QUICKSORT

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

#define MAX 5000

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

int main() {
   int i, n, a[MAX], ch = 1;
   clock_t start, end;

   while (ch) {
      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]);

      start = clock();
      quicksort(a, 0, n - 1);
      end = clock();
```

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

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

        printf("\nDo you wish to continue? (1 = Yes / 0 = No): ");
        scanf("%d", &ch);
    }

    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 pivot = a[low];
    int i = low + 1;
    int j = high;
    int temp;

    while (i <= j) {
        while (i <= high && a[i] <= pivot)
            i++;
        while (a[j] > pivot)
```

```
        j--;
      if (i < j) {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
      }
    }

    a[low] = a[j];
    a[j] = pivot;

    return j;
}
```

Output

```
Enter the number of elements: 2
The randomly generated array is:
183 86

The sorted array is:
86 183

Time taken = 0.000002 seconds

Do you wish to continue? (1 = Yes / 0 = No):
```