A busca binária é um algoritmo eficiente para encontrar um valor específico em um conjunto de dados ordenados. Ele funciona dividindo repetidamente o conjunto pela metade e comparando o valor de busca com o valor do meio. Se forem iguais, a busca termina com sucesso. Caso contrário, a busca continua na metade relevante do conjunto, reduzindo a área de busca pela metade a cada iteração. Esse processo é repetido até que o valor desejado seja encontrado ou até que não haja mais elementos para verificar.

Em relação ao código de busca binária, temos:

a) Se a chave de busca estiver ausente no arranjo, a função buscaBinaria() retornará -1. Isso indica que a chave não foi encontrada no arranjo.

b) É necessário que o arranjo esteja ordenado para que a busca binária funcione corretamente. A estratégia da busca binária é baseada na comparação entre a chave de busca e o elemento do meio do arranjo. Se o arranjo não estiver ordenado, não é possível determinar corretamente em qual metade do arranjo a chave de busca pode estar, tornando a busca binária ineficaz.

c) A busca binária pode ser aplicada tanto em arranjos estáticos (definidos em tempo de compilação) quanto em arranjos dinâmicos (definidos em tempo de execução). A eficiência da busca binária está relacionada à sua estratégia de divisão e busca pela metade do arranjo, não dependendo do arranjo ser estático ou dinâmico.

d) Para invocar a função buscaBinaria() no arranjo VET, que possui 10 elementos, e procurar pela chave de busca igual a 15, a chamada seria:

int resultado = buscaBinaria(VET, 0, 9, 15);

Essa linha de código invoca a função buscaBinaria(), passando o arranjo VET, o índice inicial (0), o índice final (9) e a chave de busca (15). O resultado da busca será armazenado na variável "resultado".