Login!

Escreva a função `login(dicionario, email, senha)` que recebe um dicionario com dados de empregados, um email e uma senha. O dicionário mapeia uma senha para uma lista de funcionários, uma vez que pode haver duas pessoas com a mesma senha.

Sua função deve retornar `True` se o email e senha passados são válidos.

Exemplos de asserts

```
dados = {1313:['j@gmail.com'], 1226:['e@cc.com', 'd@cc.com'], 1507:['pedro@cc
assert login(dados, 'd@cc.com',1313) == False
assert login(dados, 'd@cc.com', 1226) == True
assert login(dados, 'joao@gmail.com', 123) == False
```

Última atualização por daltonserey, 2 anos atrás

5844866149908480/login.py

```
# luiz.augusto.farias@ccc.ufcg.edu.br

def meu_in(lista, elemento):
    for i in range(len(lista)):
        if lista[i] == elemento:
            return True


def login(dicionario, email, senha):
    for senhas, emails in dicionario.items():
        if meu_in(emails, email) == True:
            if senha == senhas:
                return True
    return False

dados = {1313:['j@gmail.com'], 1226:['e@cc.com', 'd@cc.com'], 1507:['pedro@cc.com'] }
assert login(dados, 'd@cc.com',1313) == False
assert login(dados, 'd@cc.com', 1226) == True
assert login(dados, 'joao@gmail.com', 123) == False
```

```
! .
---
! undefined
```

Para editar sua resposta:

1. faça o *checkout* da atividade com o comando `tst checkout 5844866149908480`;
2. altere sua resposta, editando o arquivo ou criando novos no diretório do *checkout*;
3. faça o *commit* com o comando `tst commit ⟨nome-do-seu-arquivo⟩`.