QUICK SORT

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

#define MAX 5000

void quicksort(int[], int, int);
int partition(int[], int, int);

int main() {
   int i, n, a[MAX], ch;
   clock_t start, end;

   srand(time(NULL));

   while (1) {
      printf("\nEnter the number of elements: ");
      scanf("%d", &n);

      for (i = 0; i < n; i++) {
         a[i] = rand() % 200;
      }

      printf("The randomly generated array is:\n");
      for (i = 0; i < n; i++) {
         printf(" %d", a[i]);
      }
      printf("\n");

      start = clock();

      quicksort(a, 0, n - 1);

      end = clock();

      printf("\nThe sorted array is:\n");
      for (i = 0; i < n; i++) {
         printf("%d\n", a[i]);
      }

      printf("\nTime taken = %f seconds\n", (double)(end - start) / CLOCKS_PER_SEC);

      printf("\nDo you wish to continue (0/1)? ");
      scanf("%d", &ch);
      if (ch == 0) {
```

```
            break;
        }
    }

    return 0;
}

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

int partition(int a[], int low, int high) {
    int key = a[low];
    int i = low + 1;
    int j = high;
    int temp;

    while (i <= j) {
        while (i <= high && a[i] <= key) {
            i++;
        }
        while (a[j] > key) {
            j--;
        }
        if (i < j) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        } else {
            temp = a[j];
            a[j] = a[low];
            a[low] = temp;
        }
    }
    return j;
}
```

OUTPUT:

```
Enter the number of elements: 5
The randomly generated array is:
 43 56 116 180 175

The sorted array is:
43
56
116
175
180

Time taken = 0.000000 seconds
```