

Lista de Exercícios 1

Tuplas

1. Faça um programa que ajude um menino a decidir como deve comprar o presente de sua namorada. A forma como o apaixonado pagará pelo presente depende de quanto tem na carteira ("valor em caixa") e o valor do presente. Caso o preço do presente seja superior a 80% do valor em caixa, a compra deve ser feita a prazo (5x), com juros de 10% sobre o valor do presente. Caso o presente custe entre 50% e 80% do valor em caixa, a compra deve ser feita a prazo (3x), com juros de 8% sobre o valor do presente. Caso contrário, a compra deverá ser realizada a vista, onde o apaixonado receberá 5% de desconto.

Seu programa deve capturar o "valor em caixa" e o valor do presente, ativar a função abaixo, e mostrar o número de parcelas e o valor a pagar em cada parcela de acordo com a regra acima.

- Faça uma função que receba o "valor em caixa", o valor do presente e retorne uma tupla com o número de parcelas (1, 3 ou 5) e o valor da parcela.

2. Faça uma função **ordena** que receba três valores inteiros a, b e c e retorna uma tupla com os valores ordenados crescentemente.

Os três jurados de um concurso de fantasias avaliam 2 critérios distintos: originalidade e beleza. A nota final do candidato é calculada do seguinte modo: $0.6 * \text{nota_da_originalidade} + 0.4 * \text{nota_da_beleza}$

Para evitar distorções, são desprezadas a maior e a menor nota de cada critério. Faça um programa, que leia, para o candidato, o seu número de inscrição (inteiro) e suas 3 notas (reais) de cada critério, exibindo sua nota final.

O seu programa deve, obrigatoriamente, utilizar a função **ordena**.

Modificação: Crie um arquivo com o número de inscrição e notas de vários candidatos (um candidato por linha com seus dados separados por ';'). Altere seu programa para utilizá-lo.

3. Escreva uma função que receba um número inteiro representando a quantidade total de segundos e retorne uma tupla com a quantidade de segundos expressa em Horas, Minutos e Segundos.

Faça um programa que leia, para cada atleta, seu número de inscrição e o tempo (em s) que levou para completar uma maratona, exibindo-o no formato HH:MM:SS. Para finalizar, digite a inscrição 0.

No final, deve mostrar o tempo (no formato HH:MM:SS) e o nº do atleta que levou menos tempo e o tempo (no formato HH:MM:SS) e o nº do atleta que levou mais tempo.

Observação 1: sabe-se que atletas com tempo superior a 36000s são desclassificados.

Modificação: Crie um arquivo com o número de inscrição do atleta e tempo em s que levou para completar a maratona (um atleta por linha com seus dados separados por tab '\t'). Altere seu programa para utilizá-lo.

4. Um número inteiro x maior do que 1 é dito um número perfeito se ele é igual à soma de todos os seus divisores, com exceção dele mesmo. Um divisor só pode aparecer uma única vez no somatório. Um número B é divisor de um número A se o resto da divisão inteira de A por B é igual a 0 (ZERO).

Por exemplo: 6 é um número perfeito, pois $6 = 1+2+3$

8 não é perfeito, pois a soma dos divisores é 7 ($1+2+4$)

28 é perfeito, pois $28 = 1+2+4+7+14$

a) Escreva a função **ehPerfeito** que :

- recebe um número inteiro maior do que 1
- retorna a tupla (True/False, [lista dos divisores do número]), onde True/False representa se o número é perfeito ou não.

Para o exemplo com o número 6 a tupla retornada é (True,[1,2,3])

b) Escreva a função **avaliaNumero** que recebe um número inteiro >1 e exibe na tela o número, seus divisores (com exceção dele mesmo) e se o número é ou não perfeito. É obrigatório que esta função chame a função **ehPerfeito**.

5. Escreva uma função recursiva **contaElementos** que receba 1 tupla de elementos de qualquer tipo e retorna a quantidade de valores inteiros existentes na tupla em qualquer nível. Exemplos:

contaElementos(([1,2], 'a', 7.3, 3, [10,20])) retorna 5

contaElementos(([1,2,3], ['a', 'b', 'c'])) retorna 3

contaElementos(([1, (2, 3), ('a', (2.0, ([3, 9.0], 'casa', (1, 'Ana'), ['a', 'b']))], 4.4, 20, 10, ['a', 'b', 'c'])) retorna 7

Construa um programa para testar sua função.

6. Escreva uma função recursiva **testaContida** que receba 1 tupla de elementos de qualquer tipo e retorna True se em qualquer nível existir a sequência 'oi'. Exemplos:

testaContida(("foi",)) retorna True

testaContida(([1,2,3],(('onde',2.3), 'oi',78), ['a', 'b', 'c'])) retorna True

testaContida((('MAO', 'MOA') , 13.8 , 'c' , 6 , [2,3])) retorna False

Construa um programa para testar sua função.

7. Escreva uma função recursiva **conta** que receba 1 tupla de elementos de qualquer tipo e uma string, retornando a quantidade de ocorrências da string na tupla (em qualquer nível). Exemplos:

testa(("foi",), "oi") retorna 1

testa(((([1,2,3],(('onde',2.3), 'oi',78), ['a', 'b', 'c'])), "oi") retorna 1

testa(([1,2,3],(('onde', 'noiva',2.3), 'oi',78), ['a', 'b', 'c']) , "oi") retorna 2

testa((('MAO', 'MOA') , 13.8 , 'c' , 6 , [2,3]) , "oi") retorna 0

Construa um programa para testar sua função.

8. Faça uma função que receba uma tupla com a temperatura média de cada mês do ano. Esta função deve:

- Exibir todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . .).
- Devolver uma tupla com 2 elementos. O primeiro elemento representa a menor temperatura e o mês em que ocorreu. O segundo elemento representa a maior temperatura e o mês em que ocorreu. Considere que não há empate.

Construa um programa para testar sua função.

9. A programação das apresentações de um concurso de duplas sertanejas, realizado em dois dias diferentes, utiliza a diferença das alturas dos integrantes da dupla para escolher o dia da apresentação.

Faça um programa (usando a função **organiza**), que capture o número da dupla, o nome dos integrantes da dupla e a altura, em cm, de seus (é óbvio) 2 integrantes. O término do processamento ocorre quando é informado 0 como número da dupla.

Este programa deve exibir a programação das apresentações de cada dia. As apresentações de cada dia devem estar numa lista, onde cada dia é uma

sublista. A 1ª sublista armazena os dados das duplas do 1º dia e a 2ª sublista armazena os dados das duplas do 2º dia. No 1º dia, ficam as duplas com diferença de altura inferior ou igual a 15 cm e no 2º dia as demais duplas.

Os dados das duplas de sertanejos são armazenados em tuplas com 3 elementos: o nº da dupla, nome do integrante com maior altura e nome do integrante com menor altura.

A função **organiza**: recebe os dados dos integrantes em uma tupla com dois elementos. O 1º elemento desta tupla é também uma tupla com o nome, e altura do 1º integrante e o 2º elemento é uma tupla com o nome e altura do 2º integrante. Esta função deve retornar uma tupla com dois elementos, onde o 1º elemento é a diferença de altura entre eles e o segundo elemento é uma tupla com o nome do integrante mais alto e o nome do integrante mais baixo.

Exemplo: tupla recebida (('Zozó',100),('Zazá',190))

tupla retornada: (90,('Zazá', 'Zozó'))

10. Uma estufa de flores é monitorada por 6 sensores que registram a temperatura e a umidade da estufa 8 vezes ao longo do dia. Estes registros são realizados em horas variadas e estão no arquivo **Registros.txt**, agrupados por sensor. Faça um programa, utilizando a função **um_sensor**, que leia o arquivo e apresente para cada sensor:

- a menor temperatura e em que hora ocorreu
- o menor grau de umidade e em que hora ocorreu

O arquivo **Registros.txt** armazena em cada linha, separados por ponto e vírgula, o nº do sensor, a hora da monitoração, a temperatura e o grau de umidade nesta hora. As 8 medidas de um sensor estão em linhas consecutivas no arquivo.

No final, o programa deve mostrar, agrupados por hora:

- os menores valores de temperatura medidos e respectivo sensor
- os menores valores de umidade medidos e respectivo sensor

Saída esperada para o arquivo:

```
Sensor 3- Temperatura Mínima 12.3°C às 2h e Umidade Mínima: 35.8% às 2h
Sensor 2- Temperatura Mínima 12.5°C às 0h e Umidade Mínima: 32.5% às 10h
Sensor 4- Temperatura Mínima 12.3°C às 6h e Umidade Mínima: 33.7% às 4h
Sensor 5- Temperatura Mínima 16.7°C às 4h e Umidade Mínima: 32.8% às 20h
Sensor 1- Temperatura Mínima 14.3°C às 4h e Umidade Mínima: 36.1% às 4h
Sensor 6- Temperatura Mínima 14.2°C às 5h e Umidade Mínima: 32.5% às 6h
```



DEPARTAMENTO
DE INFORMÁTICA
PUC-RIO

INF1026 2017.2

Ciclo Básico do Centro Técnico Científico (CB-CTC)

E-mail: coordprog@inf.puc-rio.br

URL: www.inf.puc-rio.br/~inf1025

Hora 2h: Sensor: 3 - Temperatura: 12.3°C
Hora 0h: Sensor: 2 - Temperatura: 12.5°C
Hora 6h: Sensor: 4 - Temperatura: 12.3°C
Hora 4h: Sensor: 5 - Temperatura: 16.7°C
Sensor: 1 - Temperatura: 14.3°C
Hora 5h: Sensor: 6 - Temperatura: 14.2°C

Faça a função **um_sensor** que recebe o arquivo, lê os 8 registros de um sensor e retorna uma tupla com três elementos:

- o nº do sensor
- registro da menor temperatura (a menor temperatura, a hora em que ocorreu)
- registro da menor umidade (o menor grau de umidade, hora em que ocorreu)

11. Em um jogo de bingo promovido pela prefeitura, há vários participantes. Cada participante concorre com apenas uma cartela. A cartela é representada por m linhas numeradas de 1 a m, sendo que cada linha contém n posições que variam de 1 a n.

Considere a seguinte representação para os dados:

- Lugar da cartela: (linha, posição, número), onde linha é representada por um inteiro de 1 a m e posição, por um inteiro de 1 a n e número é o valor entre 1 e 60 do lugar da cartela.
- Cartela: Lista de duplas (lugar, situação) sendo que a situação é: 1 para indicar que o lugar foi sorteado e 0 para indicar lugar vago.
- Jogo Finalizado: Lista de duplas (jogador, cartela), onde jogador é o nome do jogador da cartela.

Escreva funções que:

- Dada uma cartela retorne a quantidade total de números sorteados.
- Dado um lugar e uma cartela retorne True se o lugar está vago ou False caso contrário.
- Dado um lugar e uma cartela, verifique se existe algum vizinho lateral deste lugar que está livre.
- Dada uma cartela, retorne a(s) linha(s) com o maior número de lugares livres.

- Dado um Jogo, retorne o(s) jogador(es) vencedor(es) (todos os lugares ocupados).

12. Uma instituição monitora a temperatura e a umidade de cada um dos seus 6 laboratórios mantendo 1 sensor em cada laboratório. Durante 8 vezes ao longo do dia em cada laboratório seu sensor registra hora, a temperatura e a umidade naquela hora. Estes registros são realizados em horas cheias (sem minutos). As informações de um determinado dia estão gravadas no arquivo **RegistrosLab.txt**.

O arquivo **RegistrosLab.txt** armazena em cada linha, separados por ponto e vírgula, o nome do laboratório, a hora da monitoração (inteiro), a temperatura (real) e o grau de umidade (inteiro de 0 a 100) nesta hora. As 8 medidas de um laboratório estão em linhas consecutivas no arquivo.

Assuma que não há repetição de temperaturas em um mesmo laboratório em um determinado dia.

Faça um programa que leia o arquivo e apresente para cada laboratório:

- o nome do laboratório
- a menor temperatura e em que hora ocorreu
- o menor grau de umidade e em que hora ocorreu

Ao final, o programa deve mostrar as temperaturas mínimas dos laboratórios em um dia organizadas por hora.

Exemplo:

Temperaturas mínimas dos laboratórios em um dia organizadas por hora

h = 6: LAB4 12; LAB1 18

h = 9: LAB2 8 ; LAB3 12

h = 10: LAB5 20

h = 18: LAB6 20

O programa deve obrigatoriamente utilizar a função **trataUmLaboratorio** para processar as informações de um laboratório, que deve ser escrita por você.

A função **trataUmLaboratorio**:

- recebe o arquivo aberto
- lê 8 linhas consecutivas do arquivo (correspondendo aos 8 registros de um único laboratório), retornando uma tupla com três elementos:
 - o nome do laboratório
 - a tupla: (a menor temperatura, a hora em que ocorreu)



INF1026 2017.2

Ciclo Básico do Centro Técnico Científico (CB-CTC)

E-mail: coordprog@inf.puc-rio.br

URL: www.inf.puc-rio.br/~inf1025

- a tupla :(o menor grau de umidade, hora em que ocorreu)

Saída esperada para o arquivo fornecido:

Lab DI546 - Temperatura Mínima 12.3°C às 2h e Umidade Mínima: 34.0% às 4h
Lab CB252 - Temperatura Mínima 12.5°C às 0h e Umidade Mínima: 53.0% às 16h
Lab DI548 - Temperatura Mínima 12.3°C às 6h e Umidade Mínima: 55.0% às 4h
Lab FI362 - Temperatura Mínima 16.7°C às 4h e Umidade Mínima: 30.0% às 12h
Lab CB318 - Temperatura Mínima 14.3°C às 4h e Umidade Mínima: 34.0% às 6h
Lab QI170 - Temperatura Mínima 14.2°C às 5h e Umidade Mínima: 36.0% às 7h

Dep DI:

Lab DI546 - Mínima: 12.3°C - 2h

Lab DI548 - Mínima: 12.3°C - 6h

Dep CB:

Lab CB252 - Mínima: 12.5°C - 0h

Lab CB318 - Mínima: 14.3°C - 4h

Dep FI:

Lab FI362 - Mínima: 16.7°C - 4h

Dep QI:

Lab QI170 - Mínima: 14.2°C - 5h