```c
#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;
}
```

```
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;
}
```