

## 1 Laboratório 3

1. **Bônus:** Cada funcionário de uma empresa possui um nome, seu tempo de casa (em meses) e seu salário. A empresa armazena os dados de cada funcionário utilizando tuplas, e possui uma lista de tuplas com os dados de todos os funcionários:

```
1 l = [("Valentina", 4, 1500), ("Enzo", 33, 1200),  
2      ("Anna Julia", 22, 3000), ("Simaria", 33, 1400)]
```

O segundo elemento da lista  $l$ , por exemplo, indica que o funcionário “Enzo” foi contratado há 33 meses e recebe R\$1.200,00 por mês. A empresa pediu para você implementar uma sub-rotina que descubra qual é o funcionário com mais tempo de casa para lhe dar uma bonificação de 10% este mês (caso haja mais de um funcionário com este mesmo tempo de casa, todos eles receberão a bonificação). O procedimento a seguir recebe a lista de funcionários como parâmetro e imprime o nome do(s) funcionário(s) mais antigo(s) com seu salário diferenciado neste mês.

2. **Lucro:** A mesma empresa deseja saber se seus produtos estão dando lucro. Para isso, pediu para você criar uma sub-rotina que recebe uma lista de tuplas contendo o preço de custo e o preço de venda de cada mercadoria e imprima:
  - a quantidade de produtos com menos de 20% de lucro
  - a porcentagem de produtos com lucro superior a 25%
3. **Alinhados:** Dada três tuplas representando pontos no plano cartesiano, verifique se os pontos estão alinhados.
4. **Agência de turismo:** Uma agência de turismo possui armazenados os voos realizados por diversas companhias aéreas. Cada voo é representado como uma tupla com as seguintes informações:
  - Número do voo
  - Companhia que realizou o voo (String).
  - Lista de escalas (cada elemento da lista é o nome de uma cidade, na ordem em que foram visitadas).

Exemplo: voos = [ (1024, "TAM", ["ES", "RJ", "SP", "NY"]), (1025, "GOL", ["ES", "SP"]) ]

Crie funções para:

- (a) Dada a lista de voos, uma cidade origem  $a$  e uma cidade destino  $b$ , imprima o número e a companhia de todos os voos que se iniciem em  $a$  e cujo destino final seja  $b$ .

- (b) Dada a lista de voos, uma cidade origem  $a$  e uma cidade destino  $b$ , imprima quantos voos se iniciem em  $a$  e que façam alguma escala em  $b$ .
- (c) Dada a lista de voos, uma cidade origem  $a$  e uma cidade destino  $b$ , verifique se há algum voo direto de  $a$  para  $b$ , mesmo que  $a$  e  $b$  não sejam os destinos iniciais e finais do voo.

5. **Copa do Mundo:** [*Maratona de Programação 2008*] Uma Copa do Mundo de futebol de botões está sendo realizada com times de todo o mundo. A classificação é baseada no número de pontos ganhos pelos times, e a distribuição de pontos é feita da forma usual. Ou seja, quando um time ganha um jogo, ele recebe 3 pontos; se o jogo termina empatado, ambos os times recebem 1 ponto; e o perdedor não recebe nenhum ponto. Dada a classificação atual dos times e o número de times participantes na Copa do Mundo, sua tarefa é criar uma função que determine quantos jogos terminaram empatados até o momento. A função deve receber dois parâmetros: o número de jogos já realizados até o momento e uma lista com a classificação atual. A classificação atual é dada por uma lista de tuplas, onde cada tupla contém o nome de um time e os pontos ganhos por aquele time. Exemplos:

<i>Entrada</i>		<i>Saída</i>
<i>Qtd de jogos</i>	<i>Classificação</i>	
3	$[(Brasil, 3), (Australia, 3), (Croacia, 3)]$	0
3	$[(Brasil, 5), (Japao, 1), (Australia, 1)]$	2

## 2 Laboratório 4

Em um Bolão, vários apostadores se juntam para adquirir uma série de cartões de apostas. Pode-se optar também por uma única aposta, mas com uma quantidade maior de números. Por exemplo, escolhem-se 8 números para apostar na Mega Sena, sendo que apenas 6 números serão sorteados nesta loteria. Se 6 dentre estes 8 números forem sorteados, a aposta é considerada vencedora. As chances de acerto aumentam, mas é bem mais caro apostar mais de 6 números em um mesmo cartão.

Você decidiu criar um sistema para administrar bolões. Cada cartão apostado poderá ter uma quantidade qualquer de pessoas participando, e ao menos 6 números. Seu sistema deve ser capaz de armazenar nome e CPF de cada jogador, além de várias apostas diferentes. A interface do sistema, por enquanto, será no terminal e deve permitir, ao menos:

- Inserir novo jogador
- Visualizar jogadores cadastrados
- Inserir nova aposta

- Visualizar apostas cadastradas
- Inserir resultado e listar apostas vencedoras

**Estruturas:** Cada jogador será representado por uma tupla contendo nome e CPF. Exemplos:

```
("Enzo", "123-456-789-00")  
("Valentina", "234.567-890-00")
```

Cada aposta será representada por uma tupla contendo os CPFs dos apostadores e os números sorteados. Exemplos:

```
( ["123-456-789-00", "234.567-890-00"], [12, 23, 35, 39, 41, 46, 52, 55] )  
( ["123-456-789-00"], [14, 25, 31, 33, 41, 46, 53] )
```

Repare que Enzo participou de duas apostas. Na primeira, junto da Valentina, eles apostaram 8 números. Na segunda, ele apostou 7 números sozinho. Seu programa, portanto, irá armazenar duas listas (uma lista de jogadores, e uma lista de apostas). Exemplo:

```
1 jogadores = [ ("Enzo", "123-456-789-00"),  
2             ("Valentina", "234.567-890-00") ]  
3 apostas   = [ ( ["123-456-789-00", "234.567-890-00"],  
4               [12, 23, 35, 39, 41, 46, 52, 55] ) ,  
5             ( ["123-456-789-00"],  
6               [14, 25, 31, 33, 41, 46, 53] ) ]
```

### Exercícios:

1. Defina um menu para o programa, a ser exibido no console, com as seguintes opções:
  - Cadastrar novo jogador
  - Visualizar jogadores cadastrados
  - Inserir nova aposta
  - Visualizar apostas cadastradas
  - Inserir resultados e listar vencedores
  - Sair do programa

Utilize uma função auxiliar para limpar a tela do programa cada vez que uma opção for escolhida.

2. Crie uma função para a primeira opção do menu (cadastrar novo jogador). A função deve fazer a leitura do nome e CPF de um jogador e inseri-lo no sistema. Caso um jogador com o mesmo CPF já tenha sido inserido, deve exibir uma mensagem de erro e volta para o menu sem que ele tenha sido inserido. Utilize uma função auxiliar para verificar se algum funcionário com aquele CPF já foi inserido no sistema.
3. Crie uma função para a segunda opção do menu (visualizar jogadores cadastrados). A função deve imprimir os dados dos jogadores no seguinte formato:

```
Enzo - CPF: 123.456.789-00  
Valentina - CPF: 234.567.890-00
```

4. Crie uma função para cadastrar uma nova aposta. Modularize a função, de forma que ela utilize duas sub-rotinas auxiliares:
  - Função que faz a leitura da quantidade de jogadores que irão dividir o bilhete e dos CPFs dos jogadores participantes da aposta (sempre verificando se o CPF pertence a algum jogador já cadastrado no sistema). Quando um CPF não cadastrado for digitado, permanece solicitando um novo CPF até que o usuário digite um CPF correto, ou digite 0 para desistir do cadastro da aposta e voltar ao menu.
  - Função que faz a leitura da quantidade de números a serem apostados no bilhete e dos números apostados. Caso o usuário digite um número já inserido, deve permanecer solicitando outro número até que seja digitado um número válido.
5. Crie uma função para visualizar as apostas cadastradas no sistema. A função deve imprimir os dados das apostas no seguinte formato:

```
APOSTA 1  
Numeros: 12 23 35 39 41 46 52 55  
Jogadores:  
Enzo - CPF: 123.456.789-00  
Valentina - CPF: 234.567.890-00
```

6. Crie uma função para inserir um sorteio, verificar as apostas vencedoras, dividir o prêmio entre os bilhetes vencedores, e listar o prêmio que cada participantes irá receber. Modularize a função utilizando ao menos as seguintes sub-rotinas:
7. Crie uma função para inserir um sorteio, verificar as apostas vencedoras, dividir o prêmio entre os bilhetes vencedores, e listar o prêmio que cada participantes irá receber. Modularize a função utilizando ao menos as seguintes sub-rotinas:

- Função que faz a leitura dos 6 números sorteados, e retorna uma lista com esses números.
- Função que verifica se uma lista  $l_1$  está contida na lista  $l_2$ . Utilizada para conferir se todos os números sorteados fazem parte dos números apostas em um bilhete.
- Função que conta quais bilhetes foram premiados.
- Função que lista todos os vencedores de cada bilhete premiado, além do prêmio que cada apostador receberá por ele. Recebe como parâmetro os índices dos bilhetes premiados na lista de apostas, e o prêmio daquele bilhete, já dividido entre a quantidade de bilhetes que foram premiados. Para identificar cada bilhete, enumere-os a partir de 1. Exemplo:

```
Vencedores no bilhete 1
Enzo - CPF: 123.456.789-00 - R$ 50000.0
Valentina - CPF: 234.567.890-00 - R$ 50000.0
```

8. Implemente um jogo semelhante ao Genius. Muito popular na década de 1980, o brinquedo buscava estimular a memorização de cores. Com um formato semelhante a um OVNI, possuía botões coloridos que emitiam sons harmônicos e se iluminavam em sequência. Cabia aos jogadores repetir a ordem das cores sem errar. Ao invés de cores, utilizaremos números. A cada iteração, o programa irá sortear um algarismo aleatório, que será exibido para o usuário. Caberá ao usuário memorizar e digitar corretamente toda a sequência de caracteres exibidos.



**Figura 1:** Jogo Genius.