

Quicksort

Name: Tanjima Akhanda Mim
ID: 191-15-2455

1.

```
#include <stdio.h>

int partition (int arr[], int left, int right) {
    int pivot = arr[right];
    int i = (left - 1);
    for (int j = left; j <= right - 1; j++){
        if (arr[j] < pivot){
            i++;
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    int temp = arr[i + 1];
    arr[i + 1] = arr[right];
    arr[right] = temp;
    return (i + 1);
}

void quickSort(int arr[], int left, int right){
    if (left < right) {
        int pi = partition(arr, left, right);
        quickSort(arr, left, pi - 1);
        quickSort(arr, pi + 1, right);
    }
}

int main( ){
    int n = 5, t;
    int arr[n];
    scanf("%d", &t);
    while(t--){
        for(int i = 0; i < n; i++){
            scanf("%d", &arr[i]);
        }
    }
}
```

```
        quickSort(arr, 0, n - 1);  
        printf("Sorted data: ");  
        for(int i = 0; i < n; i++){  
            printf("%d ", arr[i]);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

2.

```
#include <stdio.h>

int partition (int arr[], int left, int right) {
    int pivot = arr[right];
    int i = (left - 1);
    for (int j = left; j <= right - 1; j++){
        if (arr[j] < pivot){
            i++;
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    int temp = arr[i + 1];
    arr[i + 1] = arr[right];
    arr[right] = temp;
    return (i + 1);
}

void quickSort(int arr[], int left, int right){
    if (left < right) {
        int pi = partition(arr, left, right);
        quickSort(arr, left, pi - 1);
        quickSort(arr, pi + 1, right);
    }
}

int main( ){
    int n = 6, pick;
    int arr[n];
    for(int i = 0; i < 6; i++){
        scanf("%d", &arr[i]);
    }
    printf("Pick Value:");
    scanf("%d", &pick);
```

```
quickSort(arr, 0, n - 1);
printf("Ascending: ");
int flag = 0;
for(int i = 0; i < n; i++){
    printf("%d ", arr[i]);
    if(arr[i] == pick)
        flag = i;
}
printf("\n");
printf("Large Value: %d\n", n - (flag + 1));
printf("Small Value: %d\n", flag - 1);
return 0;
}
```