

Nome: \_\_\_\_\_ Matr.: \_\_\_\_\_

## Introdução

Hoje vamos considerar o problema de sincronização de agendas de telefone. Águeda Muniz tem duas agendas de telefones de seus contatos: agenda1 e agenda2. Ela deseja combinar as duas agendas em apenas uma contendo os telefones não repetidos das duas agendas. Portanto, considerando que as agendas sejam armazenadas usando conjuntos, o que ela quer é a *união* do conjunto que contém a agenda1 com o conjunto que contém a agenda2. Para facilitar a solução do problema, vamos supor que somente os números dos telefones é que serão armazenados respectivamente em dois arquivos. Suponha que o nome do arquivo com a agenda1 é agenda1.dat e o da agenda2 é agenda2.dat. Cada número de telefone será digitado em uma linha. Para testar o programa a ser elaborado, use os seguintes dados para o arquivo agenda1.dat:

3123-4567  
3234-5678  
3345-6789  
3123-4570  
3345-6799

Para o arquivo agenda2.dat, use os dados:

3456-7890  
3345-6789  
2351-1809  
3234-5678  
2981-5476  
2780-1245

Veja um exemplo de saída resultante da execução do programa mais adiante neste roteiro. Siga as instruções abaixo para fazer o programa.

## Instruções

1. Abra o IDLE e crie um novo arquivo fonte denominado p12.py. Não se esqueça de salvá-lo de tempos em tempos, porque pode ocorrer uma falha de energia elétrica durante a aula prática.
2. Digite os comentários obrigatórios (nome, matrícula, data e uma breve descrição sobre o que o programa faz).
3. Estruture seu programa em duas funções: `main()` e `sinc(a1, a2)`.
4. A função `main` deve proceder à leitura dos dados a partir dos dois arquivos, chamar a função `sinc` e imprimir a agenda final sincronizada. Para a leitura dos dados, use um arquivo de cada vez. Observe que cada arquivo de dados contém, em uma linha, um número de telefone. Veja o algoritmo em alto nível da função `main` abaixo.
5. Não se esqueça de fechar os arquivos de dados ao terminar de usá-los na função `main`.
6. A função `sinc` deve receber como parâmetros as duas agendas e retornar a agenda sincronizada de acordo com o que se pede.

7. Não se esqueça de chamar a função `main` no final de seu código fonte para iniciar todo o processo.
8. Veja como deve ser formatada a saída do programa no exemplo dado abaixo.

### Algoritmo para a função `main`

1. Abra o arquivo contendo a agenda1.
2. Inicialize o conjunto que conterá a agenda1.
3. Faça a leitura dos dados da agenda1 e adicione-os ao conjunto.
4. Imprima a quantidade de telefones lidos.
5. Feche o arquivo da agenda1.
6. Repita os passos de 1 a 5 para a agenda2.
7. Sincronize as agendas chamando a função `sinc`.
8. Imprima formatadamente os telefones da agenda sincronizada. Veja o exemplo de execução do programa abaixo.

☞ Não se esqueça de preencher o cabeçalho do código fonte com seus dados, a data de hoje e uma breve descrição do programa.

Após certificar-se de que seu programa esteja correto, envie o arquivo do programa fonte (`p12.py`) através do sistema de entrega do LBI.

### Exemplo de Execução do Programa

```
5 número(s) de telefone lido(s) na agenda1
6 número(s) de telefone lido(s) na agenda2
```

Agenda Sincronizada

```
2351-1809
2780-1245
2981-5476
3123-4567
3123-4570
3234-5678
3345-6789
3345-6799
3456-7890
```