

# Vetores Idênticos

João e seu amigo **pé de feijão** procuram por elementos parecidos. Um certo dia, João encontrou dois conjuntos de números, o **pé de feijão** achou que eles pudessem ser idênticos (isto é, que possuindo o mesmo conjunto de números). No entanto, ambos não conseguem identificar rapidamente esta suposição, para tanto pediram SUA ajuda para implementar um programa PARALELO que consiga ler os dois conjuntos de números e identificar se eles são idênticos ou não.

## Entrada

A entrada possui um único caso de teste. A primeira, do caso de teste, possui um inteiro  $N$  ( $2 \leq N \leq 10000000$ ) representando os tamanhos dos vetores. A segunda linha, do caso teste possui  $N$  inteiros  $V_i^1$  ( $0 \leq V_i^1 \leq 2000000000$ ) (cabe em um int de 32 bits), e a terceira linha possui  $N$  inteiros  $V_i^2$  ( $0 \leq V_i^2 \leq 2000000000$ ).

## Saída

A saída possui uma única linha contendo a frase **Diferentes**, caso os vetores não sejam idênticos (depois de ordenados), ou **Mesmos elementos** caso os vetores sejam idênticos (depois de ordenados).

## TAREFA

Implementar a solução utilizando o conceito de threads. O algoritmo sequencial (*e ineficiente*) está representado abaixo:

```
1 int main(void)
2 {
3     le_tamvetores(TAM)
4     le_v1()
5     le_v2()
6     Ordena(v1)
7     Ordena(v2)
8     Compara(v1,v2)
9     if iguais; then
10         print "Mesmos elementos"
11     else
12         print "Diferentes"
13     return 0;
14 }
```

Para agilizar, você poderá utilizar a função `qsort(3)` (da libc) para ordenar o vetor. Mas não se engane, se não ordenar os dois vetores paralelamente, você vai levar TLE :(

Caso nunca tenha utilizado a função `qsort(3)` leia o seu manual.

## Exemplos

### Exemplo de entrada

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

### Saída para o exemplo acima

```
Mesmos elementos
```

### Exemplo de Entrada

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

**Saída para o exemplo acima**

Mesmos elementos

**Exemplo de Entrada**

```
5
1 1 3 1 1
1 1 2 1 1
```

**Saída para o exemplo acima**

Diferentes

**Exemplo de Entrada**

```
5
1 7 3 1 9
9 1 2 7 1
```

**Saída para o exemplo acima**

Diferentes

**Exemplo de Entrada**

```
5
2 7 1 1 9
9 1 2 7 1
```

**Saída para o exemplo acima**

Mesmos elementos

*Author: Bruno Ribas*