

## LAB 8 QUICKSORT

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

```
void swap(int *a, int *b)
```

```
{
    int t = *a;
    *a = *b;
```

```
    *b = t;
}
```

```
int partition(int arr[], int low, int high)
```

```
{
    int pivot = arr[high];
```

```
    int i = (low - 1);
```

```
    for (int j = low; j <= high; j++)
```

```
    {
        if (arr[j] < pivot)
```

```
        i++;
```

```
        swap(&arr[j], &arr[i]);
```

```
    }
```

```
    swap(&arr[i+1], &arr[high]);
```

```
    return (i+1);
}
```



```
void quicksort (int arr[], int low, int high)
{ if (low < high).
```

```
    int pi = partition (arr, low, high);
    quicksort (arr, low, pi - 1); }
```

```
void printarray (int arr[], int size)
```

```
{ int i;
```

```
  for (i = 0; i < size; i++)
```

```
    printf ("%d", arr[i]);
```

```
  printf ("\n"); }
```

```
int main()
```

```
{
```

```
  int arr[15000], size, i, j, ch, temp;
```

```
  clock_t start, end;
```

```
  while (1)
```

```
  { printf ("\n 1. for manual entry");
```



printf ("In 2. to display time taken for N elements ranging from 500 to 14000");

printf ("3. exit");

printf ("Enter ur choice:");

scanf ("%d", &ch);

switch (ch).

case 1: printf ("enter size:");  
scanf ("%d", &size);

for (i = 0; i < size; i++)

scanf ("%d", &arr[i]);

start = clock();

quicksort (arr, 0, size-1);

end = clock();

printf ("Sorted Array:");

printArray (arr, size);

case 3: exit(0);  
break;

}}}