



## Introdução à Ciência da Computação - 113913

### 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, 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 questões estão em **ordem de dificuldade**. Esta lista possui **10 exercícios**, cada questão possui uma indicação para o tema da questão, tuplas ou dicionários. Isso deve facilitar a resolução de cada uma delas, porém cada questão pode ser resolvida da forma que você achar melhor/mais conveniente.
- Assim como as listas, as provas devem ser feitas na versão Python 3 ou superior.
- Leia com atenção e faça **exatamente** o que está sendo pedido.

## Questão A - Borderline (Dicionários)

O Transtorno de Personalidade Limítrofe (Borderline Personality Disorder, em inglês), é um transtorno de personalidade que se caracteriza pela abrupta mudança de comportamento, impulsividade e instabilidade de afetos, e é muitas vezes confundida com bipolaridade.

Joana tem uma variante de borderline tão rara que só existe em questões de listas de exercícios. Ela tem um conjunto de palavras críticas que desencadeiam uma alteração entre seus arquétipos de personalidades.

Tristes por usar essas palavras sem querer durante conversas do dia-a-dia e ver sua amiga explodindo com eles, os amigos de Joana pediram a você que escrevesse um programa que ajudasse-os a melhor controlar a situação.

### Entrada

A primeira linha da entrada consiste de um inteiro **N**, o número de palavras críticas para Joana. As próximas **N** linhas contém, cada uma, duas strings **T** e **P** separadas por espaço, a palavra crítica e a personalidade que ela desencadeia, respectivamente.

A última linha da entrada consiste de uma string, a frase que os amigos de Joana querem falar. Considere que os sinais de pontuação estarão sempre isolados com espaços.

### Saída

Seu programa deve imprimir uma única linha contendo “**Tudo bem!**” caso a frase não contenha nenhuma palavra crítica, ou os nomes das personalidades que serão desencadeadas, separadas por espaço, na ordem de input.

Exemplo de Entrada	Exemplo de Saída
3 boldo explosiva tampa chorosa beijo calorosa Você pôs a tampa no boldo ?	chorosa explosiva
2 dedos animada tinta criativa Quantos irmãos você tem , mesmo ?	Tudo bem!

Tabela 1: Questão A

## Questão B - Depita (Dicionários)

Reginalda quer abrir uma startup revolucionária no Vale do Silício, a Depita, que irá alugar depósitos de marmitas. É uma tecnologia revolucionária: quando você chega para trabalhar, pela manhã, deposita seu almoço em uma Depita™, e saca só mais tarde, quando for comer.

Porém, Reginalda não tem muita experiência desenvolvendo, então decidiu terceirizar a parte de software da sua nova empresa. Você, sabendo da incrível oportunidade de poder revolucionar a indústria, logo se prontificou para resolver o problema de Reginalda pela modesta quantia de 3 mil dólares.

### Entrada

A primeira linha da entrada consiste de um inteiro **N**, o número de iterações subsequentes. As próximas **N** linhas contêm, cada uma, uma string sem espaços **S**, o nome do depositante, e uma string que pode conter espaços **D**, a descrição da marmita depositada.

Se o mesmo depositante realizar o depósito múltiplas vezes, somente a última instância deve ser contabilizada, pois a Depita™ é programada para jogar fora a marmita anterior.

### Saída

A primeira linha da saída deve conter um inteiro **J**, o número de depositantes diferentes. As próximas **J** linhas devem conter, cada uma, a descrição de cada uma das marmitas depositadas, ordenadas em ordem alfabética do nome do depositante.

Exemplo de Entrada	Exemplo de Saída
3 João Bife do oião Ricardo Salsicha recheada com queijo João Cuscuz com leite	2 Cuzcuz com leite Salsicha recheada com queijo
2 Zé abacates Alberto sashimi à passarinho	2 sashimi à passarinho abacates

Tabela 2: Questão B

## Questão C - Deus Ex Machina (Tuplas)

*Deus ex Machina* é um artifício de escrita utilizado principalmente quando o escritor não sabe como resolver um problema que ele mesmo criou. Ele introduz um elemento que, convenientemente, guia o protagonista para a solução do problema. Um clássico exemplo de uso extensivo de Deus ex Machina é o Mestre dos Magos, no desenho Caverna do Dragão.

Um amigo escritor seu está tendo dificuldade pois está criando problemas demais para os seus personagens resolverem. Você, então, sugere que ele introduza o *Deus ex Machina* perfeito, o qual você gerará usando python.

### Entrada

A primeira linha da entrada consiste em um inteiro **N**, o número de problemas a serem resolvidos.

As próximas **N** linhas consistem, cada uma, de uma string sem espaços **P**, o nome do problema, uma string sem espaços **S**, a solução do problema **P**, e um inteiro **D**, a dificuldade (de 0 a 10) de resolver o problema.

### Saída

Seu programa deve imprimir uma única linha contendo as soluções **S** concatenadas na ordem de dificuldade, da maior para a menor. Quando houverem dois problemas com a mesma dificuldade, mantenha a ordem de input.

Exemplo de Entrada	Exemplo de Saída
2 voldemort_imortal reliquias_da_morte 8 grupo_perdido mestre_dos_magos 4	reliquias_da_mortemestre_dos_magos
4 jovem_carente muitos_beijos 1 falta_dinheiro ligacao_da_agencia 7 chao_molhado sapatos_com_espinhos 6 ceu_pegando_fogo extintor_magico 10	extintor_magicoligacao_da_agenciasapatos_com_espinhosmuitos_beijos

Tabela 3: Questão C

## Questão D - Estrada (Tuplas)

Cunegonde é uma jovem que tem dificuldade em se localizar dentro da cidade onde mora. Por sorte, ela mora na Cartésia, uma cidade em que todas as quadras são identificadas por coordenadas em um plano cartesiano que atravessa a cidade.

Cansada de não saber qual direção tomar para voltar para casa, Cunegonde trouxe seu dilema ao melhor programador que conhece, você. Ela pediu para escrever um programa que, dada a descrição dos seus movimentos desde que saiu de casa, saiba quais os movimentos mínimos para que ela regresse.

### Entrada

A primeira linha da entrada consiste de um inteiro **N**, o número de movimentos que Cunegonde performou após sair de casa.

As próximas **N** linhas consistem de um caractere **D** e um inteiro positivo **Q**, respectivamente, a direção ('N'orte, 'S'ul, 'L'este ou 'O'este) e a quantidade de blocos que Cunegonde andou nesta direção. **Saída**

Seu programa deve imprimir quatro inteiros **N**, **S**, **L**, **O**, nesta ordem, a quantidade mínima de quadras que Cunegonde precisa andar para retornar para a sua casa, na quadra de início.

Exemplo de Entrada	Exemplo de Saída
6 N 4 O 1 S 8 O 3 L 2 N 7	0 3 2 0
2 S 10 O 10	10 0 10 0

Tabela 4: Questão D

## Questão E - Enumeração (Dicionários)

Gilsernardo é um curioso bibliotecário que deseja catalogar todas as obras de Shakespeare. Desde os títulos, em ordem lexicográfica, até cada uma das palavras em cada um dos livros. Só que ele percebeu que é palavra demais para ele organizar no seu bloco de notas (884.647, para ser mais exato), então ele decidiu chamar o seu amigo programador, você, para auxiliá-lo nesta atividade.

Seu trabalho é escrever um programa que, dado um trecho de obra literária, organize-a por frequência de palavras.

### Entrada

A entrada consiste de apenas uma linha, a obra em questão. Perceba que todas as palavras estarão separadas por espaços, mas pode haver pontuação e variação de capitalização, que não deverão ser levados em conta na contagem. Por exemplo, “abacate”, “Abacate”, “ABACATE”, “abacate.” e demais variantes sempre contam como a mesma palavra.

### Saída

Seu programa deve imprimir múltiplas linhas. Cada linha deve conter uma string **W** e um inteiro **Q**, a palavra capitalizada e sua quantidade no trecho fornecido, respectivamente. A saída deve estar em ordem de **Q** maior para **Q** menor e também **W** do maior para menor (ordem lexicográfica).

Exemplo de Entrada	Exemplo de Saída
Come, night, come, Romeo, come, thou day in night, For thou wilt lie upon the wings of night Whiter than new snow on a raven's back.	Night 3 Come 3 Thou 2 Wings 1 Wilt 1 Whiter 1 Upon 1 The 1 Than 1 Snow 1 Romeo 1 Ravens's 1 On 1 Of 1 New 1 Lie 1 In 1 For 1 Day 1 Back 1 A 1

Tabela 5: Questão E

## Questão F - Filosofia (Tuplas)

Aristônio é um filósofo grego daqueles que usa toga. Ultimamente, Aristônio tem desenvolvido teorias em quantidades industriais, sobre os mais variados temas: o Universo, a Vida, e Todas as Coisas.

Porém, existe um lado negativo em ser um filósofo no século XXI: se você corre pelado na rua gritando ‘Eureka’, você acaba preso. E toda vez que Aristônio é preso, ele acaba perdendo todas as suas teorias.

Frustrado em ter que escrever tudo novamente, toda vez, Aristônio decidiu que iria aderir às novas tecnologias e guardar tudo na nuvem. Porém, ele sempre se perde em tantos arquivos, e precisa da ajuda de um programador para organizar seus pensamentos para ele. E é aí que você entra na história.

### Entrada

A primeira linha da entrada contém um inteiro **N**, o número de trabalhos que Aristônio escreveu.

As próximas **N** linhas contêm, cada uma, uma string **N**, o caminho do arquivo a ser indexado, e quatro strings **T1**, **T2**, **T3** e **T4**, separadas por espaço, tags que identificam o arquivo.

A última linha da entrada é a pesquisa do Aristônio, e contém um número arbitrário de strings, as tags dos arquivos que ele quer encontrar.

### Saída

Seu programa deve procurar as tags que o Aristônio está requisitando na última linha da entrada e imprimir cada um dos caminhos encontrados em linhas separadas, na ordem de input.

Exemplo de Entrada	Exemplo de Saída
3 tratado/inteligencia.pdf int tratado x ruim ensaio/cegueira.doc visao ensaio y bom teoria/caverna.jpg teoria visao z bom visao	ensaio/cegueira.doc teoria/caverna.jpg
4 nihon/accioly.pages ota abacate rio 10 brazil/renato.form ota aba n cate code/rafael.cpp 10 20 30 40 sha256/md5.py ololo secret abacate x 10 ota	nihon/accioly.pages brazil/renato.form code/rafael.cpp

Tabela 6: Questão F

## Questão G - Função de Ackermann(Dicionário)

Raphael é um jovem aluno de computação que anda estudando complexidade de algoritmos, e esbarrou na função de Ackermann. Na teoria da computabilidade, a **Função de Ackermann**, nomeada por *Wilhelm Ackermann*, é um dos mais simples e recém-descobertos exemplos de uma função computável que não são funções recursivas primitivas. Todas as funções recursivas primitivas são totais e computáveis, mas a Função de Ackermann mostra que nem toda função total-computável é recursiva primitiva.

Esta função é conhecida pela sua capacidade de crescer absurdamente com pequenas entradas. Ela cresce mais rápido do que qualquer função exponencial! Ela pode ser definida por:

$$F_{Ack}(x, y) = \begin{cases} y + 1, & \text{se } x = 0 \\ F_{Ack}(x - 1, 1), & \text{se } y = 0 \\ F_{Ack}(x - 1, F_{Ack}(x, y - 1)), & \text{caso contrário} \end{cases}$$

Porém, Raphael anda com dificuldades em calcular os valores de Ackermann na mão. Por isso, ele procurou o melhor programador python que conhece, você, para criar um programa que compute os valores de Ackermann.

### Entrada

A primeira e única linha da entrada contém apenas dois inteiros X e Y, as entradas da função de Ackermann.

### Saída

Seu programa deve imprimir um único inteiro **R**, o valor de retorno de  $F_{Ack}(x, y)$

Exemplo de Entrada	Exemplo de Saída
2 1	5
3 6	509

Tabela 7: Questão G



## Questão H - Gabirint (Tuplas)

A Cada rodada, o jogador da vez escreve seu nome e um número numa lista. O número referencia uma posição da lista, o próximo elemento a ser analisado. Por exemplo, na lista a seguir:

1. Roberto, 3
2. Ricardo, 2
3. José, 5
4. Roberto, 3
5. Ricardo, 2

O jogo acaba quando o último valor jogado fecha um loop (no exemplo acima, o loop é 5,2,1,3), e o vencedor é o dono da linha que aponta para quem fechou o loop (no caso, José).

As crianças de Inventadu estão de saco cheio de ficar discutindo por quem ganhou e por que ainda jogam um jogo tão complicado e sem sentido. Por isso, você se disponibilizou para escrever um programa que resolve o problema deles e, dado um jogo já finalizado, aponta o vencedor.

### Entrada

A primeira linha da entrada consiste de um inteiro **N**, o número de jogadas. As próximas **N** linhas contêm, cada uma, uma string **J** e um inteiro **I**, os valores de cada jogada de Gabirint.

### Saída

Seu programa deve imprimir na tela uma única linha contendo o nome do vencedor da partida em questão.

Exemplo de Entrada	Exemplo de Saída
5 Roberto 3 Ricardo 1 José 5 Roberto 3 Ricardo 2	José
2 Fulano 2 Sicrano 1	Fulano

Tabela 8: Questão H

## Questão I - Guloso (Dicionários)

Bobernardo é um grande produtor de um soro muito especial, uma injeção que te dá capacidades especiais de se transformar em um titã comedor de gente. O problema é que a produção do soro envolve muitos passos independentes, mas que têm horários específicos de começo e término, e ele só pode executar uma deles de cada vez.

Tendo dificuldade em otimizar a produção de soro, Bobernardo pediu a sua ajuda para auxiliá-lo na sua tarefa de aniquilar a humanidade. Seu trabalho é desenvolver um programa que, dados os horários de início e fim de muitas atividades (que podem ser concorrentes ou não), diga qual o número máximo de atividades daquele dia.

### Entrada

A primeira linha da entrada consiste de um inteiro **N**, o número de atividades propostas. As próximas **N** linhas contém, cada uma, três strings **T**, **C** e **D**, o título da atividade, e os horários de começo e de fim, no formato **hh:mm** de 24h no dia, respectivamente. Suponha que nenhum horário irá atravessar meia-noite e que duas atividades diferentes nunca terão o mesmo nome.

### Saída

Seu programa deve imprimir várias linhas na saída padrão, das quais a primeira deve conter um único inteiro **M**, o número máximo de atividades a serem realizadas naquele dia. As próximas linhas devem conter os nomes das **M** atividades a serem realizadas naquele dia, em ordem de execução, separadas por espaços.

Exemplo de Entrada	Exemplo de Saída
3 Retirada de sangue 07:15 08:00 Testes com macacos 07:30 09:00 Futebol titã 08:10 08:20	2 Retirada de sangue Futebol titã
5 Oração do dia 01:20 18:00 Oração de fechamento 19:00 21:00 Desmembramento casual 03:00 04:00 café da manhã 05:00 06:00 apocalipse zumbi 05:50 22:00	3 Desmembramento casual café da manhã Oração de fechamento

Tabela 9: Questão I

## Questão J - Filosfigthers (Dicionários e Tuplas)

Filosfigthers é um jogo que a revista Superinteressante lançou em meados de 2011, que estrela diversos figurões da Filosofia mundial em uma batalha de ideias e sangue.

Depois de seis anos, a Superinteressante decidiu que seria o momento de criar uma nova feature para o jogo dela, o suporte à criação de torneios. Mas para isso, eles precisam de um programador python muito inteligente, por isso eles chamaram você.

Seu trabalho é definir quem são os derrotados de um torneio, fornecidos os resultados das lutas.

### Entrada

A primeira linha da entrada consiste de um inteiro  $\mathbf{F}$ , o número de filósofos cadastrados no torneio atual.

As próximas  $\mathbf{F}$  linhas contém, cada uma, um inteiro  $T_F$  e uma string  $\mathbf{N}$ , o identificador de um filósofo no torneio e seu nome, respectivamente.

As linhas seguintes descreverão cada uma das lutas do torneio e conterão, cada uma, três inteiros  $T_L$ , o identificador desta luta,  $T_{F_1}$  e  $T_{F_2}$ , os lutadores envolvidos, e  $T_v$ , o vencedor. A última linha da entrada, por fim, possui uma string '**FINISHHIM**', e o identificador da luta final.

Note que não necessariamente a ordem de input será a ordem de execução das lutas.

Considere sempre que dois filósofos diferentes nunca compartilharão o mesmo nome, ou o mesmo identificador.

Considere que filósofos nunca terão o mesmo identificador de uma luta.

Perceba que  $T_{F_1}$  e  $T_{F_2}$  podem ser identificadores de lutadores ou de outras lutas. Neste caso, ele identifica o vencedor da luta identificada.

### Saída

Seu programa deve imprimir um único inteiro na saída padrão, o número de lutas que o vencedor teve que ganhar para vencer o torneio. Perceba que nem todas as lutas cadastradas fazem parte da sequência de lutas que leva à final. Estas lutas não devem ser consideradas.

Exemplo de Entrada	Exemplo de Saída
4 1 Aristóteles 2 Platão 3 Nietzsche 4 Descartes 5 1 2 1 6 3 4 4 7 5 6 6 FINISHHIM 7	2
6 10 SantoAgostinho 0 DavidHume 7 Wittgenstein 1 SãoTomásDeAquino 8 Hegel 2 Kant 102 7 1 7 100 0 8 0 103 102 101 102 104 100 103 100 101 2 10 10 FINISHHIM 104	2

Tabela 10: Questão J