|  |
| --- |
| **Skill 28.1 Exercise 1** |
| If an array of Integer objects contain the following elements,  89 42 -3 13 109 70 2  Write the method search, that can be implemented as illustrated below, |
| public static void main(String[] args) {  int arr[] = {2, 3, 4, 10, 40};  int x = 10;  int result = Sequential.search(arr, x);  if(result == -1)  System.out.print("Element is not present in array");  else  System.out.print("Element is present at index " + result);  }  **//complete the search method below.** |

|  |
| --- |
| **Skill 28.2 Exercise 1** |
| For which of the following arrays could a binary search be applied? Explain. |
| {1, 10, 22, 32, 100, 200, 302}  {x, y, z, a, b, c, d, f}  {and, ant, bat, cat, dog, rat}  {300.12, 200, 100, 50, 2, 0, -80} |

|  |  |
| --- | --- |
| **Skill 28.3 Exercise 1** | |
| Consider the following binarySearch method. The method correctly performs a binary search.  /\*\* Precondition: data is sorted in increasing order. \*/  public static int binarySearch(int[] data, int target) {    int start = 0;  int end = data.length - 1;  while (start <= end) {  int mid = (start + end) / 2; /\* Calculate midpoint \*/    if (target < data[mid]) {  end = mid - 1;  } else if (target > data[mid]) {  start = mid + 1;  } else {  return mid;  }  }  return -1;  } | |
| Consider the following code segment.    int[] values = {1, 2, 3, 4, 5, 8, 8, 8};  int target = 8;    What value is returned by the call binarySearch(values, target) ? | Suppose the binarySearch method is called with an array containing 2,000 elements sorted in increasing order.  What is the maximum number of times that the statement indicated by /\* Calculate midpoint \*/ could execute? |