

## Implementação de algoritmos de substituição de páginas

Neste projeto você deve escrever um programa para simular o funcionamento dos seguintes algoritmos de substituição de páginas:

- **Segunda Chance** - Uma modificação simples para o FIFO que evita o problema de jogar fora uma página intensamente usada é inspecionar o bit de referência da página mais antiga. **Considere que o bit R de todas as páginas é zerada a cada 4(quatro) referências à memória.**
- **Algoritmo Ótimo** - Cada página deve ser rotulada com o número de instruções que serão executadas antes de aquela página ser referenciada pela primeira vez. O algoritmo ótimo diz que a página com o maior rótulo deve ser removida.
- **Conjunto de trabalho** - encontra uma página que não esteja no conjunto de trabalho e removê-la. Para isso, o sistema mantém: o instante do último uso para cada página, o tempo virtual atual(incrementado a cada referência a memória) e um limiar que deve ser sempre metade do número de molduras de página da memória mais 1(um).Ex: Se  $n=4$ , então  $\text{limiar}=4/2+1=3$ . **Considere que o bit R de todas as páginas é zerada a cada 4(quatro) referências à memória.**

O programa deverá ler da entrada padrão um conjunto de número inteiros onde o primeiro número representa a quantidade de molduras de página disponíveis na RAM e os demais representam a sequência de referências às páginas, sempre um número por linha.

O programa deverá imprimir na saída o número de faltas de páginas obtido com a utilização de cada um dos algoritmos.

**Descrição da Entrada:** A entrada é composta por uma série números inteiros, um por linha, indicando, primeiro a quantidade de quadros disponíveis na memória RAM e, em seguida, a sequência de referências à memória.

### Exemplo de entrada:

4  
1  
2  
3  
4  
1  
2  
5  
1  
2  
3  
4  
5

**Descrição da saída:** A saída é composta por linhas contendo a sigla de cada um dos três algoritmos e a quantidade de faltas de página obtidas com a utilização de cada um deles.

**Exemplo de saída:**

```
SC 7
OTM 6
CT 8
```

**Importante:** A corretude do seu programa será testada automaticamente. Portanto, o programa deve ler a entrada e imprimir a saída EXATAMENTE como no formato especificado acima. A impressão de qualquer outro caractere ou de dados fora da ordem solicitada implicará em uma saída incorreta.