



**Universidade de Brasília**

Departamento de Ciência da Computação

## **Introdução à Ciência da Computação - 113913**

### **Gabarito da Lista de Exercícios 7**

#### **Tuplas e Dicionários**

##### **Observações:**

- As listas de exercícios serão corrigidas por um **corretor automático**, portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado em cada questão (exemplo de entrada e saída). Por exemplo, a não ser que seja requisitado na questão, não use mensagens escritas durante o desenvolvimento do seu código como “Informe a primeira entrada”. Estas mensagens não são tratadas pelo corretor, portanto a correção irá resultar em **resposta errada**, mesmo que seu código esteja correto;
- As instâncias de Entrada serão as usadas pelo corretor e suas saídas deve estar **iguais** às apresentadas em Instâncias de Saída.

# Borderline

```
N = int(input())
```

```
personal = { }
```

```
for i in range(N):
```

```
    T, P = input().split()
```

```
    personal[T] = P
```

```
F = input().split()
```

```
out = []
```

```
for word in F:
```

```
    if word in personal:
```

```
        out.append(personal[word])
```

```
if len(out) == 0:
```

```
    print("Tudo bem!")
```

```
else:
```

```
    print(' '.join(out))
```

Instâncias de Entrada	Instâncias de Saída
0 Não importa muito o que eu puser aqui , tem que sempre ter ' Tudo bem ! ' no final : )	Tudo bem!
1 palavra normal Tudo bem	Tudo bem!
1 palavra normal palavra	normal
2 tomate sedutora tiara reporter tomei um suco de tomate e achei estranho	sedutora
3 A X B Y C D A B C	X Y D

Instâncias de Entrada	Instâncias de Saída
<p>7</p> <p>paralelepípedo xatiada chocolate confusa olho encabulada papelzinho eufórica espinha concentrada ventilador entediada toalhinha nazaré a toalhinha em cima do ventilador está impedindo o vento de chegar no papelzinho e acertar o seu olho</p>	<p>nazaré entediada eufórica encabulada</p>
<p>10</p> <p>Etiam suscipit sit amet justo id elementum Duis et lectus enim Ut eleifend imperdiet gravida Integer erat quam iaculis nec iudhwe id dwe . Duis lectus uidhew</p>	<p>Tudo bem!</p>
<p>10</p> <p>Vivamus eget dignissim ante Phasellus sit amet quam sit amet massa porta gravida Sed a efficitur tortor sit amet rhoncus Illus sit amet quam sit amet massa p</p>	<p>amet rhoncus amet rhoncus porta</p>
<p>1</p> <p>palavra palavra palavra palavra</p>	<p>palavra palavra</p>

23

Morbi malesuada

neque sed

tincidunt Ut

imperdiet nunc

ac euismod

porta mauris

purus convallis

eros ac

ornare nisi

risus ut

orci Praesent

vestibulum porta

justo sit

amet posuere

est gravida

eget Integer

id sagittis

elit Curabitur

cursus gravida

maximus risus

placerat et

Nunc tincidunt

vulputate tellus

Aenean porta lorem quis tempor semper .

Donec est libero , pulvinar a pulvinar vel ,

facilisis non nisi . Duis porta sollicitudin

massa eu hendrerit . Ut auctor est dui , nec

scelerisque nisl lobortis in . Praesent in

tincidunt elit . Nam gravida , elit a porttitor

viverra , mi lorem cursus magna , at

fermentum eros risus nec nisi . Aenean

pulvinar convallis nunc , non bibendum

eros mattis varius . Curabitur vulputate ,

mauris ut euismod ornare , augue eros

tempus dolor , ac laoreet libero est vel elit .

Integer ac felis dapibus , facilisis nulla nec ,

volutpat eros . Phasellus vulputate euismod

diam , eget feugiat erat euismod vel .

Vivamus molestie tortor quis lorem rhoncus

, nec vehicula nulla scelerisque . Mauris ac

dapibus augue , at cursus elit . Nam vel mi

imperdiet , pharetra nunc quis , malesuada

leo . Ut rutrum diam eget nibh dapibus

accumsan . Donec facilisis lorem vel

rhoncus pulvinar . Etiam et vestibulum

mauris . Nunc ac ex imperdiet , rutrum

lorem et , sollicitudin urna . Nam

elementum vestibulum enim in tempus . In

fermentum neque quis ligula hendrerit , vel

tristique orci rhoncus . Fusce faucibus ,

massa id varius vestibulum , tortor mi

mauris gravida mauris gravida Ut Curabitur  
Curabitur gravida ac ut ac tellus nisi ac  
euismod gravida Curabitur euismod ac  
tellus Integer euismod gravida Curabitur  
nunc Integer porta tincidunt euismod nunc  
porta sed Praesent sagittis porta Integer  
euismod euismod convallis et Curabitur  
malesuada ut Integer Curabitur gravida  
Praesent gravida Ut gravida ut tellus Ut nisi  
sagittis Integer Curabitur

efficitur erat , eu facilisis nulla felis eget  
tellus . Vivamus condimentum , lacus  
pellentesque pretium pulvinar , leo lectus  
consectetur odio , non pulvinar dolor nisi  
vel augue . Phasellus ac vehicula leo . Duis  
ac purus mattis , placerat elit sed , dictum  
erat . Morbi at enim aliquet , luctus risus ut ,  
facilisis ipsum . Curabitur semper , mi eu  
venenatis fermentum , nibh dui molestie  
velit , nec laoreet lacus sapien et velit .  
Donec interdum eget elit a cursus . Aenean  
sollicitudin feugiat urna , vel porttitor nulla  
pellentesque nec . Vestibulum ante ipsum  
primis in faucibus orci luctus et ultrices  
posuere cubilia Curae; In sed tempor lectus  
 , sed cursus magna . Donec sollicitudin  
efficitur efficitur . Fusce ante magna ,  
tincidunt varius dolor sed , tempor porttitor  
diam . Aenean ultricies felis eu massa  
molestie pharetra . Donec eu est risus .  
Pellentesque interdum vulputate tincidunt .  
Fusce et suscipit metus . Cras ornare  
accumsan lorem , at tempus quam varius id  
 . Vivamus fringilla lorem eget varius  
ullamcorper . Nulla ut urna dignissim ,  
pellentesque nisi quis , bibendum nisl . Sed  
porttitor nunc et tellus sollicitudin , at  
consectetur elit lacinia .

# Depita

```
1 N = int(input())
2
3 deps = {}
4
5 for i in range(N):
6     S = input().split()
7     deps[S[0].lower()] = ' '.join(S[1:])
8
9 print(len(deps))
10
11 for S in sorted(deps):
12     print(deps[S])
```

A função da `sorted` linha 11 retorna uma cópia ordenada do dicionário, pelas chaves. Uma outra forma de realizar a mesma operação é utilizar o método `deps.keys()`, ordená-lo ele usando `sort` e iterar sobre ele para conseguir as chaves na ordem correta. O `.lower()` da linha 7 serve para impedir que a ordenação leve em consideração case.

Instâncias de Entrada	Instâncias de Saída
0	0
2 A C C A	2 C A
2 D A A D	2 D A
9 Y djwioe_dwieudw a b c d e b c d e a j d8wheiudhwui c d e a b du dijeiw d e a b c e a b c d HHH udehwid	9 b c d e c d e a d e a b e a b c dijeiw a b c d udehwid d8wheiudhwui djwioe_dwieudw
4 Dbda 00000 Dcaba 11111 ddaca 2222 daaaa -1111	4 -1111 00000 11111 2222

Instâncias de Entrada	Instâncias de Saída
5 G dnewdnw G ieo2j3o2e G dowijeoijwo G dweioow G idoewj	1 idoewj
6 Dbda 00000 Dbda 00-000 Dcaba 11111 ddaca JJJJJ ddaca 2222 daaaa -1111	4 -1111 00-000 11111 2222
4 AB AAA CD AAA EF AAA AB BBB	3 BBB AAA AAA
9 Cras varius quam quis metus scelerisque sollicitudin. Pellentesque mollis tempus euismod. quisque id varius leo. Donec ultricies sem in sapien scelerisque, vitae hendrerit sem finibus. quisque ut leo a sapien auctor varius. Phasellus vehicula tortor a dolor varius, in vulputate lectus bibendum. mauris egestas mattis lacus, a dignissim massa consectetur et. Nam at fermentum dui. ded dictum placerat ante ac tempus	8 varius quam quis metus scelerisque sollicitudin. dictum placerat ante ac tempus ultricies sem in sapien scelerisque, vitae hendrerit sem finibus. egestas mattis lacus, a dignissim massa consectetur et. at fermentum dui. mollis tempus euismod. vehicula tortor a dolor varius, in vulputate lectus bibendum. ut leo a sapien auctor varius.

Instâncias de Entrada	Instâncias de Saída
<p>36</p> <p>Ut pulvinar neque sed nulla pulvinar in lobortis mauris suscipit Pellentesque tincidunt vehicula porta Proin egestas felis massa et vulputate elit sodales a Integer at erat sed libero cursus ornare Vivamus id enim nibh In feugiat velit convallis sem rhoncus eu ultricies est semper Sed vel nulla id nulla tincidunt ornare a at libero Nunc purus risus ultrices ac nisl eget ultrices aliquet nisi Nulla magna purus rutrum ac euismod consequat pellentesque vel metus Mauris ultrices urna ut vestibulum posuere Ut tempus magna in consequat laoreet metus lorem elementum purus id vestibulum urna leo vitae libero Sed vitae massa sollicitudin lacinia eros quis interdum nisi Nulla vitae ante eu orci cursus vestibulum Vestibulum viverra ac felis eget sagittis Quisque cursus interdum justo vel placerat lectus auctor id Duis turpis urna rutrum sit amet aliquam at finibus sit amet turpis Praesent nec suscipit massa Suspendisse imperdiet neque vitae condimentum congue Fusce et mauris odio Nulla fermentum metus et massa luctus non hendrerit nulla tristique</p>	<p>28</p> <p>turpis urna vulputate elit sodales a ultrices est semper sit amet turpis et mauris odio vestibulum urna leo vitae libero feugiat velit convallis sem rhoncus at erat sed libero cursus ornare nisi eros quis in consequat laoreet ultrices urna ut vestibulum posuere lorem elementum purus hendrerit nulla tristique fermentum metus et massa luctus purus risus vel metus nec suscipit massa egestas felis massa cursus interdum justo sit amet aliquam at vitae massa sollicitudin imperdiet neque vitae condimentum congue aliquet nisi tempus placerat lectus auctor id viverra ac felis eget sagittis id enim nibh</p>



# Deus Ex Machina

```
1  N = int(input())
2
3  list_of_problems = []
4
5  for i in range(N):
6      P, S, D = input().split()
7      D = int(D)
8
9      list_of_problems.append((S, D))
10
11
12  o_list = sorted(list_of_problems, key=lambda x: -x[1])
13
14  for solution in o_list:
15      print(solution[0], end="")
16  print() #for newline
```

A função `sorted` é garantida de fazer uma ordenação estável, então a ordem de input é preservada no caso de empate.

Instâncias de Entrada	Instâncias de Saída
0	
1 a a 1	a
3 a a 1 b b 2 c c 2	bca
3 a a 10 b b 5 c c 1	abc
3 a a 1 b b 5 c c 10	cba
5 uwehdwde d 1 iwudnieuw c 2 ndiuewwd e 3 a f 4 b g 5	gfecd

Instâncias de Entrada	Instâncias de Saída
5 d uwei 10 c udwiendnei w 9 b yewedee 8 a iwemdwe 9 _ _ 7	uwe iudwiendneiwiwemdweyewedee_
10 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1 a a 1	aaaaaaaaaa
8 duewn dewyhew 8 dweiu uiwejd w 7 dwnhehndew diewwd 8 dziej ndwueindmw 10 dwjhenw dwnedwmd 6 dneiwdmew uidwjdei w 1 dwmedow dwemodmw 1 dmwiedmeiw due 4	ndwueindmw dewyhew diewwd uiwejd w dwnedw md due uidwj dei wd w emodmw

Instâncias de Entrada	Instâncias de Saída
<p>49  Duis sit 5  amet odio 8  vitae diam 2  tempus aliquet. 3  Aenean fringilla 4  interdum risus, 4  id ultrices 2  erat rhoncus 8  ac. Sed 2  eu tellus 9  vestibulum leo 7  mattis aliquet 1  sit amet 6  vel leo. 3  Fusce eget 4  euismod enim. 8  Maecenas consectetur 3  vehicula diam, 1  ut interdum 6  justo commodo 3  id. Maecenas 5  iaculis lectus 7  metus. Fusce 6  porta, nisl 2  eget accumsan 5  rutrum, sem 2  nisi maximus 1  tellus, a 1  semper sapien 5  augue ut 5  augue. Vivamus 1  scelerisque, massa 0  et convallis 3  ullamcorper, mauris 9  massa condimentum 2  magna, nec 6  dictum augue 0  tellus in 0  elit. Quisque 1  in elit 2  sed neque 8  fermentum elementum. 9  Aliquam enim 7  diam, scelerisque 7  non dictum 9  a, molestie 1  nec ante. 3  Vestibulum ante 2  ipsum primis 5</p>	<p>tellusmauriselementum.dictumodiorhonusenim  .nequelelectusenimscelerisqueametinterdumF  uscenecsitMaecenasaccumsansapienutprimisfr  ingillarius,egetaliquet.leo.consecteturcommod  oconvallisante.diamultricesSednislsemcondime  ntumelitantealiquetdiam,maximusaVivamusQui  squemolestiemassaauguein</p>

# Estrada

```
1  N = int(input())
2
3  def move(pos, D, Q):
4      if D == "N":
5          return (pos[0]+Q, pos[1])
6      elif D == "S":
7          return (pos[0]-Q, pos[1])
8      elif D == "L":
9          return (pos[0], pos[1]+Q)
10     elif D == "O":
11         return (pos[0], pos[1]-Q)
12
13 def format(y, x):
14     return (
15         -y if y < 0 else 0,
16         y if y > 0 else 0,
17         -x if x < 0 else 0,
18         x if x > 0 else 0
19     )
20
21 pos = (0, 0)
22
23 for i in range(N):
24     D, Q = input().split()
25     pos = move(pos, D, int(Q))
26
27 print("%d %d %d %d" % format(*pos))
```

Evidentemente esta solução não precisa das funções move e format, mas achei um exemplo apropriado de funções que retornam mais de um valor.

Instâncias de Entrada	Instâncias de Saída
0	0 0 0 0
4 N 1 S 1 O 1 L 1	0 0 0 0

Instâncias de Entrada	Instâncias de Saída
10 N 7 L 28 S 14 O 10 N 19 N 10 N 28 S 100 S 45 O 10	95 0 0 8
5 O 5 O 5 O 5 O 5 O 5	0 0 25 0
5 L 10 L 10 L 10 L 10 L 10	0 0 0 50
6 S 3 S 3 S 3 S 3 S 3 S 3	18 0 0 0

Instâncias de Entrada	Instâncias de Saída
20 S 22 S 63 L 27 O 83 O 10 O 9 S 100 O 32 S 76 N 11 S 56 L 68 L 55 N 42 S 61 O 21 O 20 L 31 S 93 N 69	349 0 0 6
26 N 283 N 201 S 511 N 914 S 443 O 700 S 513 S 692 N 92 S 329 L 187 O 69 S 657 N 542 S 706 N 645 S 2 S 946 O 641 S 77 L 398 O 242 O 315 N 852 L 964 N 371	976 0 418 0

Instâncias de Entrada	Instâncias de Saída
47 O 628 S 782 S 782 L 433 S 987 O 810 O 72 O 34 N 759 L 337 S 883 O 596 L 74 S 299 S 890 L 954 N 11 L 183 O 392 L 550 N 93 S 969 S 697 N 721 S 88 L 95 N 308 N 433 O 742 S 467 O 913 N 414 N 161 N 394 N 213 S 133 O 779 S 954 N 755 S 500 S 417 S 480 L 197 N 876 S 764 O 463 O 108	4954 0 2714 0

Instâncias de Entrada	Instâncias de Saída
19 O 86 S 64 L 102 O 335 N 250 O 799 N 163 N 19 O 766 N 594 O 775 L 210 O 487 O 925 L 3767 L 135 N 832 N 148 S 1904	0 38 0 41



# Enumeração

```
# remove pontuação
trecho = input().replace(".", "").replace(",", "").replace("!", "").replace("?", "").replace(":", "")

d = {}

for word in trecho.split():
    word = word.lower()

    if word in d:
        d[word] = d[word] + 1
    else:
        d[word] = 1

for word in sorted(d, key=lambda x: d[x], reverse=True):
    print(word.capitalize(), d[word])
```

O atributo `reverse` da função `sorted` permite criarmos a lista ordenada ao contrário (maior para menor), mas você também pode fazer um loop a mais para criar uma lista nova, invertida.

Instâncias de Entrada	Instâncias de Saída
<p>But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes,</p>	<p>Of 5  The 5  And 3  I 2  You 2  But 1  Must 1  Explain 1  To 1  How 1  All 1  This 1  Mistaken 1  Idea 1  Denouncing 1  Pleasure 1  Praising 1  Pain 1  Was 1  Born 1  Will 1  Give 1  A 1  Complete 1  Account 1  System 1  Expound 1  Actual 1  Teachings 1  Great 1  Explorer 1  Truth 1  Master-builder 1  Human 1  Happiness 1  No 1  One 1  Rejects 1  Dislikes 1</p>

Instâncias de Entrada	Instâncias de Saída
<p>illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolo</p>	<p>Quia 3  Voluptatem 2  Aut 2  Qui 2  Illo 1  Inventore 1  Veritatis 1  Et 1  Quasi 1  Architecto 1  Beatae 1  Vitae 1  Dicta 1  Sunt 1  Explicabo 1  Nemo 1  Enim 1  Ipsam 1  Voluptas 1  Sit 1  Aspernatur 1  Odit 1  Fugit 1  Sed 1  Consequuntur 1  Magni 1  Dolores 1  Eos 1  Ratione 1  Sequi 1  Nesciunt 1  Neque 1  Porro 1  Quisquam 1  Est 1  Dolorem 1  Ipsum 1  Dolo 1</p>

<p>Etiam orci urna, tincidunt vitae velit eu, sagittis varius ante. Maecenas molestie tempor ante vitae molestie. Fusce urna ex, ornare pretium ultricies in, tincidunt eget nisi. Pellentesque velit lacus, fringilla commodo orci quis, scelerisque convallis magna. Suspendisse sodales ac dolor non molestie. Sed eu mi vulputate, hendrerit mauris in, scelerisque quam. Quisque tincidunt sodales fermentum. Mauris dapibus nunc ut neque luctus sagittis. Phasellus sollicitudin enim tellus, eu finibus orci congue vitae. Sed vel dapibus lorem. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.</p>	<p>Orci 3 Tincidunt 3 Vitae 3 Eu 3 Molestie 3 Urna 2 Velit 2 Sagittis 2 Ante 2 In 2 Pellentesque 2 Scelerisque 2 Sodales 2 Ac 2 Sed 2 Mauris 2 Dapibus 2 Et 2 Etiam 1 Varius 1 Maecenas 1 Tempor 1 Fusce 1 Ex 1 Ornare 1 Pretium 1 Ultricies 1 Eget 1 Nisi 1 Lacus 1 Fringilla 1 Commodo 1 Quis 1 Convallis 1 Magna 1 Suspendisse 1 Dolor 1 Non 1 Mi 1 Vulputate 1 Hendrerit 1 Quam 1 Quisque 1 Fermentum 1 Nunc 1 Ut 1 Neque 1 Luctus 1 Phasellus 1 Sollicitudin 1 Enim 1 Tellus 1</p>
---	---

	<div>Finibus 1</div> <div>Congue 1</div> <div>Vel 1</div> <div>Lorem 1</div> <div>Habitant 1</div> <div>Morbi 1</div> <div>Tristique 1</div> <div>Senectus 1</div> <div>Netus 1</div> <div>Malesuada 1</div> <div>Fames 1</div> <div>Turpis 1</div> <div>Egestas 1</div>
--	--

Etiam tellus justo, rutrum ut quam congue, ullamcorper dictum dui. Vivamus gravida lacinia ipsum, et mattis libero. Pellentesque vel risus urna. Cras augue purus, fermentum eu odio sed, dictum molestie leo. Vivamus turpis nisl, malesuada ac tempus quis, mollis quis velit. Proin efficitur elementum imperdiet. Aenean at purus sit amet augue efficitur blandit vitae porttitor nisl.	Dictum 2 Vivamus 2 Augue 2 Purus 2 Nisl 2 Quis 2 Efficitur 2 Etiam 1 Tellus 1 Justo 1 Rutrum 1 Ut 1 Quam 1 Congue 1 Ullamcorper 1 Dui 1 Gravida 1 Lacinia 1 Ipsum 1 Et 1 Mattis 1 Libero 1 Pellentesque 1 Vel 1 Risus 1 Urna 1 Cras 1 Fermentum 1 Eu 1 Odio 1 Sed 1 Molestie 1 Leo 1 Turpis 1 Malesuada 1 Ac 1 Tempus 1 Mollis 1 Velit 1 Proin 1 Elementum 1 Imperdiet 1 Aenean 1 At 1 Sit 1 Amet 1 Blandit 1 Vitae 1 Porttitor 1
--	---

Instâncias de Entrada	Instâncias de Saída
<p>Sed dictum maximus faucibus. Phasellus eleifend augue magna, sed lobortis libero iaculis et. Integer luctus ex eros, ut auctor leo scelerisque nec. In finibus dapibus tortor, et hendrerit ex varius quis.</p> <p>Suspendisse luctus tellus ut nisl mattis, sed faucibus erat porta. Nulla nunc sem, luctus pulvinar ultricies ac, ultricies tempus nisl.</p> <p>Morbi in malesuada metus. Integer a tincidunt metus, eget lacinia nunc.</p>	<p>Sed 3</p> <p>Luctus 3</p> <p>Faucibus 2</p> <p>Et 2</p> <p>Integer 2</p> <p>Ex 2</p> <p>Ut 2</p> <p>In 2</p> <p>Nisl 2</p> <p>Nunc 2</p> <p>Ultricies 2</p> <p>Metus 2</p> <p>Dictum 1</p> <p>Maximus 1</p> <p>Phasellus 1</p> <p>Eleifend 1</p> <p>Augue 1</p> <p>Magna 1</p> <p>Lobortis 1</p> <p>Libero 1</p> <p>Iaculis 1</p> <p>Eros 1</p> <p>Auctor 1</p> <p>Leo 1</p> <p>Scelerisque 1</p> <p>Nec 1</p> <p>Finibus 1</p> <p>Dapibus 1</p> <p>Tortor 1</p> <p>Hendrerit 1</p> <p>Varius 1</p> <p>Quis 1</p> <p>Suspendisse 1</p> <p>Tellus 1</p> <p>Mattis 1</p> <p>Erat 1</p> <p>Porta 1</p> <p>Nulla 1</p> <p>Sem 1</p> <p>Pulvinar 1</p> <p>Ac 1</p> <p>Tempus 1</p> <p>Morbi 1</p> <p>Malesuada 1</p> <p>A 1</p> <p>Tincidunt 1</p> <p>Eget 1</p> <p>Lacinia 1</p>

<p>Proin gravida fringilla orci sit amet porttitor. Phasellus facilisis leo ac rhoncus hendrerit. Mauris sit amet leo malesuada, vehicula diam sit amet, imperdiet arcu. Integer eu gravida nibh. Donec imperdiet accumsan pretium. Nam aliquam, lacus quis mollis vestibulum, enim tellus dapibus risus, eu fermentum quam mi pretium velit. Fusce commodo sollicitudin viverra. Sed quis purus eget ex varius condimentum. Aliquam metus ante, vulputate vitae vulputate at, ultricies et risus. Aliquam erat volutpat. Mauris et interdum mauris. In pulvinar justo fringilla ligula suscipit, ac sollicitudin neque porta. Praesent risus arcu, porttitor at lobortis sed, tempus sit amet nisl. Suspendisse tempus commodo volutpat.</p>	<p>Sit 4  Amet 4  Mauris 3  Aliquam 3  Risus 3  Gravida 2  Fringilla 2  Porttitor 2  Leo 2  Ac 2  Imperdiet 2  Arcu 2  Eu 2  Pretium 2  Quis 2  Commodo 2  Sollicitudin 2  Sed 2  Vulputate 2  At 2  Et 2  Volutpat 2  Tempus 2  Proin 1  Orci 1  Phasellus 1  Facilisis 1  Rhoncus 1  Hendrerit 1  Malesuada 1  Vehicula 1  Diam 1  Integer 1  Nibh 1  Donec 1  Accumsan 1  Nam 1  Lacus 1  Mollis 1  Vestibulum 1  Enim 1  Tellus 1  Dapibus 1  Fermentum 1  Quam 1  Mi 1  Velit 1  Fusce 1  Viverra 1  Purus 1  Eget 1  Ex 1</p>
---	---



	Varius 1 Condimentum 1 Metus 1 Ante 1 Vitae 1 Ultricies 1 Erat 1 Interdum 1 In 1 Pulvinar 1 Justo 1 Ligula 1 Suscipit 1 Neque 1 Porta 1 Praesent 1 Lobortis 1 Nisl 1 Suspendisse 1
--	--

<p>In vehicula libero quis facilisis vestibulum. Vivamus lectus justo, lacinia sed luctus eu, tristique id nisi. Sed iaculis aliquam luctus. Nam a blandit ligula. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Integer ut elit condimentum arcu tempus tincidunt. Sed ac turpis tortor. Fusce lacinia bibendum erat id interdum. Aliquam ullamcorper eu enim ac mattis. Praesent gravida enim nec diam condimentum, id tristique felis lobortis. Donec vitae feugiat velit, ultrices pulvinar sem.</p>	<p>Sed 3  Tristique 3  Id 3  Ac 3  Lacinia 2  Luctus 2  Eu 2  Aliquam 2  Et 2  Turpis 2  Condimentum 2  Enim 2  In 1  Vehicula 1  Libero 1  Quis 1  Facilisis 1  Vestibulum 1  Vivamus 1  Lectus 1  Justo 1  Nisi 1  Iaculis 1  Nam 1  A 1  Blandit 1  Ligula 1  Pellentesque 1  Habitant 1  Morbi 1  Senectus 1  Netus 1  Malesuada 1  Fames 1  Egestas 1  Integer 1  Ut 1  Elit 1  Arcu 1  Tempus 1  Tincidunt 1  Tortor 1  Fusce 1  Bibendum 1  Erat 1  Interdum 1  Ullamcorper 1  Mattis 1  Praesent 1  Gravida 1  Nec 1  Diam 1</p>
--	--

	<p>Felis 1 Lobortis 1 Donec 1 Vitae 1 Feugiat 1 Velit 1 Ultrices 1 Pulvinar 1 Sem 1</p>
--	---

Sed sit amet est pellentesque, gravida  
 tortor at, congue lorem. In nec est non nisi  
 tristique sollicitudin. Nulla in ligula  
 pellentesque eros pretium malesuada.  
 Vivamus dignissim sagittis tempus.  
 Praesent porttitor at elit tincidunt dictum.  
 Vivamus vitae mattis risus, et rhoncus  
 mauris. Vivamus id neque porttitor, mollis  
 erat vel, rutrum nisl. Suspendisse potenti.  
 Sed ac lectus at ipsum porta porta sit amet  
 nec velit. Fusce interdum neque sit amet  
 arcu placerat gravida. Nulla consectetur elit  
 arcu, id luctus lorem dignissim sed. In vitae  
 nibh urna. Praesent imperdiet in risus sit  
 amet fringilla. Integer nec eros pharetra  
 eros egestas consectetur.

Sit 4  
 Amet 4  
 In 4  
 Sed 3  
 At 3  
 Nec 3  
 Eros 3  
 Vivamus 3  
 Est 2  
 Pellentesque 2  
 Gravida 2  
 Lorem 2  
 Nulla 2  
 Dignissim 2  
 Praesent 2  
 Porttitor 2  
 Elit 2  
 Vitae 2  
 Risus 2  
 Id 2  
 Neque 2  
 Porta 2  
 Arcu 2  
 Consectetur 2  
 Tortor 1  
 Congue 1  
 Non 1  
 Nisi 1  
 Tristique 1  
 Sollicitudin 1  
 Ligula 1  
 Pretium 1  
 Malesuada 1  
 Sagittis 1  
 Tempus 1  
 Tincidunt 1  
 Dictum 1  
 Mattis 1  
 Et 1  
 Rhoncus 1  
 Mauris 1  
 Mollis 1  
 Erat 1  
 Vel 1  
 Rutrum 1  
 Nisl 1  
 Suspendisse 1  
 Potenti 1  
 Ac 1  
 Lectus 1  
 Ipsum 1  
 Velit 1

	<p>Fusce 1 Interdum 1 Placerat 1 Luctus 1 Nibh 1 Urna 1 Imperdiet 1 Fringilla 1 Integer 1 Pharetra 1 Egestas 1</p>
--	--

Instâncias de Entrada	Instâncias de Saída
<p> Donec DOnec DONec DONEc DONEC  DONEC, DONEC. DONEC! DONEC?  DONEC: dONEC doNEC doNEC donEC  doneC donec donec! Donec? DonEc:  DOnec!!!!!! DONEc?!?!?!  DoNEC:::!!::!!?!?!::!!?!?!::!!::!!  ?!?!?DoNec ?!?!?!DONEc !!!!DoneC?!?!  .....Donec,,,,,,,,!! Donec Donec Donec  Donec Donec Donec Donec Donec Donec  Donec Donec </p>	<p>Donec 37</p>

<p>Nulla aliquet semper elit, eget fringilla enim rutrum quis. Sed quis arcu vitae mauris porta malesuada eget eu nulla. Curabitur fringilla semper nibh, scelerisque pellentesque sem finibus a. Morbi ex risus, laoreet nec urna non, posuere porta justo. Curabitur eleifend, dui eu ullamcorper varius, augue nibh aliquet odio, sit amet fringilla nisi mauris id velit. Pellentesque aliquet elit ac justo pharetra accumsan. Ut at libero consectetur leo porttitor sodales. Sed nec magna purus. Phasellus laoreet nunc id dictum consequat. Proin quis enim blandit, efficitur ante sit amet, lacinia neque.</p>	<p>Aliquet 3 Fringilla 3 Quis 3 Nulla 2 Semper 2 Elit 2 Eget 2 Enim 2 Sed 2 Mauris 2 Porta 2 Eu 2 Curabitur 2 Nibh 2 Pellentesque 2 Laoreet 2 Nec 2 Justo 2 Sit 2 Amet 2 Id 2 Rutrum 1 Arcu 1 Vitae 1 Malesuada 1 Scelerisque 1 Sem 1 Finibus 1 A 1 Morbi 1 Ex 1 Risus 1 Urna 1 Non 1 Posuere 1 Eleifend 1 Dui 1 Ullamcorper 1 Varius 1 Augue 1 Odio 1 Nisi 1 Velit 1 Ac 1 Pharetra 1 Accumsan 1 Ut 1 At 1 Libero 1 Consectetur 1 Leo 1 Porttitor 1 Sodales 1 Magna 1 Purus 1 Phasellus 1</p>
---	---

	<p>Nunc 1 Dictum 1 Consequat 1 Proin 1 Blandit 1 Efficitur 1 Ante 1 Lacinia 1 Neque 1</p>
--	---



# Filosofia

```
1  N = int(input())
2
3  files = []
4
5  for i in range(N):
6      temp = input().split()
7      files.append((temp[0], temp[1:]))
8
9  tags = input().split()
10
11 request = list(filter(lambda f: list(filter(lambda tag: tag in tags, f[1])), files))
12
13 for r in request:
14     print(r[0])
```

A linha 11 pode parecer super complexa, e ela meio que é. Tente quebrá-la em passos para entender melhor o que está acontecendo.

**lambda x: y** cria uma função que toma x de argumento e retorna y, na hora.

`list(filter(x, y))` cria uma lista a partir de y contendo somente os elementos que, quando alimentados para a função x, retornem true.

Consulte o monitor mais próximos para maiores dúvidas.

Instâncias de Entrada	Instâncias de Saída
0 potato	
1 a b c d e c	a
2 a b c d e e d c b a a	d
2 a b c d e e d c b a d	a
2 a b c d e e d c b a c	a d

Instâncias de Entrada	Instâncias de Saída
2 a b c d e e d c b a d a	a d
2 a b c d e e a b c d a	
2 a b c d e e a b c d d	a a
2 a b e h i a e f g j e f	a a

Instâncias de Entrada	Instâncias de Saída
<p>40  Sed at voluptat en im.  Sed non sollicitudin di am.  Vivamus turpis turpis, fa cilisis  eu gravida e u, voluptat  et nibh. Aenean faucibus  turpis in metus voluptat,  a malesuada libe ro pellentesque.  Morbi dictum sapien ut  finibus consequat. Nulla eget  rutrum tellus, ac laoreet  magna. Pellentesque a lo rem  quam. Integer rutrum scelerisque  eros vitae ornare. a t.  Mauris ullamcorper fringilla diam.  Cras non nisl congue,  finibus ex in, pulvinar  ipsum. Vestibulum lectus est,  pellentesque sit amet et varius  in faucibus orci luctus  et ultrices posuere cubilia  Curae; Fusce scelerisque ipsum  quis tempus sollicitudin. Pellentesque  eget odio commodo, lobortis  tellus sed, vestibulum orci.  Pellentesque purus quam, tincidunt  ut varius quis, maximus  at tortor. Pellentesque et  egestas risus, sit amet et  ullamcorper risus. Fusce tincidunt  arcu eget eros. Praesent  id diam malesuada, eleifend  odio quis, consequat nisi.  Aenean porttitor enim id  justo congue ornare. Fusce  auctor ipsum libero, lobortis  molestie metus faucibus in.  Proin ut sodales elit,  sit amet accumsan est.  Nullam condimentum quam eget  purus tristique aliquet. Vivamus  voluptat tristique Pellentesque</p>	<p>Sed  eu  magna.  quis  at  purus</p>

# Função de Ackermann

```
1  computed = dict()
2
3  def Fack(x,y):
4      if not (x,y) in computed:
5          if x == 0:
6              computed[x,y] = y+1
7          elif x == 1:
8              computed[x,y] = y+2
9          elif x == 2:
10             computed[x,y] = 2*y+3
11         elif y == 0:
12             computed[x,y] = Fack(x-1, 1)
13         else:
14             computed[x,y] = Fack(x-1, Fack(x, y-1))
15
16     return computed[x,y]
17
18 X, Y = [int(x) for x in input().split()]
19
20 print(Fack(X, Y))
```

A ideia de computar os valores é permitir o cálculo de valores mais “altos”, sem exceder o nível máximo de recursão do python.

Os testes da linha 7 e 9, apesar de não previstos na fórmula inicial, são triviais de serem deduzidos depois de brincar um tempo com os resultados (teste 2 100, 2 200, 2 300 e tente achar um padrão), e permitem calcular até 3 996, em contraste com o máximo de 3 7 da fórmula crua.







Instâncias de Entrada	Instâncias de Saída
3 8	2045
3 7	1021
3 6	509
4 0	13
0 0	1
1 1000	1002
3 996	53575430359313366047421252453000090528 07024058527668037218751941851755255624 68061246599189407847929063797336458776 57341259357264284615702179922887873492 87401967283887412115492710537302531185 57093897709107652323749179097063369938 37795827719730385314572855982388432710 83830214915826312193418602834034685

<b>Instâncias de Entrada</b>	<b>Instâncias de Saída</b>
2 100	203
2 2000	4003
4 1	65533

# Gabirint

```
1  N = int(input())
2
3  plays = []
4
5  for i in range(N):
6      J, I = input().split()
7      I = int(I)
8
9      plays.append((J, I))
10
11  current = plays[N-1][1]
12  while plays[current-1][1] != N:
13      current = plays[current-1][1]
14
15  print(plays[current-1][0])
```

Instâncias de Entrada	Instâncias de Saída
1 semGraca 1	semGraca
3 yurick 3 qualquer 3 yurick 1	yurick
5 yurick 3 rafael 5 remy 4 joao 2 manoel 5	manoel
10 abacate 8 rafael 3 potato 4 byebye 7 fägel 10 brutamontes 2 joelho 18 abacate 5 rafael 3 potato 1	fägel

Instâncias de Entrada	Instâncias de Saída
10 verifique 10 se 10 ele 10 so 10 esta 10 checando 10 pela 10 presenca 10 de 10 N 6	checando
10 verifique 318 se 420 ele 9999 da 9999999 suporte 0 a -24 numeros 10 invalidos 7 :( 8 :) 9	numeros
5  5  3  1  1  3	
5 5 5 3 3 1 1 1 1 3 3	5
3 >< 3 >< 1 >< 1	><

Instâncias de Entrada	Instâncias de Saída
6 a 5 b 4 c 4 d 1 e 10 f 6	f



# Guloso

```
N = int(input())

if N != 0:
    act = {}

    for i in range(N):
        inp = input().split()
        T, C, F = ''.join(inp[:-2]), inp[-2], inp[-1]

        # AB:CD -> ABCD
        C = int(''.join(C.split(':')))
        F = int(''.join(F.split(':')))

        act[T] = {
            'start': C,
            'end': F
        }

    order = sorted(act, key=lambda x: act[x]['end'])

    S = [order[0]]

    for activity in order[1:]:
        if act[activity]['start'] >= act[S[-1]]['end']:
            S.append(activity)

    print(len(S))
    for activity in S:
        print(activity)
    else:
        print('0')
```

O algoritmo dentro do penúltimo for (`for activity in order[1:]`) é o motivo dessa questão se chamar ‘Guloso’. Essa questão é um clássico problema de seleção de atividades, que é notável por ser sempre resolvível usando a abordagem gulosa.

Instâncias de Entrada	Instâncias de Saída
0	0
1 S 00:01 23:59	1 S
2 A 00:01 12:00 B 12:01 23:59	2 A B

Instâncias de Entrada	Instâncias de Saída
3 A 00:01 23:59 B 00:01 12:00 C 12:01 23:59	2 B C
8 🌮 01:00 20:00 💩 💩 💩 03:15 03:30 Abacate 02:00 03:00 Roberto 04:00 05:00 João 06:00 07:00 Fantástico 02:30 03:10 🌮 🌮 🌮 04:15 23:00 🥑 🥑 🥑 03:40 04:00	5 Abacate 💩 💩 💩 🥑 🥑 🥑 Roberto João
8 🌮 08:00 10:00 💩 🗿 💩 04:15 06:30 Abacate 01:00 09:00 Roberto 09:00 11:00 João 04:00 22:00 Fantástico 03:30 10:10 🌮 🌮 🌮 06:15 23:00 🥑 🥑 🥑 00:40 04:00	3 🥑 🥑 🥑 💩 💩 💩 🌮
8 I'm sorry 01:00 22:00 this 00:01 01:10 question 03:32 04:00 is 01:11 01:30 a 02:01 02:30 letter 02:31 03:00 counting 03:01 03:31 not 01:31 02:00	7 this is not a letter counting question
4 please 22:00 22:01 greedy 22:03 22:22 use 22:02 22:03 algorithm 22:22 22:23	4 please use greedy algorithm

Instâncias de Entrada	Instâncias de Saída
<p>8  refer 04:20 08:00  to 08:30 10:00  <a href="http://tinyurl.com/h3z2aej">http://tinyurl.com/h3z2aej</a> 10:10 22:00  in 10:10 12:00  order to 13:00 20:00  help 21:30 21:45  :) 22:00 23:59  <a href="http://tinyurl.com/2fcpre6">http://tinyurl.com/2fcpre6</a> 10:00 10:09</p>	<p>7  refer  to  <a href="http://tinyurl.com/2fcpre6">http://tinyurl.com/2fcpre6</a>  in  order to  help  :)</p>
<p>1  S 00:01 00:01</p>	<p>1  S</p>

# Filosofighters

```
F = int(input())
matches = {}

def finish_him(T):
    if matches[T][1]:
        return 0
    else:
        return 1 + finish_him(matches[T][0])

for i in range(F):
    T, N = input().split()

    matches[T] = (N, True)

while True:
    inp = input().split()

    if inp[0] == 'FINISHHIM':
        print(finish_him(inp[1]))
        break
    else:
        T, F1, F2, W = inp
        matches[T] = (W, False)
```