**Universidade de Brasília**

Departamento de Ciência da Computação

Introdução à Ciência da Computação - 113913

Gabaritos

**Observações:**

* + - * + Assim como as listas de exercícios, as provas serão corrigidas por um **corretor automático**, portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado (exemplo de entrada e saída).
        + Por este motivo, nunca use mensagens escritas para requisitar input (e.g. ‘Informe o número de casos de uso’). Estas mensagens são consideradas parte do output do seu programa e resultarão em Resposta Errada, mesmo que o resto do 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.

Ada

mix = input()

new\_mix = ""

for c in mix:

if c.isupper():

new\_mix += "\_"

new\_mix += c

splitted = new\_mix.split("\_")

print(“\_“.join([x.capitalize() for x in splitted if x != ""]))

# você também pode fazer algo como

# while "" in splitted: splitted.remove("")

# para remover os - repetidos :)

Poderíamos usar regex ao invés do loop, mas acredito que este recurso esteja fora do escopo desta matéria.

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| simple\_test\_case\_with\_snake\_case\_only | Simple\_Test\_Case\_With\_Snake\_Case\_Only |
| NotSoSimpleTestCaseWithCamelCaseOnly | Not\_So\_Simple\_Test\_Case\_With\_Camel\_Case\_Only |
| simpleMix\_of\_snakeAndCamel\_case\_haHa | Simple\_Mix\_Of\_Snake\_And\_Camel\_Case\_Ha\_Ha |
| THIS\_IS\_SO\_MUCH\_FUN | T\_H\_I\_S\_I\_S\_S\_O\_M\_U\_C\_H\_F\_U\_N |
| what\_is\_love\_BabyDon’tHurtMe\_Don’tHurtMe\_NoMore | What\_Is\_Love\_Baby\_Don’t\_Hurt\_Me\_Don’t\_Hurt\_Me\_No\_More |
| eu-selecionei-o-preto\_eu\_selecionei\_o\_azulEEuSelecioneiOAmarelo | Eu-selecionei-o-preto\_Eu\_Selecionei\_O\_Azul\_E\_Eu\_Selecionei\_O\_Amarelo |
| moar\_one\_type\_of\_case\_testing | Moar\_One\_Type\_Of\_Case\_Testing |
| QuIsque\_susCIpIt\_Cursus\_nIsI,\_ultrICes\_rutrum\_duI\_pulvInar\_a.\_AlIquam\_ante\_lorem,\_sCelerIsque\_In\_justo\_vItae,\_fInIbus\_laCInIa\_sapIen.\_Pellentesque\_ut\_neque\_eu\_odIo\_eleIfend\_molestIe\_non\_non\_maurIs.\_Nulla\_venenatIs\_laCInIa\_augue\_sed\_trIstIque.\_ProIn\_eget\_lIgula\_vel\_neque\_ultrICIes\_sagIttIs.\_ProIn\_et\_quam\_laCInIa\_Ipsum\_bIbendum\_porttItor\_In\_ornare\_massa.\_NunC\_malesuada\_dapIbus\_rIsus\_frIngIlla\_Commodo.\_QuIsque\_neC\_alIquam\_ex.\_CurabItur\_sIt\_amet\_eleIfend\_velIt.\_FusCe\_turpIs\_nIsl,\_susCIpIt\_at\_vestIbulum\_neC,\_hendrerIt\_a\_est.\_Integer\_Consequat\_pharetra. | Qu\_Isque\_Sus\_C\_Ip\_It\_Cursus\_N\_Is\_I,\_Ultr\_I\_Ces\_Rutrum\_Du\_I\_Pulv\_Inar\_A.\_Al\_Iquam\_Ante\_Lorem,\_S\_Celer\_Isque\_In\_Justo\_V\_Itae,\_F\_In\_Ibus\_La\_C\_In\_Ia\_Sap\_Ien.\_Pellentesque\_Ut\_Neque\_Eu\_Od\_Io\_Ele\_Ifend\_Molest\_Ie\_Non\_Non\_Maur\_Is.\_Nulla\_Venenat\_Is\_La\_C\_In\_Ia\_Augue\_Sed\_Tr\_Ist\_Ique.\_Pro\_In\_Eget\_L\_Igula\_Vel\_Neque\_Ultr\_I\_C\_Ies\_Sag\_Itt\_Is.\_Pro\_In\_Et\_Quam\_La\_C\_In\_Ia\_Ipsum\_B\_Ibendum\_Portt\_Itor\_In\_Ornare\_Massa.\_Nun\_C\_Malesuada\_Dap\_Ibus\_R\_Isus\_Fr\_Ing\_Illa\_Commodo.\_Qu\_Isque\_Ne\_C\_Al\_Iquam\_Ex.\_Curab\_Itur\_S\_It\_Amet\_Ele\_Ifend\_Vel\_It.\_Fus\_Ce\_Turp\_Is\_N\_Isl,\_Sus\_C\_Ip\_It\_At\_Vest\_Ibulum\_Ne\_C,\_Hendrer\_It\_A\_Est.\_Integer\_Consequat\_Pharetra. |
| Sed\_aliquet\_vehicula\_nunc\_eget\_fauAibus.\_Proin\_molestie\_ipsum\_aA\_effiAitur\_euismod.\_Cras neA orNare quam. N\_llam e\_ rhoNA\_s mi. Phasell\_s Nisl estXsagittis vitae maxim\_s iN, imperdiet NeA q\_am. N\_NA s-t amet ma\_r-s e\_ vel-t aAA\_msaN grav-da. Cras f-N-b\_s \_llamAorper d-am vel p\_lv-Nar. Pro-N plaAerat N\_NA et lorem s\_sA-p-t al-q\_et. AeNeaN portt-tor od-o -d l-bero molest-eXaAA\_msaN lobort-s q\_am s\_sA-p-t. Sed tiNAid\_Nt met\_s dolorXvitae a\_Ator ligula auAtor eu. PraeseNt malesuadaXleAtus sit amet aliquet vehiculaXNisl NuNc coNsequat. | Sed\_Aliquet\_Vehicula\_Nunc\_Eget\_Fau\_Aibus.\_Proin\_Molestie\_Ipsum\_A\_A\_Effi\_Aitur\_Euismod.\_Cras ne\_A or\_Nare quam. \_N\_Llam e\_ rho\_N\_A\_S mi. \_Phasell\_S \_Nisl est\_Xsagittis vitae maxim\_S i\_N, imperdiet \_Ne\_A q\_Am. \_N\_N\_A s-t amet ma\_R-s e\_ vel-t a\_A\_A\_Msa\_N grav-da. \_Cras f-\_N-b\_S \_Llam\_Aorper d-am vel p\_Lv-\_Nar. \_Pro-\_N pla\_Aerat \_N\_N\_A et lorem s\_S\_A-p-t al-q\_Et. \_Ae\_Nea\_N portt-tor od-o -d l-bero molest-e\_Xa\_A\_A\_Msa\_N lobort-s q\_Am s\_S\_A-p-t. \_Sed ti\_N\_Aid\_Nt met\_S dolor\_Xvitae a\_Ator ligula au\_Ator eu. \_Praese\_Nt malesuada\_Xle\_Atus sit amet aliquet vehicula\_X\_Nisl \_Nu\_Nc co\_Nsequat. |
| WUEIFNWL\_DKQLJDMOWEIWKLD,MXIOLFJ\_MDWQIOWA\_KLSJDMX \_WIJLDK,XEWPS’;X.ZM COPWKD,/\_COLFSDAMXI\_LAK.SZX IMMWEOADJKAM \_XIC AIODJX ZM\_ISLHADJ/FJOE’F WQFSNA;/V;DISANV;E ABV\_WJ/;M | W\_U\_E\_I\_F\_N\_W\_L\_D\_K\_Q\_L\_J\_D\_M\_O\_W\_E\_I\_W\_K\_L\_D,\_M\_X\_I\_O\_L\_F\_J\_M\_D\_W\_Q\_I\_O\_W\_A\_K\_L\_S\_J\_D\_M\_X \_W\_I\_J\_L\_D\_K,\_X\_E\_W\_P\_S’;\_X.\_Z\_M \_C\_O\_P\_W\_K\_D,/\_C\_O\_L\_F\_S\_D\_A\_M\_X\_I\_L\_A\_K.\_S\_Z\_X \_I\_M\_M\_W\_E\_O\_A\_D\_J\_K\_A\_M \_X\_I\_C \_A\_I\_O\_D\_J\_X \_Z\_M\_I\_S\_L\_H\_A\_D\_J/\_F\_J\_O\_E’\_F \_W\_Q\_F\_S\_N\_A;/\_V;\_D\_I\_S\_A\_N\_V;\_E \_A\_B\_V\_W\_J/;\_M |

Batalha de Pokémon

F = int(input())

matches = {}

def finish\_him(T):

if matches[T][1]:

return matches[T][0]

else:

return 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] == 'FINAL':

print(finish\_him(inp[1]))

break

else:

T, F1, F2, W = inp

matches[T] = (W, False)

A utilização de função recursiva nesta questão, embora opcional, é preferível.

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 1  1 Natália  2 1 1 1  FINAL 2 | Natália |
| 2  1 Yurick  2 André  3 1 2 1  4 1 2 1  5 1 1 1  6 2 2 2  7 3 4 3  8 7 6 7  FINAL 8 | Yurick |
| 2  10 10  11 11  100 10 10 10  101 11 11 11  102 10 11 10  FINAL 100 | 10 |
| 10  1 1  2 2  3 3  4 4  5 5  6 6  7 7  8 8  9 9  10 10  11 10 11 10  FINAL 11 | 10 |
| 4  0 WhatsApp  1 FBMessenger  2 Telegram  3 Hangouts  100 0 2 2  101 0 1 1  102 0 3 3  103 1 2 2  104 1 3 1  105 2 3 3  106 102 104 102  107 101 3 101  108 2 105 105  109 108 106 108  FINAL 109 | Hangouts |
| 7  0 avatar  1 odin  2 belona  3 huan  4 thanatos  5 kyubii  6 nemesis  7 0 1 0  8 2 3 2  9 4 5 4  10 6 7 7  11 10 9 9  12 11 3 11  13 12 11 11  14 13 7 7  15 13 14 13  FINAL 15 | thanatos |
| 4  9 ubuntu  8 mac  7 windows  6 fedora  0 9 8 8  1 9 7 9  2 9 6 9  3 8 7 8  4 8 6 8  5 7 6 7  100 0 3 3  101 1 4 1  102 5 4 4  103 100 7 7  104 2 3 2  105 104 4 104  106 100 2 100  107 101 102 101  108 107 103 107  FINAL 108 | ubuntu |
| 5  -1 coragem  -2 teenTitans  -3 totallySpies  -4 gumble  -5 zym  0 -1 -3 -1  1 -2 -4 -4  2 -5 0 -5  3 -1 1 1  4 -2 0 -2  5 3 4 4  6 3 5 3  7 0 6 0  8 7 6 6  FINAL 8 | gumble |
| 3  10 brooklyn99  11 theAway  12 trueDetective  0 10 11 10  1 10 12 12  2 11 12 12  3 0 1 0  4 3 2 2  5 0 4 0  6 1 5 5  FINAL 6 | brooklyn99 |
| 7  0 AnupamaKundoo  1 Iman  2 MarthaKarua  3 AdaMariaIsasi-Diaz  4 ShirinEbadi  5 CarolShaw  6 FabianaClaudino  7 0 6 6  8 2 3 3  9 0 5 5  10 2 4 4  11 1 5 1  12 0 4 4  13 1 4 1  14 2 5 5  15 4 5 5  16 0 3 0  17 1 3 1  18 0 2 0  19 3 4 4  20 2 6 6  21 1 2 2  22 0 1 0  23 10 22 10  24 11 21 21  25 12 20 20  26 13 19 19  27 14 18 14  28 15 17 15  29 16 23 23  30 24 29 29  31 25 28 25  32 26 27 27  33 30 31 31  34 32 33 32  FINAL 34 | CarolShaw |

Cartésia

N = int(input())

maps = {}

for i in range(N):

S, X0, Y0, XF, YF = input().split()

maps[S] = (int(XF) - int(X0), int(YF) - int(Y0))

parties = input().split()

for party in parties:

print(maps[party][0], maps[party][1])

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0 |  |
| 3  dwedw 0 0 0 0  diowed 0 1 2 3  owdme 4 5 6 7 |  |
| 3  dwedw 0 0 0 0  diowed 0 1 2 3  owdme 4 5 6 7  dwedw diowed | 0 0  2 2 |
| 3  dwedw 0 0 0 0  diowed 10 41 22 13  owdme 1022 5 6 7  owdme diowed | -1016 2  12 -28 |
| 5  🤔 7 7 3 52  🙈 3405 18 -572 6  💩 10 -8 34 5  👌 -28 19 -28 19  🎉 74124 1041277 10010312 3111011  💩 🎉 💩 👌 💩 💩 🤔 | 24 13  9936188 2069734  24 13  0 0  24 13  24 13  -4 45 |
| 1  potato\_chips 28 28 28 28  potato\_chips | 0 0 |
| 1  potato\_chips 28 0 0 28  potato\_chips | -28 28 |
| 5  👸 0 0 23 23  potato\_chips 0 0 23 23  roberto 0 0 23 23  lalala 0 0 23 23  \_\_\_ 0 0 23 23  👸 | 23 23 |
| 4  tira 0 2 0 4  falando 0 2 0 7  prara 0 3 9 6  tirara 4 19 20 17  falando tira | 0 5  0 2 |
| 2  solucionador 111311 4111012 3 0  de\_problemas 12374 70 3117311123 11120  de\_problemas | 3117298749 11050 |

Deambulação

N = int(input())

maps = {}

for i in range(N):

S, X0, Y0, DX, DY = input().split()

maps[S] = (int(DX) + int(X0), int(DY) + int(Y0))

parties = input().split()

for party in parties:

print(maps[party][0], maps[party][1])

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0 |  |
| 3  dwedw 0 0 0 0  diowed 0 1 2 3  owdme 4 5 6 7 |  |
| 3  dwedw 0 0 0 0  diowed 0 1 2 3  owdme 4 5 6 7  dwedw diowed | 0 0  2 4 |
| 3  dwedw 0 0 0 0  diowed 10 41 22 13  owdme 1022 5 6 7  owdme diowed | 1028 12  32 54 |
| 5  🤔 7 7 3 52  🙈 3405 18 -572 6  💩 10 -8 34 5  👌 -28 19 -28 19  🎉 74124 1041277 10010312 3111011  💩 🎉 💩 👌 💩 💩 🤔 | 44 -3  10084436 4152288  44 -3  -56 38  44 -3  44 -3  10 59 |
| 1  potato\_chips 28 28 28 28  potato\_chips | 56 56 |
| 1  potato\_chips 28 0 0 28  potato\_chips | 28 28 |
| 5  👸 0 0 23 23  potato\_chips 0 0 23 23  roberto 0 0 23 23  lalala 0 0 23 23  \_\_\_ 0 0 23 23  👸 | 23 23 |
| 4  tira 0 2 0 4  falando 0 2 0 7  prara 0 3 9 6  tirara 4 19 20 17  falando tira | 0 9  0 6 |
| 2  solucionador 111311 4111012 3 0  de\_problemas 12374 70 3117311123 11120  de\_problemas | 3117323497 11190 |

Estado

N = int(input())

corrupt = {}

for i in range(N):

S = input().split()

corrupt[S[0]] = S[1:]

R = input().split()

# verifique o gabarito da questão F da lista 7

request = sorted(list(filter(lambda c: list(filter(lambda d: d in R, corrupt[c])), corrupt.keys())), key=str.lower)

print(' '.join(request))

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0  potato |  |
| 1  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  c | a |
| 2  a b c d e  e a  a | e |
| 2  y b c d e  d c c b a  d | y |
| 2  a b c d e g h  e d c b  c | a e |
| 3  j b c d e  k d c b a  i idew ddew djwiudw  d a | j k |
| 1  a b c d e  a |  |
| 7  oiwejowjdoiejiow wedew dioewjdiowjeio djwiodjiowjedoiwe wedew  iodwjoijewoj deiowjdoiwejdoiejw doijw eoidjw oiedj woidjow ijd oiwje idowje doi  dweijd wedew dwehdiwhudiw iuewhd uiweh  doeiwd iuwehd iwedhuiw heduiwhed ewdhw uidhwe wedew wedew  d ewudihweuid wiudw iuhduw ihd uwi hduiew wedew  dwiuh dwiuhduiwe duiweh duiwh duiwhe dw  duwi diuwh diuw  wedew | d doeiwd dweijd oiwejowjdoiejiow |
| 5  Nullam placerat ut nibh eget aliquam. Nunc et quam id sapien tempor sollicitudin. Suspendisse fringilla, mi vitae egestas laoreet, sem dui facilisis ipsum, vitae porta sapien tortor non lorem. Aenean eget sapien eget neque porta ultrices. In accumsan odio nunc, non rutrum ex eleifend vel. Integer finibus vehicula metus, sed malesuada magna. Duis non est mattis, congue ipsum eget, bibendum elit. Maecenas diam lacus, feugiat vulputate gravida eget, consequat vel erat. Ut elit diam, elementum sit amet posuere ac, maximus ac neque. Pellentesque lacinia mauris nec tincidunt viverra. Donec efficitur risus facilisis quam luctus, eget fringilla risus finibus. Fusce commodo mollis cursus. Cras urna nibh, bibendum eget tortor quis, semper imperdiet felis.  Etiam ultricies viverra ex, quis efficitur purus iaculis non. In at pharetra tortor. Nam id porta ipsum, vel interdum orci. Aenean egestas imperdiet varius. Cras dignissim scelerisque tortor non pellentesque. Curabitur tincidunt orci vitae hendrerit tempor. Curabitur at sem eros. Mauris non ullamcorper neque. Etiam viverra quis turpis in ultricies. Phasellus quis pretium elit, nec cursus nulla.  Nam nisi enim, placerat ac pretium vel, porttitor ut massa. Nunc volutpat mi mollis, commodo mauris ac, tristique sapien. Nulla vitae ultricies dolor. Aliquam cursus purus ligula, ac semper sem semper in. Quisque lobortis orci nec suscipit molestie. Ut tristique orci quis ipsum porttitor vestibulum. Integer gravida diam ut nisi auctor tincidunt. Nullam tristique condimentum suscipit. Vestibulum eu sodales sem, quis volutpat mi. Nunc sed gravida odio. Integer rutrum rutrum massa quis pretium. Aliquam non purus dolor. Nulla egestas, massa sit amet volutpat porta, ex urna consequat eros, sed lacinia velit felis in nibh. Phasellus vulputate risus ut auctor accumsan.  Pellentesque at tellus pulvinar, suscipit eros ac, lacinia dui. Mauris efficitur egestas turpis, nec tincidunt arcu auctor in. Vivamus sit amet hendrerit ipsum. Suspendisse potenti. Etiam ullamcorper sed odio eu volutpat. Nulla facilisi. Integer iaculis lacus vitae euismod tempus. Nulla sit amet risus eget orci pretium imperdiet.  Etiam quis nulla at justo ultrices auctor. Praesent sit amet posuere turpis. Phasellus tempor hendrerit lacus, vel cursus arcu pulvinar sit amet. Aenean risus quam, suscipit nec rutrum et, tempus accumsan odio. Pellentesque egestas libero dolor, ac laoreet metus tempor non. Praesent eu justo vitae lectus elementum posuere a blandit orci. Proin egestas maximus turpis at imperdiet. Morbi id consectetur lorem. Pellentesque nisi justo, vehicula et auctor sed, convallis ut enim. Suspendisse potenti. Praesent lacus eros, fermentum ut pharetra non, feugiat et orci. Phasellus facilisis elit sed porttitor venenatis. Cras lacinia venenatis dolor, sit amet consectetur tellus tempus eget. Sed justo sem, hendrerit tempus lacus vel, ultricies finibus neque. Nunc molestie turpis id aliquam semper. Suspendisse potenti.  aliquam ut enim | Etiam Nam Nullam |
| 40  Seda at volutpat en im.  Sed non sollicitudin di am.  Vivamus turpis turpis, fa cilisis  eu gravida e u, volutpat  eta nibh. Aenean fauc ibus  turpis in met us volutpat  a malesuada libe ro pellentesque.  Morbi dictum sap ien ut  finibusa consequat. Nu lla eget  rutrum tellus, ac lao reet  magna. Pellentesque a lo rem  quam. Integer rutrum scele risque  eros vitae ornare. a t.  Mauris ullamcorper frin gilla diam.  Cras non nisl con gue,  finibus ex in, pul vinar  ipsum. Vestibulum lec tus est,  pellentesque sit am et varius  in faucibus orci lu ctus  et ultrices posu ere cubilia  Curae; Fusce sceler isque ipsum  quis tempus sollic itudin. Pellentesque  eget odio com modo, lobortis  tellus sed, vesti bulum orci.  Pellentesque pu rus quam, tincidunt  ut varius quis, max imus  at tortor. Pellentesque e t  egestas risus, sit am et  ullamcorper ri sus. Fusce tincidunt  arcu eget eros. Pra esent  id diam male suada, eleifend  odio quis, cons equat nisi.  Aenean port titor enim id  justo congue orna re. Fusce  auctor ipsum lib ero, lobortis  molestie metus fau cibus in.  Proin ut sod ales elit,  sit amet acc umsan est.  Nullam condim entum quam eget  purus tristique aliq uet. Vivamus  volutpat tristique Pellentesque | at eu magna. purus quis Seda turpis |

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)

A utilização de função recursiva nesta questão, embora opcional, é preferível.

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 1  1 Natália  2 1 1 1  FINISHHIM 2 | 1 |
| 2  1 Yurick  2 André  3 1 2 1  4 1 2 1  5 1 1 1  6 2 2 2  7 3 4 3  8 7 6 7  FINISHHIM 8 | 3 |
| 2  10 10  11 11  100 10 10 10  101 11 11 11  102 10 11 10  FINISHHIM 100 | 1 |
| 10  1 1  2 2  3 3  4 4  5 5  6 6  7 7  8 8  9 9  10 10  11 10 11 10  FINISHHIM 11 | 1 |
| 4  0 WhatsApp  1 FBMessenger  2 Telegram  3 Hangouts  100 0 2 2  101 0 1 1  102 0 3 3  103 1 2 2  104 1 3 1  105 2 3 3  106 102 104 102  107 101 3 101  108 2 105 105  109 108 106 108  FINISHHIM 109 | 3 |
| 7  0 avatar  1 odin  2 belona  3 huan  4 thanatos  5 kyubii  6 nemesis  7 0 1 0  8 2 3 2  9 4 5 4  10 6 7 7  11 10 9 9  12 11 3 11  13 12 11 11  14 13 7 7  15 13 14 13  FINISHHIM 15 | 4 |
| 4  9 ubuntu  8 mac  7 windows  6 fedora  0 9 8 8  1 9 7 9  2 9 6 9  3 8 7 8  4 8 6 8  5 7 6 7  100 0 3 3  101 1 4 1  102 5 4 4  103 100 7 7  104 2 3 2  105 104 4 104  106 100 2 100  107 101 102 101  108 107 103 107  FINISHHIM 108 | 4 |
| 5  -1 coragem  -2 teenTitans  -3 totallySpies  -4 gumble  -5 zym  0 -1 -3 -1  1 -2 -4 -4  2 -5 0 -5  3 -1 1 1  4 -2 0 -2  5 3 4 4  6 3 5 3  7 0 6 0  8 7 6 6  FINISHHIM 8 | 4 |
| 3  10 brooklyn99  11 theAway  12 trueDetective  0 10 11 10  1 10 12 12  2 11 12 12  3 0 1 0  4 3 2 2  5 0 4 0  6 1 5 5  FINISHHIM 6 | 3 |
| 7  0 AnupamaKundoo  1 Iman  2 MarthaKarua  3 AdaMariaIsasi-Diaz  4 ShirinEbadi  5 CarolShaw  6 FabianaClaudino  7 0 6 6  8 2 3 3  9 0 5 5  10 2 4 4  11 1 5 1  12 0 4 4  13 1 4 1  14 2 5 5  15 4 5 5  16 0 3 0  17 1 3 1  18 0 2 0  19 3 4 4  20 2 6 6  21 1 2 2  22 0 1 0  23 10 22 10  24 11 21 21  25 12 20 20  26 13 19 19  27 14 18 14  28 15 17 15  29 16 23 23  30 24 29 29  31 25 28 25  32 26 27 27  33 30 31 31  34 32 33 32  FINISHHIM 34 | 4 |

Game Design

N = int(input())

features = []

for i in range(N):

features.append(input())

M = int(input())

lesser = []

for i in range(M):

model = input()

# esta é a melhor forma de resolver esse problema

# mas eu não sei se `enumerate` chegou a ser apresentado em sala de aula

# failed aqui é uma array de índices de falhas

failed = [index for (index, x) in enumerate(model) if x == "O"]

# uma solução bem menos "pythonica", e que supõe que o aluno

# não sabe como usar índice no loop seria:

index = 0

failed = []

for c in model:

if c == 'O':

failed.append(index)

index += 1

# fim da solução pior

if len(failed) > len(lesser):

lesser = failed

for index in lesser:

print(features[index])

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0  0 |  |
| 4  dioew deiodew ide  idwojew jdiowjed  fmrelm jfoe  foierjoer fjeiofj eiofjeor  5  XXXX  OXOX  XXXO  OXOX  OOXO | dioew deiodew ide  idwojew jdiowjed  foierjoer fjeiofj eiofjeor |
| 7  at ex eu cursus. Proin at  r in maximus mau  ulvinar sit amet erat nec, aliqu  s varius est eget e  ces. Aliquam im  pus efficitur, volutpat sit a  retra bl  10  XXXXXXX  XXXXXXX  XXXOXXX  XOXXXXX  XXXXXOX  XXXXXOO  OXXXXXO  OXXXXOX  OXXXXXX  OOOOOOO | at ex eu cursus. Proin at  r in maximus mau  ulvinar sit amet erat nec, aliqu  s varius est eget e  ces. Aliquam im  pus efficitur, volutpat sit a  retra bl |
| 5  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  5  XXXXX  XXXXX  XXOOO  OOXXX  XOXOX | repeated features might be allowed  repeated features might be allowed  repeated features might be allowed |
| 5  doesn’t matter if we have  lots of features described  when we don’t have  any model whatsoever  to test  0 |  |
| 3  repeated models  might be allowed  as long as they are not the answer  7  XOO  OOO  XOO  OOX  XOO  OXO  XOO | repeated models  might be allowed  as long as they are not the answer |
| 10  I  just happen  to be lazy  when writing  down  some use case  tests  so please  let me go  already  11  XXXXXXXXXX  XXXXXXXXXX  XXXXOXXXXX  XXXXXXXXXX  XXXXXXXXXX  XOXXXXOXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXOXXXX  OXXXXXXXXX  XXXXXXXXXX | just happen  tests |
| 3  A  B  C  4  XOX  OXO  XXO  OXX | A  C |
| 4  X  Y  Z  W  12  XXXX  XXXO  XXOX  XOXX  OXXX  XXOO  XOXO  OXXO  XOOX  OXOX  OOXX  OOOX | X  Y  Z |
| 5  Q  W  E  R  T  22  XXXXX  OXXXX  XOXXX  XXOXX  XXXOX  XXXXO  OOXXX  OXOXX  OXOOO  OXXOX  OXXXO  XOOXX  XOXOX  XOXXO  XXOOX  XXOXO  XXXOO  OOOXX  OOXOX  OOXXO  OXOOX  OXOXO | Q  E  R  T |

Horário

N = int(input())

def parseHour(str):

return int(str.replace(":", ""))

workers = {}

for i in range(N):

S = input().split()

workers[S[0]] = [parseHour(x) for x in S[1:]]

R = parseHour(input())

request = []

for worker in sorted(workers, key=str.lower):

if workers[worker][0] <= R <= workers[worker][1]:

request.append(worker)

print(' '.join(request))

Embora seja perfeitamente possível ir imprimindo à medida que se vai encontrando os valores de ‘worker’, é preferível imprimir ao fim para que não fique sobrando um espaço vazio desnecessário ao final da linha.

A função parseHour também é perfeitamente desnecessária neste caso, porém deixa o código mais legível e, portanto, fácil de se compreender.

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0  00:01 |  |
| 1  a 00:10 23:59  00:10 | a |
| 2  a 01:00 02:00  e 03:30 23:00  05:00 | e |
| 2  y 04:00 06:00  d 01:00 03:00  05:00 | y |
| 2  a 01:00 12:00  e 11:00 23:00  11:25 | a e |
| 3  j 22:00 23:00  k 20:00 23:00  i 10:00 22:00  23:00 | j k |
| 1  a 10:00 11:00  09:00 |  |
| 7  oiwejowjdoiejiow 05:00 12:00  iodwjoijewoj 03:00 03:30  dweijd 05:00 06:28  doeiwd 05:00 22:00  d 05:00 14:00  dwiuh 01:00 03:20  duwi 03:00 03:25  05:00 | d doeiwd dweijd oiwejowjdoiejiow |
| 5  Nullam 04:00 22:00  Etiam 18:00 23:00  Nam 07:00 18:00  Pellentesque 12:00 12:13  Etiam 13:00 23:00  18:00 | Etiam Nam Nullam |
| 40  Seda 10:22 16:00  Sed 01:00 01:36  Vivamus 02:00 02:45  eu 09:19 15:58  eta 23:00 23:45  turpis 08:00 14:00  a 14:01 14:02  Morbi 13:58 13:59  finibusa 11:00 11:15  rutrum 09:00 13:56  magna. 13:59 22:01  quam. 02:00 02:30  eros 03:00 03:46  Mauris 22:19 22:22  Cras 19:00 23:50  finibus 14:01 14:02  ipsum. 12:00 13:45  pellentesque 01:00 13:58  in 07:00 08:00  et 08:00 09:00  Curae; 09:00 09:30  quis 13:00 21:00  eget 10:00 13:00  tellus 18:56 19:24  Pellentesque 19:11 22:11  ut 23:43 23:45  at 12:00 20:00  egestas 15:51 25:52  ullamcorper 18:39 19:39  arcu 11:09 11:11  id 03:30 04:40  odio 04:40 05:50  Aenean 05:50 06:00  justo 13:28 13:37  auctor 13:37 13:59  molestie 14:01 14:14  Proin 14:02 14:28  sit 23:44 23:59  Nullam 11:00 13:00  purus 11:00 19:00  14:00 | at eu magna. purus quis Seda turpis |

Interland

N = int(input())

features = []

for i in range(N):

features.append(input())

M = int(input())

lesser = []

for i in range(M):

site = input()

# esta é a melhor forma de resolver esse problema

# mas eu não sei se `enumerate` chegou a ser apresentado em sala de aula

# failed aqui é uma array de índices de falhas

failed = [index for (index, x) in enumerate(site) if x == "O"]

# uma solução bem menos "pythonica", e que supõe que o aluno

# não sabe como usar índice no loop seria:

index = 0

failed = []

for c in site:

if c == 'O':

failed.append(index)

index += 1

# fim da solução pior

if len(failed) > len(lesser):

lesser = failed

for index in lesser:

print(features[index])

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| 0  0 |  |
| 4  dioew deiodew ide  idwojew jdiowjed  fmrelm jfoe  foierjoer fjeiofj eiofjeor  5  XXXX  OXOX  XXXO  OXOX  OOXO | dioew deiodew ide  idwojew jdiowjed  foierjoer fjeiofj eiofjeor |
| 7  at ex eu cursus. Proin at  r in maximus mau  ulvinar sit amet erat nec, aliqu  s varius est eget e  ces. Aliquam im  pus efficitur, volutpat sit a  retra bl  10  XXXXXXX  XXXXXXX  XXXOXXX  XOXXXXX  XXXXXOX  XXXXXOO  OXXXXXO  OXXXXOX  OXXXXXX  OOOOOOO | at ex eu cursus. Proin at  r in maximus mau  ulvinar sit amet erat nec, aliqu  s varius est eget e  ces. Aliquam im  pus efficitur, volutpat sit a  retra bl |
| 5  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  repeated features might be allowed  5  XXXXX  XXXXX  XXOOO  OOXXX  XOXOX | repeated features might be allowed  repeated features might be allowed  repeated features might be allowed |
| 5  doesn’t matter if we have  lots of features described  when we don’t have  any model whatsoever  to test  0 |  |
| 3  repeated models  might be allowed  as long as they are not the answer  7  XOO  OOO  XOO  OOX  XOO  OXO  XOO | repeated models  might be allowed  as long as they are not the answer |
| 10  I  just happen  to be lazy  when writing  down  some use case  tests  so please  let me go  already  11  XXXXXXXXXX  XXXXXXXXXX  XXXXOXXXXX  XXXXXXXXXX  XXXXXXXXXX  XOXXXXOXXX  XXXXXXXXXX  XXXXXXXXXX  XXXXXOXXXX  OXXXXXXXXX  XXXXXXXXXX | just happen  tests |
| 3  A  B  C  4  XOX  OXO  XXO  OXX | A  C |
| 4  X  Y  Z  W  12  XXXX  XXXO  XXOX  XOXX  OXXX  XXOO  XOXO  OXXO  XOOX  OXOX  OOXX  OOOX | X  Y  Z |
| 5  Q  W  E  R  T  22  XXXXX  OXXXX  XOXXX  XXOXX  XXXOX  XXXXO  OOXXX  OXOXX  OXOOO  OXXOX  OXXXO  XOOXX  XOXOX  XOXXO  XXOOX  XXOXO  XXXOO  OOOXX  OOXOX  OOXXO  OXOOX  OXOXO | Q  E  R  T |

Javascript

mix = input()

new\_mix = ""

for c in mix:

if c.isupper():

new\_mix += "\_"

new\_mix += c.lower()

splitted = new\_mix.split("\_")

print("-".join([x for x in splitted if x != ""]))

# você também pode fazer algo como

# while "" in splitted: splitted.remove("")

# para remover os - repetidos :)

Poderíamos usar regex ao invés do loop, mas acredito que este recurso esteja fora do escopo desta matéria.

| Instâncias de Entrada | Instâncias de Saída |
| --- | --- |
| simple\_test\_case\_with\_snake\_case\_only | simple-test-case-with-snake-case-only |
| NotSoSimpleTestCaseWithCamelCaseOnly | not-so-simple-test-case-with-camel-case-only |
| simpleMix\_of\_snakeAndCamel\_case\_haHa | simple-mix-of-snake-and-camel-case-ha-ha |
| THIS\_IS\_SO\_MUCH\_FUN | t-h-i-s-i-s-s-o-m-u-c-h-f-u-n |
| what\_is\_love\_BabyDon’tHurtMe\_Don’tHurtMe\_NoMore | what-is-love-baby-don’t-hurt-me-don’t-hurt-me-no-more |
| eu-selecionei-o-preto\_eu\_selecionei\_o\_azulEEuSelecioneiOAmarelo | eu-selecionei-o-preto-eu-selecionei-o-azul-e-eu-selecionei-o-amarelo |
| moar\_one\_type\_of\_case\_testing | moar-one-type-of-case-testing |
| QuIsque\_susCIpIt\_Cursus\_nIsI,\_ultrICes\_rutrum\_duI\_pulvInar\_a.\_AlIquam\_ante\_lorem,\_sCelerIsque\_In\_justo\_vItae,\_fInIbus\_laCInIa\_sapIen.\_Pellentesque\_ut\_neque\_eu\_odIo\_eleIfend\_molestIe\_non\_non\_maurIs.\_Nulla\_venenatIs\_laCInIa\_augue\_sed\_trIstIque.\_ProIn\_eget\_lIgula\_vel\_neque\_ultrICIes\_sagIttIs.\_ProIn\_et\_quam\_laCInIa\_Ipsum\_bIbendum\_porttItor\_In\_ornare\_massa.\_NunC\_malesuada\_dapIbus\_rIsus\_frIngIlla\_Commodo.\_QuIsque\_neC\_alIquam\_ex.\_CurabItur\_sIt\_amet\_eleIfend\_velIt.\_FusCe\_turpIs\_nIsl,\_susCIpIt\_at\_vestIbulum\_neC,\_hendrerIt\_a\_est.\_Integer\_Consequat\_pharetra. | qu-isque-sus-c-ip-it-cursus-n-is-i,-ultr-i-ces-rutrum-du-i-pulv-inar-a.-al-iquam-ante-lorem,-s-celer-isque-in-justo-v-itae,-f-in-ibus-la-c-in-ia-sap-ien.-pellentesque-ut-neque-eu-od-io-ele-ifend-molest-ie-non-non-maur-is.-nulla-venenat-is-la-c-in-ia-augue-sed-tr-ist-ique.-pro-in-eget-l-igula-vel-neque-ultr-i-c-ies-sag-itt-is.-pro-in-et-quam-la-c-in-ia-ipsum-b-ibendum-portt-itor-in-ornare-massa.-nun-c-malesuada-dap-ibus-r-isus-fr-ing-illa-commodo.-qu-isque-ne-c-al-iquam-ex.-curab-itur-s-it-amet-ele-ifend-vel-it.-fus-ce-turp-is-n-isl,-sus-c-ip-it-at-vest-ibulum-ne-c,-hendrer-it-a-est.-integer-consequat-pharetra. |
| Sed\_aliquet\_vehicula\_nunc\_eget\_fauAibus.\_Proin\_molestie\_ipsum\_aA\_effiAitur\_euismod.\_Cras neA orNare quam. N\_llam e\_ rhoNA\_s mi. Phasell\_s Nisl estXsagittis vitae maxim\_s iN, imperdiet NeA q\_am. N\_NA s-t amet ma\_r-s e\_ vel-t aAA\_msaN grav-da. Cras f-N-b\_s \_llamAorper d-am vel p\_lv-Nar. Pro-N plaAerat N\_NA et lorem s\_sA-p-t al-q\_et. AeNeaN portt-tor od-o -d l-bero molest-eXaAA\_msaN lobort-s q\_am s\_sA-p-t. Sed tiNAid\_Nt met\_s dolorXvitae a\_Ator ligula auAtor eu. PraeseNt malesuadaXleAtus sit amet aliquet vehiculaXNisl NuNc coNsequat. | sed-aliquet-vehicula-nunc-eget-fau-aibus.-proin-molestie-ipsum-a-a-effi-aitur-euismod.-cras ne-a or-nare quam. -n-llam e- rho-n-a-s mi. -phasell-s -nisl est-xsagittis vitae maxim-s i-n, imperdiet -ne-a q-am. -n-n-a s-t amet ma-r-s e- vel-t a-a-a-msa-n grav-da. -cras f--n-b-s -llam-aorper d-am vel p-lv--nar. -pro--n pla-aerat -n-n-a et lorem s-s-a-p-t al-q-et. -ae-nea-n portt-tor od-o -d l-bero molest-e-xa-a-a-msa-n lobort-s q-am s-s-a-p-t. -sed ti-n-aid-nt met-s dolor-xvitae a-ator ligula au-ator eu. -praese-nt malesuada-xle-atus sit amet aliquet vehicula-x-nisl -nu-nc co-nsequat. |
| WUEIFNWL\_DKQLJDMOWEIWKLD,MXIOLFJ\_MDWQIOWA\_KLSJDMX \_WIJLDK,XEWPS’;X.ZM COPWKD,/\_COLFSDAMXI\_LAK.SZX IMMWEOADJKAM \_XIC AIODJX ZM\_ISLHADJ/FJOE’F WQFSNA;/V;DISANV;E ABV\_WJ/;M | w-u-e-i-f-n-w-l-d-k-q-l-j-d-m-o-w-e-i-w-k-l-d,-m-x-i-o-l-f-j-m-d-w-q-i-o-w-a-k-l-s-j-d-m-x -w-i-j-l-d-k,-x-e-w-p-s’;-x.-z-m -c-o-p-w-k-d,/-c-o-l-f-s-d-a-m-x-i-l-a-k.-s-z-x -i-m-m-w-e-o-a-d-j-k-a-m -x-i-c -a-i-o-d-j-x -z-m-i-s-l-h-a-d-j/-f-j-o-e’-f -w-q-f-s-n-a;/-v;-d-i-s-a-n-v;-e -a-b-v-w-j/;-m |