|  |  |  |  |
| --- | --- | --- | --- |
| **CSE :** Data Structure Lab (Sec-)  Department of Computer Science and Engineering  University of Liberal Arts Bangladesh | | | |
| **Course Title:** Algorithms Lab | | **Course Code:** CSE | |
| **Total Marks:** | | **Time:**  minutes | |
| **Name:** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **ID:** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Consider the following code:   #include <stdio.h>  #define MAX\_SIZE 7  int arr[MAX\_SIZE] = {10, 80, 30, 90, 40, 50, 70};  int partition(int pivot\_index)  {  int pivot = arr[pivot\_index];  int smaller\_index = 0;  // write your solution here  return smaller\_index;  }  int main()  {  int pivot\_index;  scanf("%d" & pivot\_index);  printf("%d\n", partition(pivot\_index));  return 0;  }  Complete the function partition that places the pivot element at its position if the array was sorted. Place all the smaller elements to the left of the pivot and put all the greater elements to the right of the pivot. Return the updated index of the pivot.   |  |  | | --- | --- | | **SAMPLE INPUT** | **SAMPLE OUTPUT** | | 6 | 4 | |  |
| 1. Interview |  |