Folha Prática 4
ALGORITMOS E ESTRUTURA DE DADOS **Ana Mafalda Martins** **2018/2019**

Filas (*Queues*) e Filas Duplamente Terminadas (*Deques*)

1. Implemente a TAD Fila (*Queue*), tal como explicado na aula teórica.
2. Implemente a TAD Fila (*Queue*), usando uma lista (*List*) de forma que a parte detrás da fila (*rear*) esteja no final da lista.
3. Simulação do jogo *Hot Potato*
   a) Implemente simulação do jogo *Hot Potato*, tal como explicado na aula teórica.
   b) Altere a simulação do jogo *Hot Potato*, implementado em 3.1, para que os nomes dos jogadores e o valor da contagem (num) sejam dados pelo utilizador.

      Observação: o valor da contagem tem de ser um inteiro positivo. Se o utilizador não inserir um inteiro positivo, o programa deve dar um aviso e deve pedi-lo novamente.
   c) Modifique a simulação do jogo *Hot Potato* de forma a que valor de contagem seja escolhido aleatoriamente em cada passagem do jogo.
4. Implemente a TAD Fila Duplamente Terminada (Deque), tal como explicado na aula teórica.
5. Verificador de palíndromo (*palindrom checker*)
   a) Implemente o verificador de palíndromo, conforme descrito na aula teórica.
   b) Estenda o verificador de palíndromo para lidar com palíndromos com espaços. Por exemplo, “I PREFER PI” é um palíndromo que se lê da mesma forma para frente e para trás se ignorar os caracteres em branco.
6. Simulação de tarefas de impressão.
   a) Implemente a simulação de tarefas de impressão descrita no livro.
   b) Modifique a implementação anterior de forma:
      i. a refletir um número maior de alunos. Suponha que o número de alunos dobrou.
      ii. a refletir um número menor de tarefas de impressão. Suponha que o número médio de tarefas de impressão é diminuído em metade.
      iii. a que o número de alunos seja um parâmetro da simulação.

7. Adicione o seguinte método à classe *Queue* implementada no exercício 1:

```
def __str__(self):
    string = ""
    for item in self.items:
        string = string + item + " "

    return string
```

8. Faça o download do ficheiro ex08.py. Implemente a função `process` que processa os elementos que vêm de uma fila `q`, na forma de um nome seguido de uma operação. Consoante a operação deve fazer o seguinte:

   - **Nome A** - Adiciona nome à fila `a`
   - **Nome B** - Adiciona nome à fila `b`
   - **Nome X** - Adiciona nome à fila que tenha menos elementos (`a` ou `b`). Se ambas as filas tiverem o mesmo número de elementos, o nome é descartado e não é adicionada a nenhuma.

   Por exemplo, se a fila `q` for

Fim											Início
Miguel	B	Jose	X	Joao	X	Luisa	A	Pedro	A	Luis	B

   Luis B - Luis é adicionado à fila `b`
   Pedro A - Pedro é adicionado à fila `a`
   Luisa A - Luisa é adicionada à fila `a`
   Joao X - Joao é adicionado à fila `b`, que tem apenas 1 elemento (Luis) contra os dois da fila `b` (Pedro e Luisa)
   Jose X - José descartado (ambas as filas têm 2 elementos)
   Miguel B - Miguel é adicionado à fila `b`

   No fim a fila `a` fica com [Pedro, Luisa] e a fila `b` fica com [Luis, Joao, Miguel]. A fila `q` deve ficar vazia.

   Dicas:

   a. A ideia é ir retirando elementos à fila `q` (dois a dois) enquanto a não estiver vazia.
   b. Consoante a operação, devemos colocar logo numa certa lista, ou comparar os tamanhos.

Exemplo de input/output 1

```
Número de nomes --> 6
Nome e Operação (separados por espaço) --> Luis B
Nome e Operação (separados por espaço) --> Pedro A
Nome e Operação (separados por espaço) --> Luisa A
Nome e Operação (separados por espaço) --> Joao X
Nome e Operação (separados por espaço) --> Jose X
Nome e Operação (separados por espaço) --> Miguel B
Início:
q:  Miguel B Jose X Joao X Luisa A Pedro A Luis B
a:
b:
--------------------------------
Final:
q:
a: Luisa Pedro
b: Miguel Joao Luis
```

Exemplo de input/output 2

```
Número de nomes --> 4
Nome e Operação (separados por espaço) --> Luis B
Nome e Operação (separados por espaço) --> Pedro B
Nome e Operação (separados por espaço) --> Luisa X
Nome e Operação (separados por espaço) --> Joao X
Início:
q:  Joao X Luisa X Pedro B Luis B
a:
b:
--------------------------------
Final:
q:
a: Joao Luisa
b: Pedro Luis
```

9. Faça o download do ficheiro ex08.zip. Implemente a função `merge(a, b)`, `a` e `b` são variáveis do tipo fila (*queue*).

Assumindo que as filas `a` e `b` estão ordenadas de forma crescente, esta função deve criar e devolver uma nova fila (`merge`) que é a união ordenada das duas filas.

Dicas:

a. Comece por criar uma nova fila de inteiros que irá conter a união
b. Depois, a ideia é "espreitar" o primeiro elemento de cada fila e escolher o que for menor.

Exemplo de input/output 2

```
Elementos da primeira fila (ordenados e separados por espaço) --> 2 4 8 10
Elementos da segunda fila (ordenados e separados por espaço) --> 1 4 9
a:  2 4 8 10
b:  1 4 9 [redacted]
merge = 1 2 4 4 8 9 10
```

10. Considere uma situação da vida real. Formule uma pergunta e crie uma simulação que o ajude a respondê-la. Situações possíveis incluem:

    Carros numa fila numa máquina de lavagem de carros
    Clientes numa fila de supermercado
    Aviões a decolar e a pousar numa pista