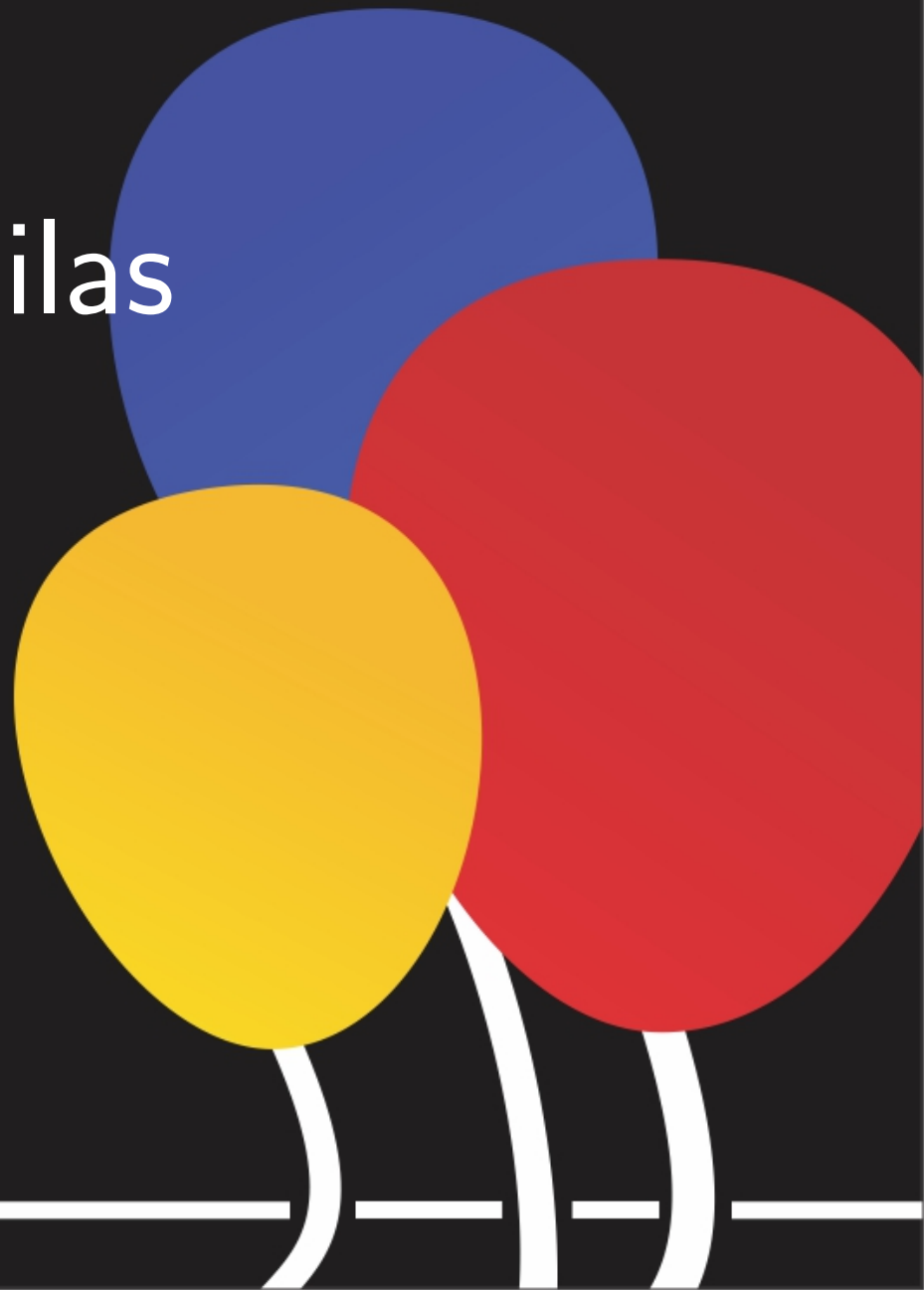
MARATONANDO

Listas, Pilhas e Filas

Vinicius Gasparini
Ex-Bolsista
v.gasparini@edu.udesc.br



Maratona de Programação
UDESC
Departamento de Ciência da Computação

Sumário

Introdução

- Alguém saberia me explicar o que é uma lista?
- A diferença entre simplesmente encadeada e duplamente encadeada

Lista

- É um conjunto de **objetos** que contém duas propriedades
- Por sua vez, cada objeto possui outras duas propriedades

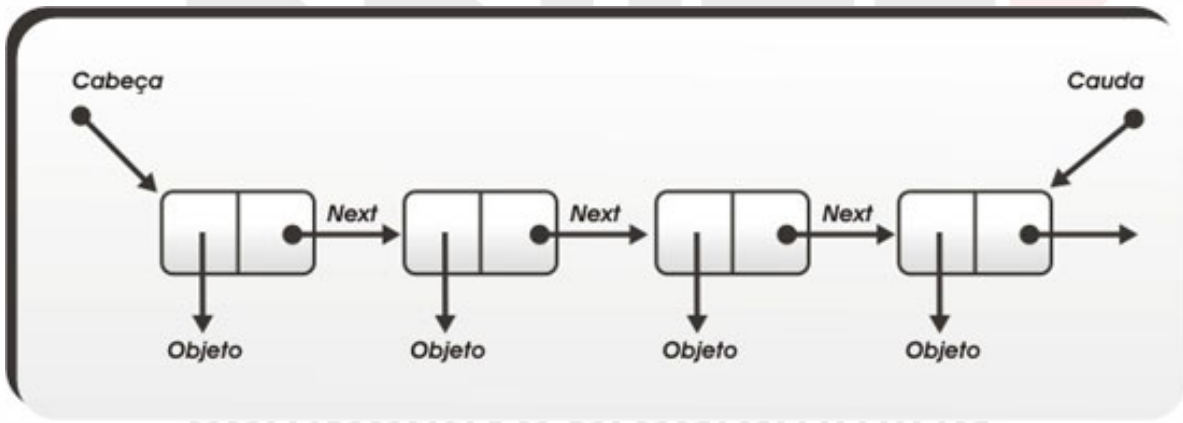


Figura: A tal da lista simplesmente encadeada

Elemento da Lista

```python
class Elemento:
    dado = None
    anterior = None
    proximo = None

    def __init__(self, dado, anterior=None, proximo=None):
        self.dado = dado
        self.anterior = anterior
        self.proximo = proximo
```

Lista

```python
class Lista:
    cabeca = None
    cauda = None

    def __init__(self, elemento):
        self.cabeca = elemento
        self.cauda = elemento

    def inserir_no_meio(self, elemento, posicao):
        contador = 0
        elemento_atual = self.cabeca
        while (contador != posicao):
            elemento_atual = elemento_atual.proximo
            contador += 1
        elemento.proximo = elemento_atual
        elemento_atual.anterior.proximo = elemento
        elemento_atual.anterior = elemento
```

```python
    def inserir_no_fim(self, elemento):
        elemento.anterior = self.cauda
        self.cauda.proximo = elemento
        self.cauda = elemento

    def inserir_no_inicio(self, elemento):
        elemento.proximo = self.cabeca
        self.cabeca.anterior = elemento
        self.cabeca = elemento
```

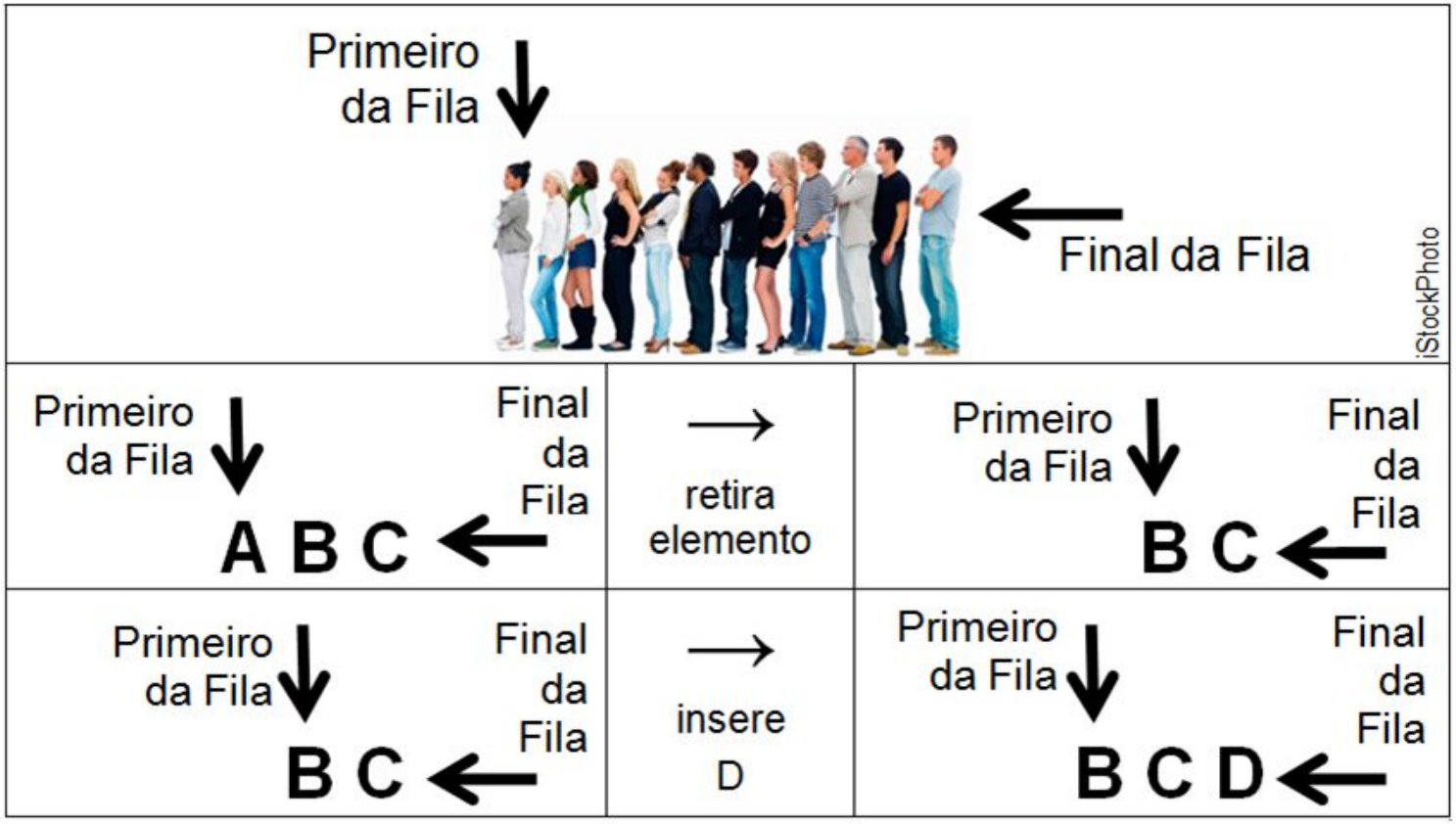
Lista

Aplicações de Listas

- Tudo
- Árvore
- Grafos
- Matrizes
- Pilhas e Filas

Fila

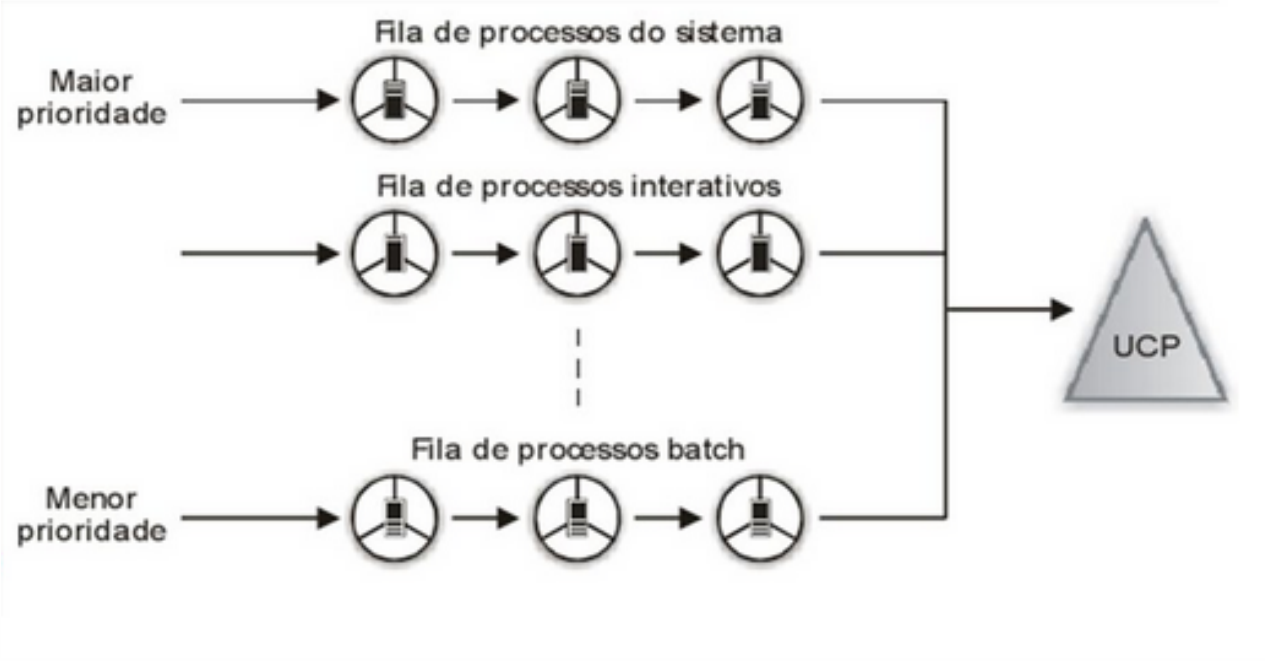
Retirando e Inserindo Elementos em uma Fila



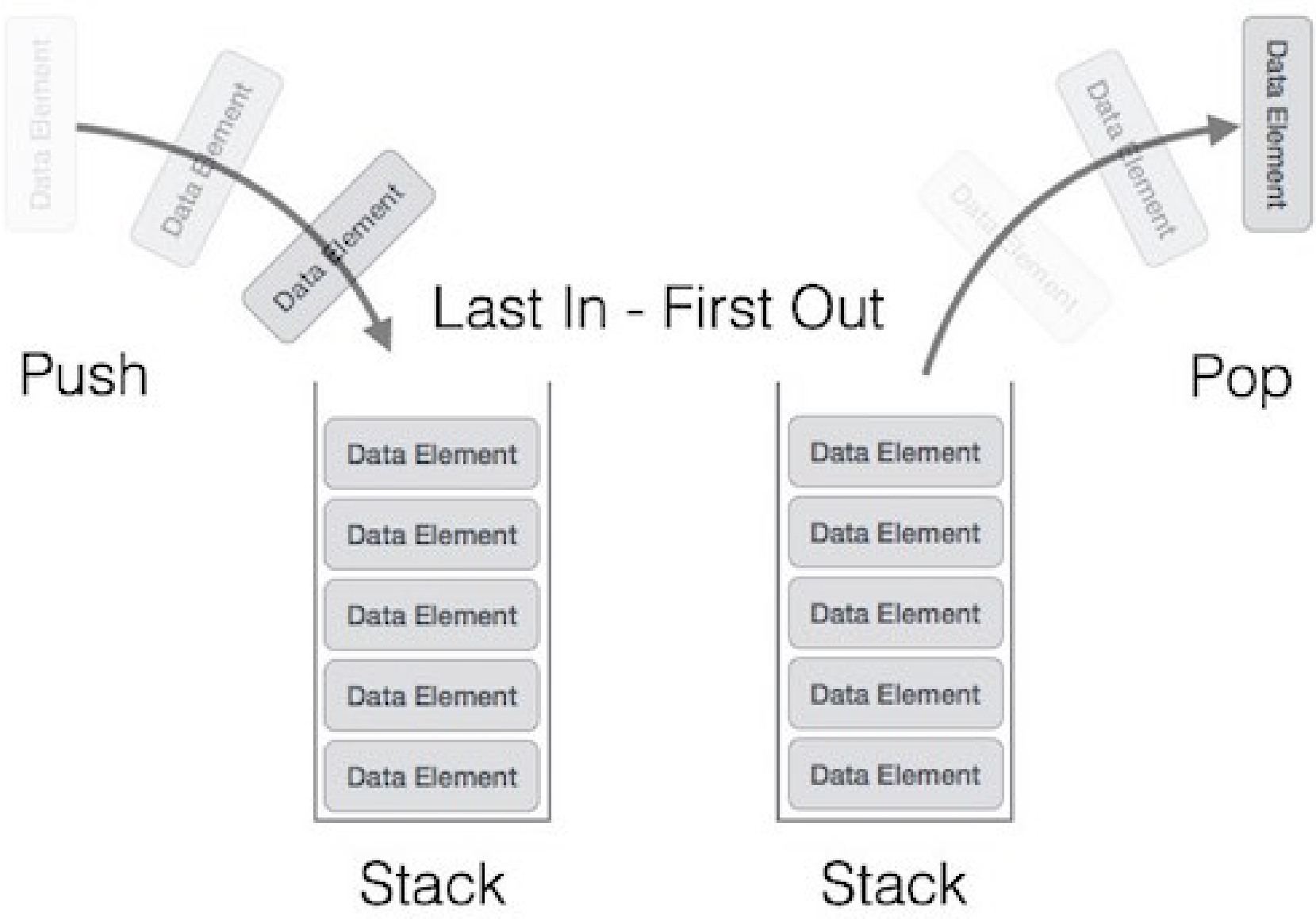
Fila

```python
class Fila:
    def __init__(self, elemento=None):
        self.fila = []
        if elemento:
            self.inserir(elemento)

    def inserir(self, elemento):
        self.fila.append(elemento)

    def remover(self):
        elemento = self.fila.pop(0)
        return elemento

    def tamanho(self):
        contador = 0
        for elemento in self.fila:
            contador += 1
        return contador

    def vazia(self):
        return self.tamanho() == 0

    def frente(self):
        return self.fila[0]
```

Fila


Fila de processos do sistema
Maior prioridade
Fila de processos interativos
UCP
Fila de processos batch
Menor prioridade

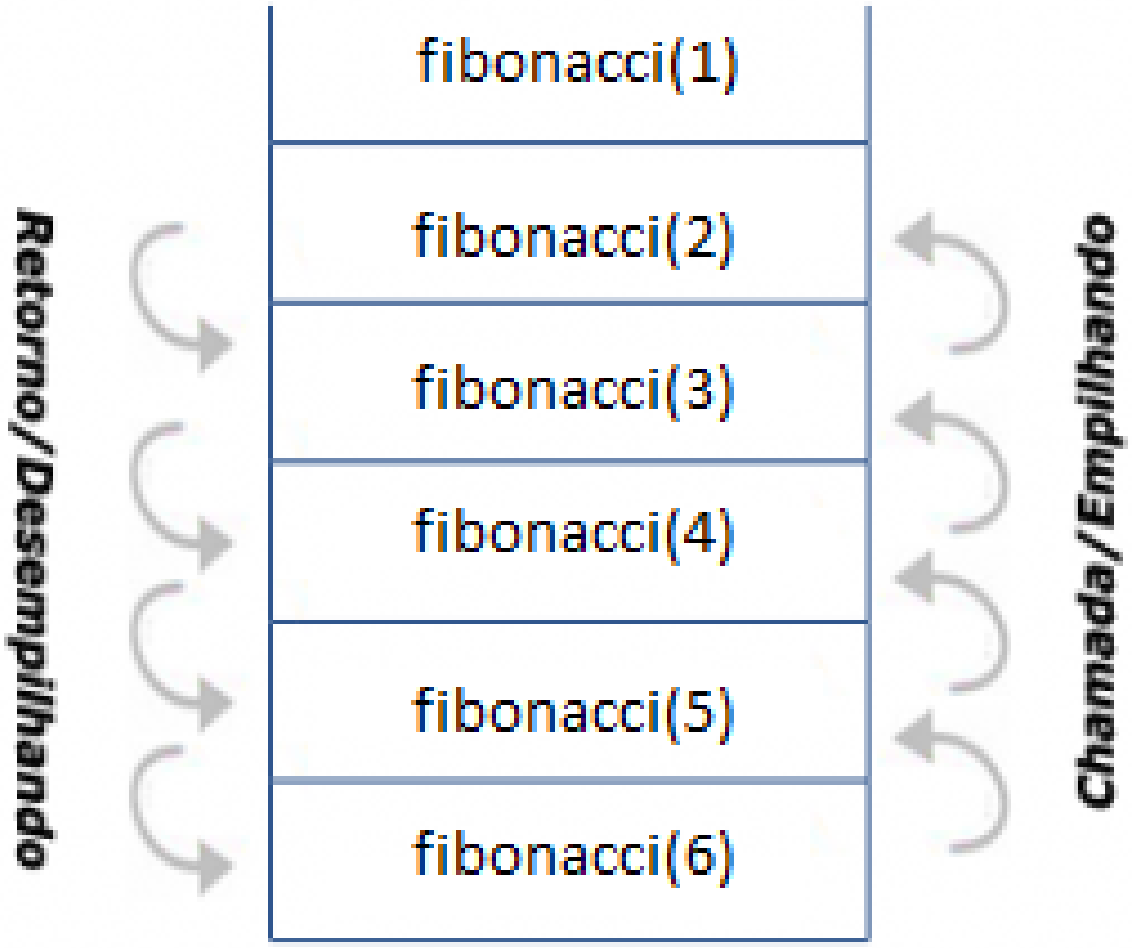
Pilha


Data Element
Data Element
Data Element
Last In - First Out
Push
Pop
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Stack
Stack

Pilha

```python
class Pilha:
    def __init__(self, elemento=None):
        self.pilha = list()
        if elemento:
            self.inserir(elemento)

    def inserir(self, elemento):
        self.pilha.insert(0, elemento)

    def remover(self):
        elemento = self.pilha.pop(-1)
        return elemento

    def tamanho(self):
        contador = 0
        for elemento in self.pilha:
            contador += 1
        return contador

    def vazia(self):
        return self.tamanho() == 0

    def topo(self):
        return self.pilha[-1]
```

Pilha


fibonacci(1)
fibonacci(2)
fibonacci(3)
fibonacci(4)
fibonacci(5)
fibonacci(6)
Retorno/Desempilhando
Chamada/Empilhando

URI 1069 - Diamantes e Areia

URI Online Judge | 1069

Diamantes e Areia

Por Neilor Tonin, URI Brasil

Timelimit: 1

João está trabalhando em uma mina, tentando retirar o máximo que consegue de diamantes "<>". Ele deve excluir todas as particulas de areia "." do processo e a cada retirada de diamante, novos diamantes poderão se formar. Se ele tem como uma entrada .<...<<..>>....>....>>>., três diamantes são formados. O primeiro é retirado de <..>, resultando .<...<>....>....>>>. Em seguida o segundo diamante é retirado, restando .<.....>....>>>. O terceiro diamante é então retirado, restando no final .....>>>., sem possibilidade de extração de novo diamante.

Entrada

Deve ser lido um valor inteiro **N** que representa a quantidade de casos de teste. Cada linha a seguir é um caso de teste que contém até 1000 caracteres, incluindo "**<,>, .**"

Saída

Você deve imprimir a quantidade de diamantes possíveis de serem extraídos em cada caso de entrada.

URI 1069 - Diamantes e Areia

Exemplo de Entrada	Exemplo de Saída
2 <..><.<..>> <<<..<.....<<<<....>	3 1

Baseado no problema "Caçador de Diamantes"

URI 1110 - Jogando Cartas Fora

URI Online Judge | 1110

Jogando Cartas Fora

Folclore, adaptado por Piotr Rudnicki Canada

Timelimit: 1

Dada uma pilha de *n* cartas enumeradas de 1 até *n* com a carta 1 no topo e a carta *n* na base. A seguinte operação é ralizada enquanto tiver 2 ou mais cartas na pilha.

Jogue fora a carta do topo e mova a próxima carta (a que ficou no topo) para a base da pilha.

Sua tarefa é encontrar a sequência de cartas descartadas e a última carta remanescente.

Cada linha de entrada (com exceção da última) contém um número $n \leq 50$. A última linha contém 0 e não deve ser processada. Cada número de entrada produz duas linhas de saída. A primeira linha apresenta a sequência de cartas descartadas e a segunda linha apresenta a carta remanescente.

Entrada

A entrada consiste em um número indeterminado de linhas contendo cada uma um valor de 1 até 50. A última linha contém o valor 0.

Saída

Para cada caso de teste, imprima duas linhas. A primeira linha apresenta a sequência de cartas descartadas, cada uma delas separadas por uma vírgula e um espaço. A segunda linha apresenta o número da carta que restou. Nenhuma linha tem espaços extras no início ou no final. Veja exemplo para conferir o formato esperado.

URI 1110 - Jogando Cartas Fora

Exemplo de Entrada	Exemplo de Saída
7 19 10 6 0	Discarded cards: 1, 3, 5, 7, 4, 2 Remaining card: 6 Discarded cards: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 8, 12, 16, 2, 10, 18, 14 Remaining card: 6 Discarded cards: 1, 3, 5, 7, 9, 2, 6, 10, 8 Remaining card: 4 Discarded cards: 1, 3, 5, 2, 6 Remaining card: 4

Dúvidas?

Implementações apresentadas