```
agenda = []
# Variável para marcar uma alteração na agenda
alterada = False
def pede_nome(padrão=""):
 nome = input("Nome: ")
 if nome == "":
 nome = padrão
 return nome
def pede_telefone(padrão=""):
 telefone = input("Telefone: ")
 if telefone == "":
 telefone = padrão
 return telefone
def pede_email(padrão=""):
 email = input("Email: ")
 if email == "":
 email = padrão
 return email
def pede_aniversário(padrão=""):
 aniversário = input("Data de aniversário: ")
 if aniversário == "":
 aniversário = padrão
 return aniversário
def mostra_dados(nome, telefone, email, aniversário):
 print(
 f"Nome: {nome}\nTelefone: {telefone}\n"
 f"Email: {email}\nAniversário: {aniversário}\n"
 )
def pede_nome_arquivo():
 return input("Nome do arquivo: ")
def pesquisa(nome):
```

```
 mnome = nome.lower()
 for p, e in enumerate(agenda):
if e[0].lower() == mnome:
 return p
 return None
def novo():
 global agenda, alterada
 nome = pede_nome()
 if pesquisa(nome) is not None:
 print("Nome já existe!")
 return
 telefone = pede_telefone()
 email = pede_email()
 aniversário = pede_aniversário()
 agenda.append([nome, telefone, email, aniversário])
 alterada = True
def confirma(operação):
 while True:
 opção = input(f"Confirma {operação} (S/N)? ").upper()
 if opção in "SN":
 return opção
 else:
 print("Resposta inválida. Escolha S ou N.")
def apaga():
 global agenda, alterada
 nome = pede_nome()
 p = pesquisa(nome)
 if p is not None:
      if confirma("apagamento") == "S":
 del agenda[p]
 alterada = True
```

```
 else:
 print("Nome não encontrado.")
def altera():
 global alterada
 p = pesquisa(pede_nome())
 if p is not None:
 nome = agenda[p][0]
 telefone = agenda[p][1]
 email = agenda[p][2]
 aniversário = agenda[p][3]
 print("Encontrado:")
 mostra_dados(nome, telefone, email, aniversário)
 nome = pede_nome(nome) # Se nada for digitado, mantém o valor
 telefone = pede_telefone(telefone)
 email = pede_email(email)
aniversário = pede_aniversário(aniversário)
      if confirma("alteração") == "S":
 agenda[p] = [nome, telefone, email, aniversário]
 alterada = True
 else:
 print("Nome não encontrado.")
def lista():
 print("\nAgenda\n\n\------")
 # Usamos a função enumerate para obter a posição na agenda
 for posição, e in enumerate(agenda):
 # Imprimimos a posição
 print(f"\nPosição: {posição}")
 mostra_dados(e[0], e[1], e[2], e[3])
 print("\------\n")
def lê_última_agenda_gravada():
 última = última_agenda()
```

```
 if última is not None:
 leia_arquivo(última)
def última_agenda():
 try:
 arquivo = open("ultima agenda.dat", "r", encoding="utf-8")
 última = arquivo.readline()[:-1]
 arquivo.close()
 except FileNotFoundError:
 return None
 return última
def atualiza_última(nome):
 arquivo = open("ultima agenda.dat", "w", encoding="utf-8")
 arquivo.write(f"{nome}\n")
 arquivo.close()
def leia_arquivo(nome_arquivo):
 global agenda, alterada
 arquivo = open(nome_arquivo, "r", encoding="utf-8")
 agenda = []
 for l in arquivo.readlines():
 nome, telefone, email, aniversário = l.strip().split("#")
 agenda.append([nome, telefone, email, aniversário])
 arquivo.close()
 alterada = False
def lê():
global alterada
 if alterada:
 print(
 "Você não salvou a lista desde a última alteração. Deseja gravá-la
agora?"
 )
      if confirma("gravação") == "S":
```

```
 grava()
 print("Ler\n---")
 nome_arquivo = pede_nome_arquivo()
 leia_arquivo(nome_arquivo)
 atualiza_última(nome_arquivo)
def ordena():
 global alterada
 # Você pode ordenar a lista como mostrado no livro
 # com o método de bolhas (bubble sort)
 # Ou combinar o método sort do Python com lambdas para
 # definir a chave da lista
 # agenda.sort(key=lambda e: return e[0])
   fim = len(agenda)
   while fim > 1:
i = 0
 trocou = False
 while i < (fim - 1):
 if agenda[i] > agenda[i + 1]:
 # Opção: agenda[i], agenda[i+1] = agenda[i+1], agenda[i]
 temp = agenda[i + 1]
 agenda[i + 1] = agenda[i]
 agenda[i] = temp
 trocou = True
 i += 1
 if not trocou:
 break
 alterada = True
def grava():
 global alterada
 if not alterada:
 print("Você não alterou a lista. Deseja gravá-la mesmo assim?")
```

```
    if confirma("gravação") == "N":
 return
 print("Gravar\n\------")
 nome_arquivo = pede_nome_arquivo()
 arquivo = open(nome_arquivo, "w", encoding="utf-8")
 for e in agenda:
 arquivo.write(f"{e[0]}#{e[1]}#{e[2]}#{e[3]}\n")
 arquivo.close()
 atualiza_última(nome_arquivo)
 alterada = False
def valida_faixa_inteiro(pergunta, inicio, fim):
 while True:
 try:
 valor = int(input(pergunta))
 if inicio <= valor <= fim:
 return valor
 except ValueError:
 print("Valor inválido, favor digitar entre {inicio} e {fim}")
def menu():
 print("""
 1 - Novo
 2 - Altera
 3 - Apaga
 4 - Lista
  5 - Grava
 6 - Lê
 7 - Ordena por nome
 0 - Sai
""")
 print(f"\nNomes na agenda: {len(agenda)} Alterada: {alterada}\n")
 return valida_faixa_inteiro("Escolha uma opção: ", 0, 7)
```

```
lê_última_agenda_gravada()
while True:
 opção = menu()
 if opção == 0:
 break
 elif opção == 1:
 novo()
 elif opção == 2:
 altera()
 elif opção == 3:
 apaga()
 elif opção == 4:
 lista()
 elif opção == 5:
 grava()
 elif opção == 6:
 lê()
 elif opção == 7:
 ordena()
```