

```

1  /* Cau 4: Viết hàm tìm kiếm một phần tử trong mảng bằng tìm kiếm tuyến tính.
2  Input:
3   + a[]: int array
4   + n: int
5   + value: int
6  Output:
7   + return: bool
8  */
9  bool linearSearch1(int a[], int n, int value)
10 {
11     for (int i = 0; i < n; i++)
12     {
13         if (a[i] == value)
14             return true;
15     }
16     return false;
17 };
18
19 /* Cau 5: Viết hàm tìm kiếm một phần tử trong mảng bằng tìm kiếm nhị phân.
20 Input:
21 + a[]: int array
22 + n: int
23 + value: int
24 Output:
25 + return: bool
26 */
27 bool binarySearch1(int a[], int n, int value)
28 {
29     int left = 0;
30     int right = n - 1;
31
32     while (left ≤ right)
33     {
34         int mid = (left + right) / 2;
35         if (a[mid] == value)
36             return true;
37         else if (a[mid] < value)
38             left = mid + 1;
39         else
40             right = mid - 1;
41     }
42     return false;
43 };
44
45 /* Cau 6: Viết hàm tìm kiếm một phần tử trong mảng bằng tìm kiếm nội suy.
46 Input:
47 + a[]: int array
48 + n: int
49 + value: int
50 Output:
51 + return: bool
52 */
53 bool interpolationSearch1(int a[], int n, int value)
54 {
55     int left = 0;
56     int right = n - 1;
57     int mid;
58
59     while (left ≤ right)
60     {
61         //  $L + ((R - L) / (a[R] - a[L])) * (value - a[L])$ 
62         mid = left + ((right - left) / (a[right] - a[left])) * (value - a[left]);
63         if (a[mid] == value)
64             return true;
65         else if (a[mid] < value)
66             left = mid + 1;
67         else
68             right = mid - 1;
69     }
70     return false;
71 };

```