Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```c
#include <stdio.h>

int findPivot(int[], int, int);
int binarySearch(int[], int, int, int);

/* Searches an element key in a pivoted
   sorted array arrp[] of size n */
int pivotedBinarySearch(int arr[], int n, int key)
{
    int pivot = findPivot(arr, 0, n - 1);

    // If we didn't find a pivot,
// then array is not rotated at all
    if (pivot == -1)
        return binarySearch(arr, 0, n - 1, key);

    // If we found a pivot, then first
// compare with pivot and then
    // search in two subarrays around pivot
    if (arr[pivot] == key)
        return pivot;
    if (arr[0] <= key)
        return binarySearch(arr, 0, pivot - 1, key);
    return binarySearch(arr, pivot + 1, n - 1, key);
}

/* Function to get pivot. For array
   3, 4, 5, 6, 1, 2 it returns 3 (index of 6) */
int findPivot(int arr[], int low, int high)
```

Search the web and Windows

22:43
26-05-2021

Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```c
   3, 4, 5, 6, 1, 2 it returns 3 (index of 6) */
int findPivot(int arr[], int low, int high)
{
    // base cases
    if (high < low)
        return -1;
    if (high == low)
        return low;

    int mid = (low + high) / 2; low + (high - low)/2;
    if (mid < high && arr[mid] > arr[mid + 1])
        return mid;
    if (mid > low && arr[mid] < arr[mid - 1])
        return (mid - 1);
    if (arr[low] >= arr[mid])
        return findPivot(arr, low, mid - 1);
    return findPivot(arr, mid + 1, high);
}

/* Standard Binary Search function*/
int binarySearch(int arr[], int low, int high, int key)
{
    if (high < low)
        return -1;
    int mid = (low + high) / 2; low + (high - low)/2;
    if (key == arr[mid])
        return mid;
    if (key > arr[mid])
        return binarySearch(arr, (mid + 1), high, key);
    return binarySearch(arr, low, (mid - 1), key);
```

Search the web and Windows

22:43
26-05-2021

Run Debug Stop Share Save {} Beautify Language C

main.c

```c
    if (arr[low] >= arr[mid])
        return findPivot(arr, low, mid - 1);
    return findPivot(arr, mid + 1, high);
}

/* Standard Binary Search function*/
int binarySearch(int arr[], int low, int high, int key)
{
    if (high < low)
        return -1;
    int mid = (low + high) / 2; low + (high - low)/2;
    if (key == arr[mid])
        return mid;
    if (key > arr[mid])
        return binarySearch(arr, (mid + 1), high, key);
    return binarySearch(arr, low, (mid - 1), key);
}

/* Driver program to check above functions */
int main()
{
    // Let us search 3 in below array
    int arr1[] = { 5, 6, 7, 8, 9, 10, 1, 2, 3 };
    int n = sizeof(arr1) / sizeof(arr1[0]);
    int key = 3;
    printf("Index of the element is : %d",
           pivotedBinarySearch(arr1, n, key));
    return 0;
}
```

Search the web and Windows

22:44
26-05-2021

onlinegdb.com/online_c_compiler

Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```
50      if (arr[low] >= arr[mid])
51          return findPivot(arr, low, mid - 1);
52      return findPivot(arr, mid + 1, high);
53  }
54
55  /* Standard Binary Search function*/
56  int binarySearch(int arr[], int low, int high, int key)
57  {
58      if (high < low)
59          return -1;
60      int mid = (low + high) / 2; low + (high - low)/2;
61      if (key == arr[mid])
62          return mid;
63      if (key > arr[mid])
64          return binarySearch(arr, (mid + 1), high, key);
```

input

```
Index of the element is : 8

...Program finished with exit code 0
Press ENTER to exit console.
```

Search the web and Windows

22:45
26-05-2021