

Para solucionar o desafio, optei por utilizar a linguagem Python devido sua simplicidade na hora de se trabalhar com arquivos e por achar que essa seria a opção mais viável em um caso profissional. A minha solução conta com duas classes, a classe main, que é onde o programa é executado, e a classe Application, que é onde eu criei os métodos responsáveis pelas funcionalidades do programa.

Observações: Desenvolvi o programa na IDE Pycharm no sistema operacional windows, importando as bibliotecas utilizadas através da própria IDE. Link para o repositório no github:

Classe main:

Na classe main, comecei mostrando para o usuário todas as funcionalidades do programa, indicando um índice que deverá ser usado para chamar cada possível funcionalidade. Em seguida, criei uma entrada para que o usuário selecione seu tipo de consulta.

```
Bem vindo ao programa para consulta de bolsas.  
Digite o número para consultar a opção desejada:  
1- Consultar bolsa zero/ano:  
2- Codificar nome:  
3- Consultar media anual:  
4- Consultar ranking de valores de bolsa:  
0- Encerrar o programa.  
  
Digite o número de sua consulta:
```

Após selecionar a opção desejada, o usuário é apresentado diretamente à funcionalidade escolhida, para isso, usei uma sequência de ifs, que quando entram em ação, retornam a opção escolhida. Para chamar os métodos 1,2,3 e 4, criei um objeto derivado da classe Application. Para encerrar o programa, conforme a opção 0, envolvi desde a apresentação das funcionalidades até as estruturas condicionais em uma função chamada program, assim quando o programa precisa ser finalizado, a função simplesmente executa um “return”. Caso a entrada seja inválida, o programa envia uma mensagem informando o erro ao terminal e executa a função program novamente.

Classe Application:

Para a classe Application, importei as bibliotecas csv e random, para ler o arquivo e embaralhar saídas, respectivamente.

A classe começa com a função __init__, que recebe uma lista chamada data e uma chamada da função read_csv(), assim, o arquivo csv é lido logo que o objeto é instanciado na classe main.

Função read_csv:

Na função read csv, começo abrindo o arquivo e o salvando em uma variável chamada csv_file, em seguida, crio uma variável chamada csv_delimited, onde com a ajuda da biblioteca csv defino que cada item é separado por um “;”. Por fim, uso um for para armazenar cada linha do arquivo csv na lista data criada dentro da função __init__ e fecho o arquivo.

Função search_one:

Quando a função é chamada, ela primeiramente apresenta um input ao usuário, que deve informar o ano que deseja realizar a busca. Em seguida, utilizo um for para percorrer cada linha da lista data, onde comparo se o ano de cada linha da coluna “AN_REFERENCIA” é igual ao ano informado pelo usuário. Caso seja, o programa informa o nome, cpf, entidade de ensino e valor da bolsa do primeiro bolsista que encontrar no ano informado. Caso seja informado um ano inexistente no arquivo, o programa mostra uma mensagem e executa novamente a função, para que o usuário tente novamente.

Execução da função:

Primeiro o usuário deve escolher a opção número 1:

```
Bem vindo ao programa para consulta de bolsas.  
Digite o número para consultar a opção desejada:  
1- Consultar bolsa zero/ano:  
2- Codificar nome:  
3- Consultar media anual:  
4- Consultar ranking de valores de bolsa:  
0- Encerrar o programa.  
  
Digite o número de sua consulta: 1|
```

Em seguida, o usuário deve informar o ano de consulta:

```
Informe o ano que deseja buscar o bolsista 0: 2016
```

Caso o ano seja válido, apresenta os dados do primeiro bolsista do ano:

```
Informe o ano que deseja buscar o bolsista 0: 2016  
Nome: ALEXANDRE RIBEIRO NETO  
cpf: ***.195.647-**  
Entidade de ensino: UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
Valor da bolsa: R$ 765
```

Caso contrário, mostra uma mensagem e permite ao usuário tentar novamente.

Ex: o ano de 2011 é inválido:

```
Informe o ano que deseja buscar o bolsista 0: 2011
Ano não disponível, tente novamente.

Informe o ano que deseja buscar o bolsista 0:
```

Função `search_two`:

Quando a função é chamada, apresenta um input ao usuário, que deve entrar com um nome, podendo ser apenas parte dele. Caso o nome seja válido, o programa retorna o nome criptografado, ano, entidade de ensino e valor da bolsa.

Execução do programa ao selecionar a opção número 2 e digitando um nome incompleto:

```
Digite o número para consultar a opção desejada:
1- Consultar bolsa zero/ano: |
2- Codificar nome:
3- Consultar media anual:
4- Consultar ranking de valores de bolsa:
0- Encerrar o programa.

Digite o número de sua consulta: 2

Digite o nome a ser codificado: cleb
```

Dados informados:

```
Dados:
Nome:  PEOBBFSFJDOSFBWTMUFCMJ
Ano:  2016
Entidade de ensino:  UNIVERSIDADE FEDERAL DE SERGIPE
Valor da bolsa:  1100
```

Para criptografar o nome, o programa verifica se existe um nome igual ao digitado, ou que comece com os caracteres digitados, usando a função do python `startswith()`. Caso encontre algum nome, armazena cada caractere em uma lista e a embaralha. Em seguida, pega cada caractere do nome embaralhado e encontra sua posição em uma lista que contém todas as letras do alfabeto, trocando a letra atual pela letra seguinte, exemplo c -> d. Por fim, mostra o

nome criptografado e os outros dados do usuário como pode ser visualizado na imagem acima.

Função search_three:

Quando a função é chamada, apresenta um input ao usuário, que deve informar o ano que deseja visualizar o valor médio das bolsas. Para isso, o programa compara cada linha do arquivo na coluna de anos, e caso o ano seja igual ao informado, acumula cada valor em uma variável chamada media_sum. Por fim, divide o valor da variável media_sum pelo número de elementos encontrados no ano informado, retornando uma mensagem com o valor médio do respectivo ano, arredondado em duas casas decimais.

Execução do programa:

Selecione a opção 3 no menu e informe um ano para busca:

```
Bem vindo ao programa para consulta de bolsas.  
Digite o número para consultar a opção desejada:  
1- Consultar bolsa zero/ano:  
2- Codificar nome:  
3- Consultar media anual:  
4- Consultar ranking de valores de bolsa:  
0- Encerrar o programa.  
  
Digite o número de sua consulta: 3  
  
Selecione o ano para consultar a media de valor das bolsas: 2016
```

Informa a média do valor das bolsas para aquele ano:

```
Média de valor das bolsas de 2016 R$ 966.9
```

Função search_four:

Quando a função é chamada, o programa exibe um ranking com os três maiores e menores valores de bolsa. Para isso, percorre cada linha do arquivo na coluna “VL_BOLSISTA_PAGAMENTO”, buscando pelos menores valores disponíveis e armazena as três primeiras aparições do menor valor possível em duas listas separadas. Por fim, mostra na tela do terminal um ranking com os dados dos bolsistas com maiores e menores bolsas.

Execução do programa:

Selecione a opção 4:

```
Bem vindo ao programa para consulta de bolsas.  
Digite o número para consultar a opção desejada:  
1- Consultar bolsa zero/ano:  
2- Codificar nome:  
3- Consultar media anual:  
4- Consultar ranking de valores de bolsa:  
0- Encerrar o programa.  
  
Digite o número de sua consulta: 4
```

Mostrando os maiores valores:

```
MAIORES VALORES:  
Nome: LIAMARA SCORTEGAGNA  
Valor da bolsa: R$ 1500  
  
Nome: CESAR ALFREDO CARDOSO  
Valor da bolsa: R$ 1500  
  
Nome: IZABEL PATRICIA MEISTER  
Valor da bolsa: R$ 1500
```

e logo abaixo, os menores:

```
MENORES VALORES:  
Nome: ALEXANDRE RIBEIRO NETO  
Valor da bolsa: R$ 765  
  
Nome: LUIZ GUSTAVO RIBEIRO ROLANDO  
Valor da bolsa: R$ 765  
  
Nome: ANTONIO RODRIGUES DE ARAUJO  
Valor da bolsa: R$ 765
```

Opção 0(fechar o programa):

Caso o usuário selecione a opção 0 ao executar o programa, a função que exibe o menu e redireciona para os métodos é finalizada através de um return.

Digite o número para consultar a opção desejada:

1- Consultar bolsa zero/ano:

2- Codificar nome:

3- Consultar media anual:

4- Consultar ranking de valores de bolsa:

0- Encerrar o programa.

Digite o número de sua consulta: 0

Programa encerrado!