devineri[i].append(j)
  if i==2:
    if calendar.isleap(an_curent)==0:
      for j in range(1,29):
        data_mea=datetime.date(an_curent, i, j)
        if calendar.weekday(an_curent,i,j)==4:
          zile_devineri[i].append(j)
    else:
      for j in range(1,30):
        data_mea=datetime.date(an_curent, i, j)
        if calendar.weekday(an_curent,i,j)==4:
          zile_devineri[i].append(j)
'''
# SAU:


# PENTRU CA SECVENTA ANETRIOARA (VERDE) CONTINE COD DUPLICAT, SE POATE
SCRIE MAI USOR ASTFEL:

zile_februarie = 28
```

```
if calendar.isleap(an_curent):

zile_februarie = 29

ultima_zi_luna = ["luna 0 nu exista", 31, zile_februarie, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31]




for i in range(1,13):

   for j in range(1,ultima_zi_luna[i] + 1):

     data_mea=datetime.date(an_curent, i, j)

     if calendar.weekday(an_curent,i,j)==4:

       zile_devineri[i].append(j)

'''



for key, value in zile_devineri.items():

  vinerif[key].append(value[-1])

f=open("date.txt","r")

linie=f.readline()

while linie:

  l= linie.split()

  l1=l[2].split('.')

  nume=l[0]+" "+l[1]+','

  luna=int(l1[1])

  d[luna].append(nume)

  linie=f.readline()

for i in range(1,13):

   if i in d.keys():

       print(*vinerif[i],luni[i],an_curent,":",*d[i])
```

```
"""

# Fie un text literar. Determinati frecvența de apariție a fiecarui cuvânt cu metoda Counter din modulul collections (se va face diferenta intre litere mici/majuscule).

# Ex.: Ana?...Ana are multe, multe mere!... Dar tu,tu cate mere ai?

# Ana 2

# mere 2

# multe 2

# tu 2

# Dar 1

# ai 1

# are 1

# cate 1

# '''

# # metoda 1

# f= open("date.txt", "r")

# s = f.read()

# print(s)

# d = "".maketrans("?!.;,:", "      ")

# cuv = s.translate(d).split()

# print(cuv)

#

#

#

# for cuvant in set(cuv):

#   print(cuvant, ": ", cuv.count(cuvant))

#
```

```
# # met 2

# import collections as c

# d = c.defaultdict(int)

#

#

#

# for cuvant in cuv:

#   d[cuvant] += 1

#

# print(dict(d))

# # met 2

# print(dict(*c.Counter(cuv)), sep = '', '')

#

# # PP

# ""1.  In timpul saptamanii Scoala Altfel, intrarea in Muzeul National al Aviatiei Romane este permisa numai pentru grupuri organizate, cu programare. Tarifele sunt urmatoarele:

# 10 lei: elevi si studenti

# 5 lei: pensionari

# 20 lei: adulti

# copii (care nu sunt elevi): gratuit

# Daca numarul de persoane care platesc bilet este mai mare de 20, atunci se aplica o reducere de 10%. Sa se citeasca din fisierul de intrare date.txt cate persoane sunt intr-un grup, din fiecare categorie in parte, cu formatul dat in exemplu si sa se afiseze suma preturilor biletelor, calculata pentru intregul grup.

# Conventii pentru citire datelor: E (elev), S (student), P (pensionar), A (adult), C (copil).

# Exemplu

# INPUT

# date.txt

# 30 E, 2 S, 3 A, 2 C
```

```
# OUTPUT
# 342 lei
# '''
#
#
# '''
# f = open('date.txt','r')
# x = f.read()
# s=0
# z=0
# x = x.replace(',', ' ')
# x = x.split()
# print(x)
#
# d={'E': 10 , 'S': 10, 'P': 5, 'A': 20, 'C': 0}
#
# for i in range(0,len(x)-1,2):
#   s=s+d[x[i+1]]*int(x[i])
#   if x[i+1]!='C':
#     z=z+int(x[i])
#
# if z>=20:
#   s=s-s/10
# print(s)
# '''
#
# '''
```

```
# 2. Sa se citeasca de la tastatura o data calendaristica zz.ll.aaaa (ce reprezinta data nasterii pentru o persoana). Sa se afiseze ziua din saptamana cand va fi urmatoare aniversare.

#

# 17.12.1980  --> 17.12.2021 --> Friday (ro/en)

# 12.03.1980 --> 12.03.2022 --> Sambata (ro/en)

#

#              1)

# '17.12.1980' -->   data  17 dec 1980

# -->   17   dec  2021  --> weekday

#

#

# '17.02.1980' -->   data  17 feb 1980

# -->   17   feb  2022  -->  weekday

# '''

# '''

# import datetime

# import time

#

# data_nasterii_string = input("Cand te-ai nascut?! ")

# data_nasterii = datetime.datetime.strptime(data_nasterii_string, "%d.%m.%Y")

#

# print(data_nasterii)

#

# acum = datetime.datetime.now()

#

# an_curent = int(acum.strftime("%Y"))

# '''

# '''
```

```
# acum_time = time.localtime()
# an_curent_time = time.strftime("%Y", acum_time)
# print(an_curent_time)
# '''
# '''
# print(an_curent)
# print(type(an_curent))
# '''
# '''
# data_nasterii: 2002-02-02
# an_curent: 2021
#
# aniversare --> 2021-02-02
# '''
# '''
# aniversare = data_nasterii.replace(year=an_curent)
# if aniversare < acum:
#   aniversare = data_nasterii.replace(year=an_curent+1)
#
#
# print(aniversare)
# zi=aniversare.strftime("%A")
# print(zi)
#
#
# zile = {0: 'luni', 2 : 'miercuri'}
# zi = aniversare.weekday()
# print(zile[zi])
```

```
# '''
#
# '''
# a) O data de nastere. Data petrecerii (sambata din sapt resp).
#
# Ex. 05.01.1980 --> 08.01.2022
#     09.01.1980 --> 08.01.2022
#
#
#     17.12.1980 --> 18.12.2021
# '''
#
# '''
# import datetime
# import time
#
# data_nasterii_string = input("Cand te-ai nascut?! ")
# data_nasterii = datetime.datetime.strptime(data_nasterii_string, "%d.%m.%Y")
#
# print(data_nasterii)
#
# acum = datetime.datetime.now()
#
# an_curent = int(acum.strftime("%Y"))
# '''
# '''
# acum_time = time.localtime()
# an_curent_time = time.strftime("%Y", acum_time)
```

```
# print(an_curent_time)
# '''
# '''
# print(an_curent)
# print(type(an_curent))
#
# aniversare = data_nasterii.replace(year=an_curent)
# if aniversare < acum:
#   aniversare = data_nasterii.replace(year=an_curent+1)
#
# #zi = int(aniversare.strftime("%w"))
# zi = aniversare.weekday()
# print(zi)
# ziua = aniversare + datetime.timedelta(days = 5 - zi)
# print(ziua)
# '''
# '''
# 26.12.2002   --> 25.12.2021
#    6              5
# aniversare
#       22.12.2021
#
#
# aniversare + datetime.timedelta(days = 5 - 6)
#
# 31.'12.2002  -->
#    1
#
```

```
# sambata = aniversare .....
# '''
#
# '''
# 6
# Tomescu Elena, 15.12.2000
# Toma Diana, 16.12.1999
# Popescu Dan, 07.12.2000
# Georgescu Mara, 12.12.1998
# Florea Andrei, 12.03.2000
# David George, 08.03.2002
#
# Dragii nostri ....., ....., ..... / Draga .....
#
#   Va / Te invitam la petrecerea organizata pentru voi/tine sambata, ......
#
#   Cu drag,
#   Scoala X
# '''
#
# import datetime
# import collections as c
# dict_oameni = c.defaultdict(list)
#
# f=open("date.txt","r")
# nr = int(f.readline())
# print(nr)
#
```

```
# for linie in range(nr):
#   f2=f.readline()
#   lista = f2.split(", ")
#
#   nume = lista[0]
#   data_nasterii_string = lista[1].strip()
#
#   data_nasterii = datetime.datetime.strptime(data_nasterii_string, "%d.%m.%Y")
#
#   acum = datetime.datetime.now()
#   an_curent = int(acum.strftime("%Y"))
#   saptamana_curenta = int(acum.strftime("%U"))
#
#   aniversare = data_nasterii.replace(year=an_curent)
#   print("---------------")
#   print(data_nasterii_string)
#   print(aniversare)
#   saptamana_aniv = int(aniversare.strftime("%U"))
#
#    if(saptamana_aniv != saptamana_curenta) or (acum.weekday() == 6 and saptamana_aniv ==
saptamana_curenta):
#     if aniversare < acum:
#         aniversare = data_nasterii.replace(year=an_curent+1)
#
#   zi = aniversare.weekday()
#   ziua = aniversare + datetime.timedelta(days = 5 - zi)
#
#   print(ziua)
```

#

'''

6. Intr-o companie, in ultima vineri din fiecare luna se tine o mica petrecere, pentru cei care sunt nascuti in respectiva luna. Din fisierul date.in se citesc, de pe cate o linie, numele unei persoane si data nasterii, in formatul zz.ll.aaaa (ca in exemplu). In fisierul date.out sa se afiseze pentru fiecare luna in parte data la care are loc petrecerea si numele sarbatoritilor (ca in exemplu).

Exemplu

INPUT

Tomescu Elena, 07.12.2000

Toma Diana, 08.12.1999

Popescu Dan, 09.08.2000

Georgescu Mara, 12.12.1998

Florea Andrei, 12.03.2000

David George, 08.03.2002

'''

```
import _strptime
from datetime import date, datetime
from datetime import timedelta
import datetime
from datetime import timezone
from dateutil.relativedelta import FR, relativedelta

from datetime import date
```

```
from datetime import datetime


dictionar = {}

f = open("datein.txt", "r")

nr = int(f.readline())

months = ["Ianuarie", "Februarie", "Martie", "Aprilie", "Mai", "Iunie", "Iulie", "August", "Septembrie",
"Oct",

          "Noiembrie", "Decembrie"]

days = ['Luni', 'Marti', 'Miercuri', 'Joi', 'Vineri', 'Sambata', 'Duminica']

for i in range(0,nr):

    string = f.readline()


    string = string.split()







    lista_data = string[2].split(".")



    today = date.today()

    current_year = date.today().year

    anniversary_og = string[2]



    def functie_aniversare():

        aniversare = date(day=int(lista_data[0]), month=int(lista_data[1]), year=current_year)

        return aniversare
```

```
aniversare = functie_aniversare()




def anniversary_checker(aniversare):
    if aniversare < today and abs(aniversare.day-today.day) > 6: #caz special
:
        aniversare = aniversare.replace(year=current_year + 1)

    return aniversare



x = anniversary_checker(aniversare)

key = x + relativedelta(day=31, weekday=FR(-1))
key = str(key)

if key in dictionar:
    lista = []
    lista.append(string[0])
    lista.append(string[1])
    dictionar[key].append(lista)
else:

    lista = []
    lista.append(string[0])
```

```
        lista.append(string[1])
        dictionar[key] = lista

print(dictionar)

for i in dictionar:
    data = i
    data = data.split("-")
    l = len(dictionar[i])
    print(f"{data[2]} {months[int(data[1])-1]} {data[0]} : {dictionar[i]}")
#afisarea nu i 100% corecta
```

7. Din fisierul *date.in* se citesc, de pe cate un rand, numele si data angajarii pentru fiecare salariat dintr-o companie (ca in exemplu). Sa se afiseze in fisierul *date.out* numele angajatilor care au o vechime mai mare de 5 ani completi (nu se vor face rotunjiri: de exemplu, un angajat cu vechime de 4 ani, 11 luni si 10 zile nu va fi afisat).

Exemplu:

INPUT

date.in

Popa Daniela 10.03.2018

Ion Laurentiu 18.07.2020

Tatu Elena 30.01.2016

Grigore Denisa 27.08.2015

Georgescu Florin 29.01.2016

Dinu Andrei 10.03.2010

Pop Floarea 20.04.2020

```
import datetime
import time
```

```
import calendar
import collections as c


f=open("date.in","r")
g=open("date.out","w")
d = c.defaultdict(list)
linie=f.readline()
nr_ani=5
zile=0
while linie:
 l = linie.split()
 nume=l[0]+" "+l[1]
 data=l[2]
 d[nume].append(data)
 linie=f.readline()
 x=time.localtime()
 an_curent=x[0]
 vechime=an_curent-5
 luna_curenta=x[1]
 zi_curenta=x[2]
 data_curenta=datetime.datetime.now()
while nr_ani:
 if calendar.isleap(an_curent)==0:
   zile=zile+365
 else:
   zile=zile+366
 an_curent=an_curent-1
 nr_ani=nr_ani-1
```

```
for i in d:
 d[i]=datetime.datetime.strptime(*d[i], '%d.%m.%Y')
 zile_necesare=data_curenta+datetime.timedelta(days=-zile)
 if d[i]<=zile_necesare:
   print(i)
```

8. Presupunem că reținem date despre compania M&K. Se citesc din fisierul *angajati.txt* urmatoarele date:

- pe prima linie a fișierului: *n* (număr natural, ce reprezintă numărul de angajați);
- pe linia a doua: numele primului angajat;
- pe linia a treia: salariul primului angajat;
- pe linia a patra: numele celui de-al doilea angajat;
- pe linia a cincea: salariul celui de-al doilea angajat;
- …… idem și pentru ceilalți angajați, până la n.

Să se afișeze în fișierul *angajati.out* toți angajații cu toate numele corectate, ordonate alfabetic. Pentru angajați cu același nume de familie aceștia vor apărea ordonați după salariu, descrescator.

9. Într-un spațiu pentru îngrijirea vârstnicilor există obiceiul ca în prima sambata de după aniversarea unui bătrân să i se facă o petrecere într-o sală de festivități. Personalul trebuie să cunoască câte aniversări sunt la fiecare dată (pentru a ști câte cadouri să pregătească). Se citește din fisierul *date.in* numarul natural *n* apoi, pentru *n* persoane, pe cate o linie, numele și data nașterii, în formatul *zz-ll-aaaa* (ca in exemplu). Afișați în fișierul *date.out* toate datele calendaristice în care vor fi următoarele lor aniversări, împreună cu persoanele aniversate la fiecare dată. Datele afișate vor fi ordonate descrescător după numărul de persoane aniversate la respectivele date. Dacă ziua bătrânului pică în weekend, atunci el va fi sărbătorit în sâmbăta din respectivul weekend.

Exemplu:

INPUT

date.in

7

Popa Daniela 10-03-1931

Ion Laurentiu-Mihai 18-07-1940

Tatu Elena Diana 09-03-1936

Grigore Denisa 27-08-1928

Georgescu Florin 20-07-1928

Dinu Andrei 13-03-1937

Pop Floarea 30-03-1943

OUTPUT (valabil pentru rularea codului in dec. 2021)

12-03-2022 Popa Daniela, Tatu Elena Diana, Dinu Andrei

23-07-2022 Ion Laurentiu-Mihai, Georgescu Florin

27-08-2022 Grigore Denisa

02-04-2022 Pop Floarea

```
import datetime
import collections as c

d = c.defaultdict(list)
f = open("date.txt", "r")
n = int(f.readline())
for i in range(n):
  linie = f.readline()
  l = linie.split()
  nume = l[0]+" "+l[1]+","
  data = l[2].strip()
  data_nastere = datetime.datetime.strptime(data, "%d-%m-%Y")
  azi = datetime.datetime.today()
  an_curent = azi.year
  aniv_an_curent = data_nastere.replace(year = an_curent)
  if aniv_an_curent<azi:
    aniv_an_curent = data_nastere.replace(year = an_curent +1)
  sambata=aniv_an_curent + datetime.timedelta(days =5 - aniv_an_curent.weekday())
  d[sambata].append(nume)
for i in d:
    print(datetime.datetime.date(i),":",*d[i])
```

10. Se citește din fisierul *date.in* numarul natural *n* apoi, pentru *n* persoane, pe cate o linie, numele și CNP-ul. Sa se afiseze in fisierul *date.out* numele persoanelor care se pensioneaza in luna curenta (cea in care are loc executia codului). Se considera ca varsta de pensionare pentru femei este 63 de ani, iar pentru barbati de 65 de ani.

INPUT

7

Popa Daniela 2800808100567

Ion Laurentiu 1560120180567

Tatu Elena 2580112100987

Grigore Denisa 6011218100234

Georgescu Florin 6020818100541

Dinu Andrei 1700911180507

Pop Floarea 2580312133987

OUTPUT

- considerand ca rulam codul azi, 29.01.2021, se vor afisa numele:

Ion Laurentiu, Tatu Elena

11. Se citește din fisierul *date.in* numarul natural *n* apoi, pentru *n* persoane, pe cate o linie, numele și CNP-ul. Sa se afiseze in fisierul *date.out* numele persoanelor care se pensioneaza in anul curent (cel in care are loc executia codului). Se considera ca varsta de pensionare pentru femei este 63 de ani, iar pentru barbati de 65 de ani.

INPUT

7

Popa Daniela 2800808100567

Ion Laurentiu 1560120180567

Tatu Elena 2580112100987

Grigore Denisa 6011218100234

Georgescu Florin 6020818100541

Dinu Andrei 1700911180507

Pop Floarea 2580312133987

OUTPUT

- daca am executa codul azi, 29.01.2021, se vor afisa numele:

Ion Laurentiu, Tatu Elena, Pop Floarea

12. Din fisierul *rezultate.in* se citesc, de pe cate o linie, numele si trei note obtinute de cate un absolvent de liceu la examenul de bacalaureat (ca in exemplu). Sa se afisez in fisierul *rezultate.out* numele absolentilor, impreuna cu notele, ordonate descrescator dupa media celor 3 note (ca in exemplu). Daca exista mai multi absolventi cu aceeasi medie, ei vor fi afisati in ordine alfabetica.

Exemplu:

INPUT

rezultate.in

Vasile Elena 9 10 9.8

Popescu Maria 10 9.2 9.6

Pop Mircea 5.5 8.2 6.7

Radu Laura 8.4 5.5 7.8

Ene Florin 8.9 7.8 5.6

Toma Ilena 5.5 5 5.6

OUTPUT

rezultate.out

Popescu Maria 10 9.2 9.6

Vasile Elena 9 10 9.8

Ene Florin 8.9 7.8 5.6

Radu Laura 8.4 5.5 7.8

Pop Mircea 5.5 8.2 6.7

Toma Ilena 5.5 5 5.6

13. Din fisierul *raport.txt* se citeste un text literar, ce contine litere ale alfabetului englezesc si semnele de punctuatie uzuale. Sa se afiseze propozitiile ordonate descrescator dupa numarul de valori numerice ce apar in text. Se vor folosi ori de cate ori este posibil expresiile regulate. Daca exista mai multe propozitii cu acelasi numar de valori numerice se va afisa mai intai propozitia cu mai multe cuvinte (si valorile numerice se vor numara drept cuvinte).

Exemplu:

INPUT

raport.txt

Anul trecut, din 32 de copii, 18 s-au inscris la cercul de pictura, 13 la cel de dansuri si restul la cercul de carting. Din acestia, pe parcursul anului, 3 au abandonat toate cercurile pe care le frecventau. Anul acesta s-au inscris cu 10 elevi mai multi fata de anul anterior.

OUTPUT

Anul trecut, din 32 de copii, 18 s-au inscris la cercul de pictura, 13 la cel de dansuri si restul la cercul de carting.

Din acestia, pe parcursul anului, 3 au abandonat toate cercurile pe care le frecventau.

Anul acesta s-au inscris cu 10 elevi mai multi fata de anul anterior.

14. Se citeste din fisierul *text.txt* un text oarecare (pe una sau mai multe linii, eventual vide). Sa se afiseze in fisierul *rezultat.txt* numarul valorilor cu exact 4 cifre din fisierul *text.txt*. Nu se vor folosi structuri repetitive si se vor utiliza expresii regulate.

Exemplu:

INPUT

text.txt

Primul razboi mondial s-a declansat in Europa si a durat de la 28 iulie 1914 pana pe 11 noiembrie 1918, la care au participat peste 70 de milioane de militari, inclusiv 60 de milioane de europeni, mobilizati intr-unul dintre cele mai mari razboaie din istorie (sursa: *https://ro.wikipedia.org/wiki/Primul_R%C4%83zboi_Mondial*).

OUTPUT

rezultat.txt

2

```
import re
```

```
print(len(list(filter(lambda                                    x:
len(x)==4,re.findall('\d+',open("date.txt",'r').read()))))))
```

15. Pentru un turist ce doreste sa se cazeze la pensiunea *Viata fara griji,* cititi de la tastatura, in aceasta ordine: CNP-ul, numele, o data de cazare, o data de decazare (datele vor fi în formatul *zz-ll-aa*), numarul de adulti si numarul de copii (cu varsta intre 5 si 12 ani). Copiii cu varsta peste 12 ani vor fi considerati adulti. Copiii (cu varsta intre 5 si 12 ani) au o reducere de 50%. Camerele sunt duble (de cate doua persoane). Intregului sejur se aplica, in plus, o reducere de 10% daca numarul de nopti pentru care s-a facut cazarea este mai mare de 5. Tarifele stabilite de pensiune sunt:

- cazare in zilele de duminica si de luni pana joi: 100 lei / camera;
- cazare in zilele de vineri si sambata: 120 lei / camera.

Folositi metoda *safe_substitute* pentru a afisa un mesaj de forma:

Stimate/stimata domnule/doamna (in functie de CNP) ...(nume, prenume)……..,

Ati solicitat ...(completati)... nopti de cazare, din care ....(completati).... tarifate cu 100 lei / noapte si …....(completati).... cu 120 lei / noapte. Din ……(completati).... persoane, ...(completati)…… sunt adulti si ...(completati)... sunt copii. Numar de camera ocupate: ....(completati).... Nu beneficiati / Beneficiati de reducerea de 10% aplicata pentru sejururile mai mari de 5 nopti.

Total de plata: ……(completati) ..... lei.

Va dorim un sejur placut!

Pensiunea "Viata fara griji"

Exemplu:

INPUT

2801012300800

Popescu Diana

02-06-21

08-06-21

3

2

OUTPUT

Stimata doamna Popescu Diana,

Ati solicitat 6 nopti de cazare, din care 4 tarifate cu 100 lei / noapte si 2 cu 120 lei / noapte. Din 5 persoane, 3 sunt adulti si 2 sunt copii. Numar de camere ocupate: 3. Beneficiati de reducerea de 10% aplicata pentru sejururile mai mari de 5 nopti.

Total de plata: 1728 lei.

Va dorim un sejur placut!

Pensiunea "Viata fara griji"

16. In fisierul de intrare *plecare_Bucuresti.txt* se afla, pe cate o linie, numele unui oras, urmat de ": ", impreuna cu pretul unui bilet, de adult, clasa a 2-a, cu loc, de la compania CFR din orasul Bucuresti catre respectivul oras (regim IR – InterRegio). Pretul este numar zecimal (in care se foloseste "." ca separator – ca in exemplul de mai jos).

CFR are urmatoarele reduceri:

- 15%, daca biletul se cumpara cu anticipatie de cel putin 21 de zile;
- 10%, daca biletul se cumpara cu anticipatie intre 11 si 20 de zile;
- 5%, daca biletul se cumpara cu anticipatie intre 6 si 10 de zile.

Pentru calculul numarului de zile se numara ziua plecarii, dar nu ziua in car este cumparat biletul.

Sa se citeasca de la tastatatura o data calendaristica, in formatul zz-ll-yyyy (exemplu: 12-03-2020), ce reprezinta data plecarii intr-o calatorie, pentru care vrem sa achizitionam biletul si un oras (dintre cele din fisier). Sa se afiseze in consola care este pretul biletului, cu reducerea maxima ce se poate aplica.

INPUT

plecare_Bucuresti.txt

Brasov: 48.6

Galati: 60.1

Arad: 100.9

Cluj-Napoca: 90.8

Iasi: 90.8

Consola:

Arad

22-06-2021

OUTPUT

90.81

Explicatie (nu face parte din OUTPUT): considerand ca se cumpara biletul azi, 2.06.2021, se poate aplica reducerea maxima de 10% (mai sunt 20 de zile pana in ziua plecarii: 22-06-2021).

17. In fisierul de intrare *plecare_Bucuresti.txt* se afla, pe cate o linie, numele unui oras, urmat de ": ", impreuna cu pretul unui bilet, de adult, clasa a 2-a, cu loc, de la compania CFR din orasul Bucuresti catre respectivul oras (regim IR – InterRegio). Pretul este numar zecimal (in care se foloseste "." ca separator – ca in exemplul de mai jos). CFR aplica o reducere de weekend (sambata-duminica) de 15%. Fiind sezonul estival, la aceasta reducere se adauga o reducere de 10% (indiferent de ziua din saptamana in care are loc calatoria) daca destinatia este un oras din fisierul *orase_reduceri.txt*, in care numele oraselor sunt separate prin virgule/spatii/pe mai multe linii.

Sa se citeasca de la tastatura doua date calendaristice in formatul zz-ll-yyyy (exemplu: 12-03-2020), ce reprezinta data plecarii si data intoarcerii si orasul de destinatie. Sa se afiseze pretul total al celor doua bilete, dupa aplicarea tuturor reducerilor posibile.

INPUT

plecare_Bucuresti.txt

Brasov: 48.6

Constanta: 59.6

Galati: 60.1

Arad: 100.9

Cluj-Napoca: 90.8

Iasi: 90.8

Mangalia: 60.2

orase_reduceri.txt

Constanta, Mangalia

Consola:

12-06-2021

14-06-2021

Constanta

OUTPUT

98.34

Explicatii (nu fac parte din OUTPUT): 12-06-2021 este zi de weekend, deci se aplica reducerea de 15%, iar 14-06-2021 este zi de luni (fara reducere). Pentru ambele bilete (zile) se adauga reducerea pentru orasul Constanta.

18. Datele de *25.12* si *26.12* din fiecare an sunt zile de concediu (indiferent de ziua din saptamana in care cad). Stiind ca intodeauna sambata si duminica sunt zile libere, afisati pentru anul curent care va fi prima zi lucratoare de dupa Craciun (in format *zz.ll.aaaa*).

Exemple: pentru anul *2020* s-ar afisa *28.12.2020*, iar pentru anul *2021* s-ar afisa *27.12.2021*

19. Intr-o companie, data de 1 mai este libera. Daca aceasta cade intr-o zi de joi, se considera ca ziua de vineri va fi, de asemenea, zi libera. Similar, daca 1 mai ar cadea intr-o zi de marti, atunci si ziua de luni va fi libera. Afisati care sunt toate zilele libere din preajma lui 1 mai anul curent, conform regulii anterioare.

Exemple: pentru anul 2019 s-ar afisa 1.05.2019 (1 mai 2019 a fost miercuri), pentru anul 2020 s-ar afisa 1.05.2020, 2.05.2020, 3.05.2020 (1 mai 2020 a fost vineri), pentru anul 2021 s-ar afisa 1.05.2021, 2.05.2021 (1 mai 2021 va fi sambata), iar pentru anul 2025 s-ar afisa 1.05.2025, 2.05.2025, 3.05.2025, 4.05.2025 (1 mai 2025 cade joia).

20. Se citește de la tastatură un număr natural *n*, apoi, pentru *n* studenți, pe câte o linie, separate prin câte o virgulă și un spațiu, numele și patru valori naturale, reprezentând notele pentru patru examene. Pentru fiecare student să se afișeze în consolă *numele, numarul de restante si media examenelor promovate*, mai întâi pentru studenții integraliști (cu toate notele mai mari sau egale cu 5) în ordinea descrescătoare a mediilor, iar pentru ceilalți în ordinea crescătoare a numărului de restanțe; pentru cei cu același numar de restanțe, afisarea se va face după media examenelor promovate (vezi exemplul – numarul de zecimale pentru medie nu este important).

Exemplu:

INPUT

5

Ionescu Marian, 5, 4, 7, 10

Popescu Elena, 4, 3, 7, 5

Pop Dana, 4, 4, 5, 6

Florea Laura, 10, 9, 7, 8

Neagu Andrei, 9, 10, 9, 10

OUTPUT

Neagu Andrei, 0 restante, media 9.50

Florea Laura, 0 restante, media 8.50

Ionescu Marian, 1 restante, media 7.33

Popescu Elena, 2 restante, media 6

Pop Dana, 2 restante, media 5.5

21. Din fișierul *text_literar.txt* se citește un text pe mai multe linii (un număr par). Să se afișeze un mesaj corespunzător: “Poezie” / “Proza”. Pentru ca textul să fie poezie trebuie ca între liniile din fișier să existe una dintre rimele:

    - împerecheată (rimează versurile 1-2, 3-4, 5-6, etc.);
    - încrucișată (rimează versurile 1-3, 2-4, 5-7, etc.);
    - îmbrățișată (rimează versurile 1-4, 2-3, 5-8, 6-7, etc);
    - monorimă (rimează versurile 1-2-3-4, 5-6-7-8, etc).

Consideram ca doua versuri rimeaza daca au ultimele 3 caractere identice.

22. (Aveti un curs cu un exemplu care va poate ajuta - media varstelor unor persoan,e generate aleator)

Din fisierul *angajati.in* se citesc, de pe cate o linie a fisierului, numele si prenumele cate unui angajat. Se citeste de la tastatura un numar natural *k*, mai mic decat numarul liniilor din fisierul *angajati.in*. Să se afiseze in fisierul *extragere.out* k nume diferite de angajati, la intamplare, din cele existente in fisierul *angajate.in.*

Exemplu:

Daca fisierul *angajati.in* contine

Jercan Elena

Ion Horia

Marin Ioana-Daniela

Neagoe Marina

Florescu Teodora

Popa Diana

Radu Tatiana

si k = 4, atunci fisierul *extragere.out* poate contine

Neagoe Marina

Popa Diana

Radu Tatiana

Ion Horia

23. Se da un text literar in limba romana, fara diacritice, care respecta toate regulile ortografice ale limbii romane (fara spatii nepotrivite, semne de punctuatie gresite, etc). Să se afișeze in consola, pe o singura linie, separate prin virgule, cuvintele din text ce contin trei consoane alaturate. Se va folosi modulul *re*, utilizând un tipar pentru cuvinetel ce indeplinesc condiția ceruta.

Exemplu:

Input:

s = "Dintr-un nimic s-a transformat intr-o situatie intr-adevar grava."

Output:

Dintr-un, transformat, intr-o, intr-adevar