

```
1  #include <stdio.h>
2
3  void printarr(int *A, int n)
4  {
5      for (int i = 0; i < n; i++)
6      {
7          printf("%d ", A[i]);
8      }
9      printf("\n");
10 }
11
12 int partition(int A[], int low, int high)
13 {
14
15     int pivot = A[low];
16     int i = low + 1;
17     int j = high;
18     int temp;
19
20     do
21     {
22         while (A[i] <= pivot)
23         {
24             i++;
25         }
26         while (A[j] > pivot)
27         {
28             j--;
29         }
30         if (i < j)
31         {
32             temp = A[i];
33             A[i] = A[j];
34             A[j] = temp;
35         }
36     } while (i < j);
37     temp = A[low];
38     A[low] = A[j];
39     A[j] = temp;
40
41     return j;
42 }
43 void quickshort(int A[], int low, int high)
44 {
45     if (low < high)
46     {
47         int pivotindex = partition(A, low, high);
48         quickshort(A, low, pivotindex - 1);
49         quickshort(A, pivotindex + 1, high);
50     }
51 }
52
53 int main()
54 {
55     int A[] = {2, 5, 3, 6, 9, 7, 4};
56     int n = 7;
57     printarr(A, n);
58     quickshort(A, 0, n - 1);
59     printarr(A, n);
60
61     return 0;
62 }
```

2 5 3 6 9 7 4  
2 3 4 5 6 7 9

```
#include <stdio.h>
```

```
void printarr(int *A, int n)
```

```
{  
    for (int i = 0; i < n; i++)  
    {  
        printf("%d ", A[i]);  
    }  
    printf("\n");  
}
```

```
int partition(int A[], int low, int high)
```

```
{  
  
    int pivot = A[low];  
    int i = low + 1;  
    int j = high;  
    int temp;  
  
    do  
    {  
        while (A[i] <= pivot)  
        {  
            i++;  
        }  
        while (A[j] > pivot)  
        {  
            j--;  
        }  
        if (i < j)  
        {  
            temp = A[i];  
            A[i] = A[j];  
            A[j] = temp;  
        }  
    }  
}
```

```
while (i < j);
```

```
    temp = A[low];
```

```
    A[low] = A[j];
```

```
    A[j] = temp;
```

```
    return j;
```

```
}
```

```
void quickshort(int A[], int low, int high)
```

```
{  
    if (low < high)  
    {  
        int pivotindex = partition(A, low, high);  
        quickshort(A, low, pivotindex - 1);  
        quickshort(A, pivotindex + 1, high);  
    }  
}
```

```
int main()
```

```
{  
    int A[] = {2, 5, 3, 6, 9, 7, 4};  
    int n = 7;  
    printarr(A, n);  
    quickshort(A, 0, n - 1);  
    printarr(A, n);  
  
    return 0;  
}
```