



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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|--|
| <p>Qulsque_susCiplt_Cursus_nIsI,_ultrICes_rutrum_dul_pulvInar_a._Allquam_ante_lorem,_sCelersIque_In_justo_vItae,_fInIbus_laCInla_saplen._Pellentesque_ut_neque_eu_odlo_eleIfend_molestle_non_non_maurIs._Nulla_venenatIs_laCInla_augue_sed_trIstIque._ProIn_eget_Ilgula_vel_neque_ultrICes_sagIttIs._ProIn_et_quam_laCInla_IpsuM_bIbendum_porttItor_In_ornare_massa._NunC_malesuada_dapIbus_rIsus_frlngIll_a_Commodo._Qulsque_neC_allquam_ex._CurabItur_sIt_amet_eleIfend_vellIt._FusCe_turpIs_nIsI,_susCiplt_at_vestIbulum_neC,_hendrerIt_a_est._Integer_Consequat_pharetra.</p> | <p>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_la_Sap_Ien._Pellentesque_Ut_Neque_Eu_Od_Io_Ele_Ifend_Molest_Ie_Non_Non_Maur_Is._Nulla_Venenat_Is_La_C_In_la_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_la_IpsuM_B_Ibendum_Portt_Itor_In_Ornare_Massa._Nun_C_Malesuada_Dap_Ibus_R_Isus_Fr_Ing_Illa_Commодо._Qu_Isque_Ne_C_Al_Iquam_Ex._Curab_Itur_S_It_Amet_Ele_Ifend_Vel_It._Fus_Ce_Turp_Is_N_IsI,_Sus_C_Ip_It_At_Vest_Ibulum_Ne_C,_Hendrer_It_A_Est._Integer_Consequat_Pharetra.</p> |
| <p>Sed_aliquet_vehicula_nunc_eget_fauAibus._Proin_molestie_ipsum_aA_effiAitur_euismod._Cras_neA_orNare_quam._N_lIam_e_rhoNA_s_mi._Phasell_s_NIsI_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_lIamAorper_d-am_vel_p_lv-Nar._Pro-N_plaAerat_N_NA_et_lorem_s_sA-p-t_al-q_et._AeNeaN_porttitor_od-o_-d_l-bero_molest-eXaAA_msaN_lobort-s_q_am_s_sA-p-t._Sed_tI_nAid_Nt_met_s_dolorXvitae_a_Ator_ligula_auAtor_eu._PraeseNt_malesuadaXleAtus_sit_amet_aliquet_vehiculaXNIsI_NuNc_coNsequat.</p> | <p>Sed_Aliquet_Vehicula_Nunc_Eget_Fau_Aibus._Proin_Molestie_IpsuM_A_A_Effi_Aitur_Euismod._Cras_ne_A_or_Nare_quam._N_lIam_e_rho_N_A_S_mi._Phasell_S_NIsI_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_lIam_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_porttitor_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_NIsI_Nu_Nc_co_Nsequat.</p> |
| <p>WUEIFNWL_DKQLJDMOWEIWKLD,MXIOLFJ_MDWQIOWA_KLSJDMX _WIJLDK,XEWPS';X.ZM COPWKD,/ _COLFSDAMXI_LAK.SZX IMMWEODJKA _XIC AIODJX ZM_ISLHADJ/FJOE'F WQFSNA;/V;DISANV;E ABV_WJ;/M</p> | <p>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</p> |

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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|-----------------------|---------------------|
| 7 | CarolShaw |
| 0 AnupamaKundoo | |
| 1 Iman | |
| 2 MarthaKarua | |
| 3 AdaMarialsasi-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 | |

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 |

| Instâncias de Entrada | Instâncias de Saída |
|--|---------------------|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|--|---------------------|
| 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 i d e w d d e w d j w i u d w d a | j k |
| 1 a b c d e a | |

| Instâncias de Entrada | Instâncias de Saída |
|--|---------------------------------------|
| <p>7 oiwejoyjdoiejow wedew dioewjdiowjeio djwiodjowjedoiwe wedew iodwjoijewoj deiwjdoiwejdziejw doijw eoidjw oiedj woidjow ijd oiwe idowje doi dweijd wedew dwehdiwhudiw iuewhd uiweh doeiwd iuewhd iwedhuiw heduiwhed ewdhw uidhwe wedew wedew d ewudihweuid wiudw iuhduw ihd uwi hduiew wedew dwih dwihduiwe duiweh duiwh duiwhe dw duwi diuw diuw wedew</p> | <p>d doiwd dweijd oiwejoyjdoiejow</p> |

| Instâncias de Entrada | Instâncias de Saída |
|---|-------------------------|
| <p>5</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Pellentesque at tellus pulvinar, suscipit eros ac, lacinia dui. Mauris efficitur egestas turpis, nec tincidunt arcu auctor in. Vivamus sit amet hendrerit ipsum. Suspendisse</p> | <p>Etiam Nam Nullam</p> |

| Instâncias de Entrada | Instâncias de Saída |
|---|--|
| <p>40</p> <p>Seda at volutpat en im. Sed non sollicitudin di am. Vivamus turpis turpis, fa cilisis eu gravida e u, volutpat eta nibh. Aenean faucibus turpis in metus volutpat a malesuada liber pellentesque. Morbi dictum sapien ut finibusa consequat. Nulla eget rutrum tellus, ac laoreet magna. Pellentesque a lorem quam. Integer rutrum scelerisque eros vitae ornare. at. Mauris ullamcorper fringilla diam. Cras non nisl congue, finibus ex in, pulvinar ipsum. Vestibulum lectus est, pellentesque sit amet varius in faucibus orci luctus et ultrices posuere cubilia Curae; Fusce scelerisque ipsum quis tempus sollicitudin. Pellentesque eget odio conmodo, lobortis tellus sed, vestibulum orci. Pellentesque purus quam, tincidunt ut varius quis, maximus at tortor. Pellentesque et egestas risus, sit amet 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 volutpat tristique Pellentesque</p> | <p>at eu magna. purus quis Seda turpis</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)
```

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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|-----------------------|---------------------|
| 7 | 4 |
| 0 AnupamaKundoo | |
| 1 Iman | |
| 2 MarthaKarua | |
| 3 AdaMarialsasi-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 | |

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 ==
"0"]

    # 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 == '0':
            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 | |

| Instâncias de Entrada | Instâncias de Saída |
|--|---|
| <p>4 dioew deiodew ide idwojew jdiowjed fmrelm jfoe foierjoer fjeiofj eiofjeor</p> <p>5 XXXX OXOX XXOX OXOX OOXO</p> | <p>dioew deiodew ide idwojew jdiowjed foierjoer fjeiofj eiofjeor</p> |
| <p>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</p> <p>10 XXXXXXXX XXXXXXXX XXX0XXX X0XXXXX XXXXXX0X XXXXXX00 0XXXXXX0 0XXXX0X 0XXXXXX 0000000</p> | <p>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</p> |
| <p>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</p> <p>5 XXXXX XXXXX XXOOO OOXXX XOXOX</p> | <p>repeated features might be allowed repeated features might be allowed repeated features might be allowed</p> |

| Instâncias de Entrada | Instâncias de Saída |
|---|---|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 3 A B C 4 XOX OXO XXO OXX | A C |
| 4 X Y Z W 12 XXXX XXX0 XX0X X0XX OXXX XX00 X0X0 0XX0 X00X 0X0X 00XX 000X | X Y Z |

| Instâncias de Entrada | Instâncias de Saída |
|--|---------------------|
| 5 Q W E R T 22 XXXXX OXXXX X0XXX XX0XX XXX0X XXXX0 00XXX 0X0XX 0X000 0XX0X 0XXX0 X00XX X0X0X X0XX0 XX00X XX0X0 XXX00 000XX 00X0X 00XX0 0X00X 0X0X0 | 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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|--------------------------------|
| 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 oiwejoyjdoiejow 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 doiwd dweijd oiwejoyjdoiejow |
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|--|-------------------------------------|
| 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 == "0"]

    # 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 == '0':
            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 | |

| Instâncias de Entrada | Instâncias de Saída |
|--|---|
| <p>4 dioew deiodew ide idwojew jdiowjed fmrelm jfoe foierjoer fjeiofj eiofjeor</p> <p>5 XXXX OXOX XXOX OXOX OOXO</p> | <p>dioew deiodew ide idwojew jdiowjed foierjoer fjeiofj eiofjeor</p> |
| <p>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</p> <p>10 XXXXXXXX XXXXXXXX XXX0XXX X0XXXXX XXXXXX0X XXXXXX00 0XXXXXX0 0XXXX0X 0XXXXXX 0000000</p> | <p>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</p> |
| <p>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</p> <p>5 XXXXX XXXXX XXOOO OOXXX XOXOX</p> | <p>repeated features might be allowed repeated features might be allowed repeated features might be allowed</p> |

| Instâncias de Entrada | Instâncias de Saída |
|---|---|
| 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 |

| Instâncias de Entrada | Instâncias de Saída |
|---|---------------------|
| 3 A B C 4 XOX OXO XXO OXX | A C |
| 4 X Y Z W 12 XXXX XXX0 XX0X X0XX OXXX XX00 X0X0 0XX0 X00X 0X0X 00XX 000X | X Y Z |

| Instâncias de Entrada | Instâncias de Saída |
|--|---------------------|
| 5 Q W E R T 22 XXXXX OXXXX X0XXX XX0XX XXX0X XXXX0 00XXX 0X0XX 0X000 0XX0X 0XXX0 X00XX X0X0X X0XX0 XX00X XX0X0 XXX00 000XX 00X0X 00XX0 0X00X 0X0X0 | 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 |
| Qulsque_susCiplt_Cursus_nisl_ultrIces_rutrum_dul_pulvinar_a._Allquam_ante_lorem,_sCelerisque_In_justo_vitae,_fInIbus_laCInla_saplen._Pellentesque_ut_neque_eu_odlo_eleifend_molestie_non_non_mauris._Nulla_venenatis_laCInla_augue_sed_trIstIque._ProIn_eget_Ilgula_vel_neque_ultrIces_sagIttIs._ProIn_et_quam_laCInla_ipsum_bIbendum_porttItor_In_ornare_massa._NunC_malesuada_dapIbus_rIsus_frlngIll_a_Commodo._Qulsque_neC_allquam_ex._Curabitur_sIt_amet_eleifend_vellIt._FusCe_turpIs_nisl,_susCiplt_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. |

| Instâncias de Entrada | Instâncias de Saída |
|---|---|
| <p>Sed_aliquet_vehicula_nunc_eget_fauAibus._Pr oin_molestie_ipsum_aA_effiAitur_euismod._Cr as 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 auAator eu. PraeseNt malesuadaXleAtus sit amet aliquet vehiculaXNisl NuNc coNsequat.</p> | <p>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-ne-a-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-atu sit amet aliquet vehicula-x-nisl -nu-nc co-nsequat.</p> |
| <p>WUEIFNWL_DKQLJDMOWEIWKLD,MXIOLFJ _MDWQIOWA_KLSJDMX _WIJLDK,XEWPS';X.ZM COPWKD,/ _COLFSDAMXI_LAK.SZX IMMWEODJJKAM _XIC AIODJX ZM_ISLHADJ/FJOE'F WQFSNA;/V;DISANV;E ABV_WJ/;M</p> | <p>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</p> |