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

void swap(int *a, int *b) {
     int t = *a;
     *a = *b;
     *b = t;
}

int partition_pivot_last(int array[], int low, int high) {
     int pivot = array[high];
     int i = (low - 1);

     for (int j = low; j < high; j++) {
          if (array[j] < pivot) {
          swap(&array[++i], &array[j]);
          }
     }

     swap(&array[i + 1], &array[high]);
     return (i + 1);
}

int partition_pivot_first(int array[], int low, int high) {
     int pivot = array[low];
     int i = (low + 1);

     for (int j = low + 1; j <= high; j++) {
          if (array[j] < pivot) {
          if (j != i) {
          swap(&array[i], &array[j]);
          }
          i++;
          }
```

```
    }

    swap(&array[i - 1], &array[low]);
    return (i - 1);
}

int partition_pivot_random(int array[], int low, int high) {
    int pivot;
    int n = rand();
    pivot = low + n % (high - low + 1); // Randomizing the pivot
    return partition_pivot_last(array, low, high);
}

int partition_pivot_median(int array[], int low, int high) {

    int pivot;
    int mid = (low + high) / 2;
    if (array[mid] < array[low])
        swap(&array[mid], &array[low]);
    if (array[high] < array[low])
        swap(&array[high], &array[low]);
    if (array[high] < array[mid])
        swap(&array[high], &array[mid]);
    swap(&array[mid], &array[high-1]);

    pivot = array[high-1];

    return partition_pivot_last(array, low, high);
}

void quickSort(int array[], int low, int high) {
    if (low < high) {
        //int pi = partition_pivot_first(array, low, high);
        //int pi = partition_pivot_last(array, low, high);
```

```
            //int pi = partition_pivot_random(array, low, high);
            int pi = partition_pivot_median(array, low, high);

            // Sort the elements on the left of pivot
            quickSort(array, low, pi - 1);

            // Sort the elements on the right of pivot
            quickSort(array, pi + 1, high);
        }
}

int main(void) {
        int array[] = { 1,  3,  2,  4,  6,  8,  7,  9,  5 };
        int size = sizeof(array) / sizeof(array[0]);
        for (int i = 0; i < size; i++)
            printf("|%d", array[i]);
        printf("\n");

        quickSort(array, 0, size - 1);

        printf("\n");
        for (int i = 0; i < size; i++)
            printf("|%d", array[i]);

        getchar();
        return 0;
}
```