

O que está garantido de ser impresso pelo código a seguir? (Escolha todas as opções que se aplicam.)

```
int[] array = {6,9,8};
System.out.println("B" + Arrays.binarySearch(array,9));
System.out.println("C" + Arrays.compare(array,
new int[] {6, 9, 8}));
System.out.println("M" + Arrays.mismatch(array,
new int[] {6, 9, 8}));
```

- A. B1
- B. B2
- C. C-1
- D. C0
- E. M-1
- F. M0
- G. O código não compila.

## ANÁLISE DA QUESTÃO

Este é um problema de programação em Java que envolve a utilização de métodos da classe Arrays para manipulação de arrays. Vamos analisar o código linha por linha:

1. `Arrays.binarySearch(array, 9)`: Este método procura o valor 9 no array e retorna o índice onde o valor foi encontrado ou onde ele deveria ser inserido para manter a ordem do array. Entretanto vamos entender melhor este método

O método `Arrays.binarySearch(array, key)` em Java é usado para realizar uma busca binária em um array ordenado:

- **Parâmetros:**
  - `array`: O array no qual você deseja realizar a busca.
  - `key`: O elemento que você está procurando no array.
- **Retorno:**
  - Se o elemento `key` for encontrado no array, o método retorna o índice desse elemento.
  - Se o elemento `key` não for encontrado, o método retorna `-(ponto de inserção) - 1`. O ponto de inserção é definido como o ponto em que a chave seria inserida no array: o índice do primeiro elemento maior que a chave, ou `a.length` se todos os elementos no array forem menores que a chave especificada. Note que isso garante que o valor de retorno será  $\geq 0$  se e somente se a chave for encontrada.
- **Requisito de Ordenação:**
  - Antes de usar `Arrays.binarySearch(array, key)`, é crucial que o array esteja ordenado. Caso contrário, o resultado da busca não será confiável. Se o array não estiver ordenado, o comportamento do método é indefinido.

2. `Arrays.compare(array, new int[] {6, 9, 8})`: Este método compara dois arrays lexicograficamente. Ele retorna um valor negativo se o primeiro array for lexicograficamente menor que o segundo, 0 se forem iguais e um valor positivo se o

primeiro for lexicograficamente maior que o segundo. Neste caso, os arrays são iguais, então o segundo `System.out.println` imprime “C0”.

3. `Arrays.mismatch(array, new int[] {6, 9, 8})`: Este método retorna o índice do primeiro elemento diferente entre os dois arrays, ou -1 se eles forem iguais. Como os arrays são iguais, o terceiro `System.out.println` imprime “M-1”.

Portanto, a resposta correta é a opção D. C0 e E. M-1 pois na busca binária o array NÃO ESTÁ ORDENADO E RETORNO CORRETO NÃO É GARANTIDO.