18.QUICK SORT

SAMPLE CODE

```
#include <stdio.h>
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
int partition(int a[], int low, int high) {
    int pivot = a[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (a[j] < pivot) {
            i++;
            swap(&a[i], &a[j]);
        }
    }
    swap(&a[i + 1], &a[high]);
    return i + 1;
}
void quickSort(int a[], int low, int high) {
    if (low < high) {
        int pi = partition(a, low, high);
        quickSort(a, low, pi - 1);
        quickSort(a, pi + 1, high);
    }
}

int main() {
```

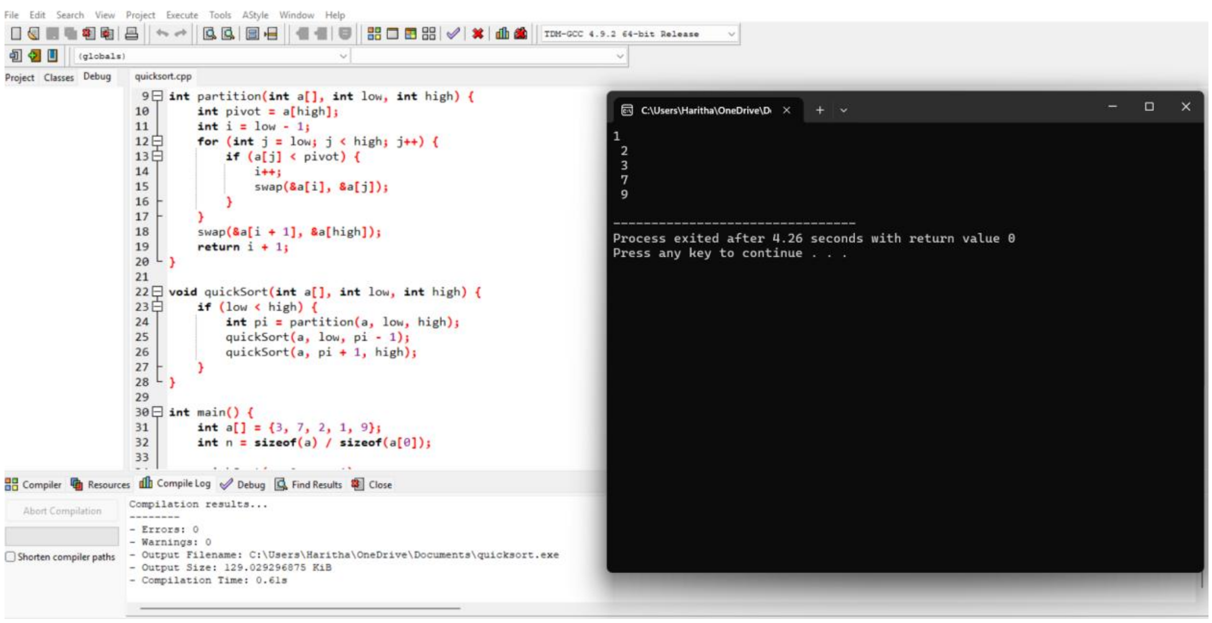
```c
    int a[] = {3, 7, 2, 1, 9};
    int n = sizeof(a) / sizeof(a[0]);

    quickSort(a, 0, n - 1);

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

    return 0;
}
```

OUTPUT


File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
quicksort.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
C:\Users\Haritha\OneDrive\D

```c
int partition(int a[], int low, int high) {
    int pivot = a[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (a[j] < pivot) {
            i++;
            swap(&a[i], &a[j]);
        }
    }
    swap(&a[i + 1], &a[high]);
    return i + 1;
}

void quickSort(int a[], int low, int high) {
    if (low < high) {
        int pi = partition(a, low, high);
        quickSort(a, low, pi - 1);
        quickSort(a, pi + 1, high);
    }
}

int main() {
    int a[] = {3, 7, 2, 1, 9};
    int n = sizeof(a) / sizeof(a[0]);
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Haritha\OneDrive\Documents\quicksort.exe
- Output Size: 129.029296875 KiB
- Compilation Time: 0.61s
```

```
1
 2
 3
 7
 9

--------------------------------
Process exited after 4.26 seconds with return value 0
Press any key to continue . . .
```