

# Ordenação de Panquecas

---

As panquecas são um prato típico da culinária norte americana. Vendo esse alimento em diversos filmes e séries, você notou que a maioria das panquecas não possuem um mesmo padrão em relação ao diâmetro.

Uma pilha de panquecas pode ser representada por uma sequência de números, de tal forma que o primeiro número representa o diâmetro da panqueca no topo da pilha, o segundo número representa o diâmetro da panqueca logo abaixo daquela, e assim por diante. Sendo assim, para formar uma pirâmide de panquecas (ou seja, arranjar as panquecas de tal forma que toda panqueca esteja em cima de uma panqueca com diâmetro maior ou igual a ela) é necessário ordenar a sequência que representa a pilha de panquecas.

Você notou que alguns cozinheiros possuem uma grande habilidade para formar uma pirâmide de panquecas, fazendo três tipos de operações:

1. Encontrar a maior panqueca ainda não posicionada corretamente;
2. Inverter a ordem das panquecas, do topo da pilha até a posição da panqueca encontrada no passo 1, deixando essa panqueca na primeira posição da sequência (topo da pilha);
3. Inverter a ordem das panquecas, do topo da pilha até a posição onde a panqueca encontrada no passo 1 deve ser posicionada (logo acima da panqueca que possui diâmetro maior do que ela).

Como você sabe programar, decidiu fazer um programa que mostra o passo a passo da ordenação de panquecas. Para cada inversão (tanto para a segunda quando para a terceira etapa), seu código deverá mostrar como está a sequência atual. Além disso, você pode considerar que não existem panquecas com o mesmo diâmetro.

O exemplo a seguir mostra o passo a passo da ordenação de panquecas. Os valores indicam os diâmetros das panquecas.

3 5 2 1 7 6 4

A maior panqueca ainda não posicionada corretamente é a que possui diâmetro 7, na posição 5 da sequência. Portanto, inverte-se a sequência da primeira posição até

a posição 5.

7 1 2 5 3 6 4

Depois, inverte-se a sequência inteira, de forma a posicionar a maior panqueca na última posição da sequência (fundo da pilha).

4 6 3 5 2 1 7

Na sequência, a maior panqueca não posicionada corretamente (aquela de diâmetro 6) é encontrada na posição 2. Os mesmo passos são seguidos até o final da ordenação.

Suponha que o algoritmo encontre a sequência da seguinte maneira:

4 3 1 2 5 6 7

onde a maior panqueca fora de posição (aquela com diâmetro 4) já está no topo da pilha, logo a inversão do passo 2 não deve ser realizada. Ao realizar a inversão do passo 3, que posicionará a panqueca com diâmetro 4 logo acima da panqueca de diâmetro 5, a panqueca com diâmetro 3 também será posicionada corretamente:

2 1 3 4 5 6 7

Portanto, a ordenação seguirá para a panqueca de diâmetro 2 e com apenas mais uma inversão a sequência será ordenada:

1 2 3 4 5 6 7

Caso a sequência já esteja ordenada inicialmente, seu programa deverá imprimir "Panquecas ja ordenadas" e nenhuma operação deve ser feita.

Exemplos de entradas e saídas esperadas pelo seu programa:

## Teste 01

### Entrada

3 5 2 1 7 6 4

## Saída

```
Posicionando a panqueca 7
Primeira inversao: 7 1 2 5 3 6 4
Segunda inversao: 4 6 3 5 2 1 7
Posicionando a panqueca 6
Primeira inversao: 6 4 3 5 2 1 7
Segunda inversao: 1 2 5 3 4 6 7
Posicionando a panqueca 5
Primeira inversao: 5 2 1 3 4 6 7
Segunda inversao: 4 3 1 2 5 6 7
Posicionando a panqueca 4
Segunda inversao: 2 1 3 4 5 6 7
Posicionando a panqueca 2
Segunda inversao: 1 2 3 4 5 6 7
```

## Teste 02

### Entrada

```
1 2 3 4 5 6 7
```

### Saída

```
Panquecas ja ordenadas
```

## Teste 03

### Entrada

```
7 6 5 4 3 2 1
```

### Saída

```
Posicionando a panqueca 7
Segunda inversao: 1 2 3 4 5 6 7
```

## Código Base

---

No arquivo auxiliar lab13.py você irá encontrar um código base para dar início ao processo de elaboração desta tarefa. Para facilitar a implementação do seu programa, o código base contém uma função para auxiliar na impressão da sequência que representa a pilha de panquecas, conforme mostrado a seguir:

```
def str_panquecas(lista):  
    return(" ".join(map(str, lista)))
```

## Orientações

---

- Veja [aqui](#) a página de submissão da tarefa.
- O arquivo a ser submetido deve se chamar lab13.py.
- No link "Arquivos auxiliares" há um arquivo compactado (aux13.zip) que contém todos os arquivos de testes abertos (entradas e saídas esperadas).
- O laboratório é composto de 10 testes abertos e 10 testes fechados.
- O limite máximo será de 20 submissões.
- Acesse o sistema SuSy com seu RA (apenas números) e a senha que você utiliza para fazer acesso ao sistema da DAC.
- Você deve seguir as instruções de submissão descritas no enunciado.
- Serão considerados apenas os resultados da última submissão.
- Esta tarefa tem peso 3.
- O prazo final para submissão é dia 28/11/2021 (domingo).