1. Quick Sort

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

void Exch(int *p, int *q) {
   int temp = *p;
   *p = *q;
   *q = temp;
}

void QuickSort(int a[], int low, int high) {
   int i, j, key, k;
   if (low >= high)
      return;
   key = low;
   i = low + 1;
   j = high;
   while (i <= j) {
      while (a[i] <= a[key])
         i = i + 1;
      while (a[j] > a[key])
         j = j - 1;
      if (i < j)
         Exch(&a[i], &a[j]);
   }
   Exch(&a[j], &a[key]);
   QuickSort(a, low, j - 1);
   QuickSort(a, j + 1, high);
```

```
}

int main() {
    int n, a[1000], k;
    clock_t st, et;
    double ts;
    // clrscr(); // Commented out since it's not a standard function

    printf("\nEnter how many numbers: ");
    scanf("%d", &n);

    printf("\nThe random numbers are:\n");
    for (k = 1; k <= n; k++) {
        a[k] = rand();
        printf("%d\t", a[k]);
    }

    st = clock();
    QuickSort(a, 1, n);
    et = clock();

    ts = (double)(et - st) / CLOCKS_PER_SEC;

    printf("\nSorted numbers are:\n");
    for (k = 1; k <= n; k++)
        printf("%d\t", a[k]);

    printf("\nThe time taken is %e seconds", ts);
```

```
    return 0;
}
```

2. Merge Sort

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

int b[50000];

void Merge(int a[], int low, int mid, int high) {
    int i, j, k;
    i = low;
    j = mid + 1;
    k = low;

    while (i <= mid && j <= high) {
        if (a[i] <= a[j])
            b[k++] = a[i++];
        else
            b[k++] = a[j++];
    }

    while (i <= mid)
        b[k++] = a[i++];
    while (j <= high)
        b[k++] = a[j++];

    for (k = low; k <= high; k++)
        a[k] = b[k];
}

void MergeSort(int a[], int low, int high) {
    int mid;

    if (low >= high)
        return;

    mid = (low + high) / 2;
    MergeSort(a, low, mid);
    MergeSort(a, mid + 1, high);
    Merge(a, low, mid, high);
}
```

```
int main() {
   int n, a[50000], k;
   clock_t st, et;
   double ts;

   printf("\nEnter how many numbers: ");
   scanf("%d", &n);

   printf("\nThe random numbers are:\n");
   for (k = 1; k <= n; k++) {
      a[k] = rand();
      printf("%d\t", a[k]);
   }

   st = clock();
   MergeSort(a, 1, n);
   et = clock();
   ts = (double)(et - st) / CLOCKS_PER_SEC;

   printf("\nSorted numbers are:\n");
   for (k = 1; k <= n; k++)
      printf("%d\t", a[k]);

   printf("\nThe time taken is %e seconds\n", ts);

   return 0;
}
```