# Problema D

# O Cliente é um Chato Versão 1

## Problem

Considere que determinada marca de automóveis produz modelos cujos nomes são números. E.g.,

```
·       Pesla Model 1
.       Pesla Model 2
·       ...
·       Pesla Model 10
·
```

Até à data, a Pesla já tem 10 modelos diferentes, todos com muito sucesso no mercado.

Quando um cliente quer comprar um Pesla, pode ir até ao concessionário da marca (só existe um único concessionário no planeta Terra; o outro é na lua), e ver os carros em montras com 4 carros. O único problema é que, de facto, só cabem 4 carros de cada vez nas montras. Para se verem todos os modelos, os carros têm de ser trocados (saí um e entra outro). O sistema funcionaria bem se cada cliente visitante se limitasse a ver os carros uma única vez. Contudo, há muitos clientes que pedem para ver alguns modelos várias vezes, mesmo depois destes já terem saído para a garagem. Por exemplo, ainda no outro dia um fulano vinha com a ideia de ver os veículos na seguinte ordem:

Pesla Model 1 Model 2 3 4 2 4 10 8 7 6 1 5 9

O vendedor já desenvolveu uma úlcera nervosa por causa deste detalhe. Assim, dirigiu-se à UBI e ao curso de Engenharia Informática no sentido de lhe fazerem um programa que testasse diferentes estratégias de substituição de automóveis, só para ele ver se há forma de otimizar o procedimento. As estratégias que ele gostava de testar são:

- Least Recently Used (substituir sempre o carro visto à mais tempo);
- Aleatório (Random);
- Primeiro a entrar, primeiro a sair (First In First Out).

O que o vendedor gostava mesmo de ver era o número de vezes que tem de trocar carros para cada uma das estratégias.

Já agora, se não for pedir muito, o vendedor gostava de ver quantas vezes teria de trocar carros se tivesse a estratégia ideal, i.e., aquela em que já sabia quais eram os pedidos que o cliente ía fazer antes mesmo deste os fazer.

## Input

O input do programa é um inteiro N numa linha seguido por N números inteiros (N<100), estritamente entre 1 (inclusive) e 10 (inclusive)e cada um numa linha separada.

Nota importante 1: considere sempre que a montra tem, no início, os modelos 1, 2, 3 e 4.

Nota importante 1: não use o gerador de números psuedo aleatórios do standard library do C para os testes. Use, para este caso, o seguinte gerador myrand() que gera a seguinte sequencia : 3 1 0 0 2 0 3 1 3 2 2 3 3 2 1

```
int myrand () {
  static int valor = 45;
  valor = 73 * valor + 2713;
  valor = (valor % 101);
  return (valor%4);
}
```

Quem utilize outra linguagem de programação terá que converter esta função para uma equivalente.

## Output

O programa deve mostrar, em 4 linhas (separadas por ), o nome da estratégia seguido por "espaço:espaço" e o número de trocas para cada estratégia. O formato é o seguinte:

```
LRU : n1\n
Rand : n2\n
FIFO : n3\n
IDEAL : n4\n
```

## Samples

### Sample Input 1

```
12
1
2
3
4
5
6
7
8
9
10
1
2
```

### Sample Output 1

```
LRU : 8
Rand : 8
FIFO : 8
IDEAL : 6
```

### Sample Input 2

```
4
1
2
3
4
```

### Sample Output 2

```
LRU : 0
Rand : 0
FIFO : 0
IDEAL : 0
```

### Sample Input 3

```
12
1 1 3 2 4 5 6 7 8 9 10 3  (na mesma linha apenas pela aspecto estético)
```

### Sample Output 3

```
LRU : 7
Rand : 7
FIFO : 7
IDEAL : 6
```

### Sample Input 4

```
12
1 1 3 2 4 5 6 7 8 9 10 3  (na mesma linha apenas pela aspecto estético)
```

### Sample Output 4

```
LRU : 7
Rand : 7
FIFO : 7
IDEAL : 6
```

### Sample Input 5

```
20
1 2  3 4  5 6 7 8 9 10 1 8 9 4 8 9 4 2 7 8
```

### Sample Output 5

```
LRU : 11
Rand : 11
FIFO : 12
IDEAL : 9
```

### Sample Input 6

```
7
1
1
1
5
2
1
6
```

### Sample Output 6

```
RU : 3
Rand : 2
FIFO : 3
IDEAL : 2

Executando a solução do Professor em modo "debug" para mostrar as trocas:

Inicialmente para cada Algoritmo temos a situação [ 1 2 3 4] neste ordem

[vmprof@localhost pe]$ ./a.out < 6.in
 1  5  3  4
```

```
 1  5  2  4
 1  5  2  6
LRU : 3
(rn<- 3)  1  2  3  5
(rn<- 1)  1  6  3  5
Rand : 2
 5  2  3  4
 5  1  3  4
 5  1  6  4
FIFO : 3
 1  2  5  4   (Podia ter escolhido entre 3 e 4 -escolhe-se o primeiro. Ou seja 3)
 6  2  5  4
IDEAL : 2
```