1.main.c

```
#include <stdio.h>
#include "header.h"
#include <stdlib.h>
int main() {
   array  arr;
   init(&arr, 5);
   append(&arr, 5);
   append(&arr, 8);
   append(&arr, 9);
   append(&arr, 3);
   append(&arr, 2);
   printf("Original array: ");
   printArray(&arr);
   quickSort(&arr, 0, arr.len - 1);
   printf("Sorted array: ");
   printArray(&arr);
   return 0;
}
```

2.header.h

```
typedef struct {
   int *A;
   int size;
   int len;
} array;
void init(array *arr, int size);
void append(array *arr, int d);
void swap(int *a, int *b);
int partition(array *arr, int low, int high);
void quickSort(array *arr, int low, int high);
void printArray(array *arr);
```

3.logic.c

```
#include <stdio.h>
#include "header.h"
#include <stdlib.h>

void init(array *arr, int size){
   arr -> A = (int *)malloc(sizeof(int) * size);
   arr -> size = size;
   arr -> len = 0;
}

void append(array *arr, int d){
   if(arr -> len < arr -> size){
      arr -> A[arr -> len++] = d;
   }
}

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

int partition(array *arr, int low, int high) {
   int pivot = arr->A[high];
   int i = low - 1;

   for (int j = low; j < high; j++) {
      if (arr->A[j] < pivot) {
         i++;
         swap(&arr->A[i], &arr->A[j]);
```

```
        }
    }
    swap(&arr->A[i + 1], &arr->A[high]);
    return (i + 1);
}

void quickSort(array *arr, int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

void printArray(array *arr) {
    for (int i = 0; i < arr->len; i++) {
        printf("%d ", arr->A[i]);
    }
    printf("\n");
}
```

Output:

```
PS D:\COEP\DSA\Serious\Assignments\Assignment7\QuickSort> gcc -Wall  main.c logic.c
PS D:\COEP\DSA\Serious\Assignments\Assignment7\QuickSort> ./a
Original array: 5 8 9 3 2
Sorted array: 2 3 5 8 9
PS D:\COEP\DSA\Serious\Assignments\Assignment7\QuickSort>
```