

1) Faça um programa que leia um arquivo texto que representa uma estrutura de menu e execute esse menu. Exemplo de arquivo:

```
1;Insere Cliente
2;Altera Cliente
3;Exclui Cliente
4;Deleta Cliente
5;Mostra Cliente
```

1o. campo, opção do menu, sempre inteiro

2o campo, descrição da opção do menu, sempre string

obs:

- A opção de fim do menu sempre será a última opção + 1, no exemplo acima 6.
- Dê mensagem de erro caso o usuário escolha uma opção inválida e imprima a opção escolhida
- coloque as rotinas de leitura do arquivo e impressão do menu em módulos separados

2) Faça uma evolução no programa acima sendo que a opção (1o campo) não seja um numero sequencial. Exemplo de arquivo:

```
1;Insere Cliente;insere_cliente
9;Mostra Cliente;mostra_cliente
```

1o. campo, opção do menu, sempre inteiro

2o campo, descrição da opção do menu, sempre string

A opção de fim do menu sempre será sempre '0' e ficará como última.

obs:

- A opção de fim do menu sempre será a última opção + 1, no exemplo acima 6.
- Dê mensagem de erro caso o usuário escolha uma opção inválida
- coloque as rotinas de leitura do arquivo e impressão do menu em módulos separados

3) Faça uma função que calcule N numeros primos sem usar COMPREHENSIONS.

4) Faça o exercicio acima usando o COMPREHENSIONS

5) Faça um programa em python que calcule o numero de domingos entre 2 datas.

6) Faça um programa que converta um json para uma classe. Estude a estrutura de um arquivo json. Use esse json como exemplo json = '{ "Name": "David", "Class": "I", "Age": 6 }'

7) Faça um programa que retorne se um numero está em uma lista. Use a pesquisa sequencial.

Exemplo:

```
print(pesquisa_sequencial([0, 1, 52, 3, 34, 14, 8, 18, 19, 34], 3)) => True
```

```
print(pesquisa_sequencial([0, 1, 52, 3, 34, 14, 8, 18, 19, 34], 17)) => False
```

8) Faça um programa que retorne se um numero está em uma lista. Use a pesquisa binária.

Exemplo:

```
print(pesquisa_binaria([0, 1, 3, 8, 14, 18, 19, 34, 52], 3)) => True
```

```
print(pesquisa_binaria([0, 1, 3, 8, 14, 18, 19, 34, 52], 17)) => False
```

9) use o COMPREHENSIONS para executar uma pesquisa sequencial

```
print(pesquisa_comprehensions([0, 1, 52, 3, 34, 14, 8, 18, 19, 34], 3)) => True
```

```
print(pesquisa_comprehensions([0, 1, 52, 3, 34, 14, 8, 18, 19, 34], 17)) => False
```

10) Faça uma função que receba uma lista e mostre somente os pares.

Exemplo:

```
print(somente_pares([0, 1, 3, 8, 14, 18, 19, 34, 52])) => [0, 8, 14, 18, 34, 52]
```