Merge sort:

3075/24

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
void split(int[], int, int);
void combine(int[], int, int, int);
void main()
{
 int a[15000], n, i, j, ch, temp;
 clock_t start, end;
 while(1)
 {
  printf("\n1: For manual entry of N value & array elements");
  printf("\n2: To display time taken for sorting number of elements N in the range 500 to 14500");
  printf("\n3: To exit");
    printf("\n Enter your choice");
    scanf("%d", &ch);
    switch(ch)
    {
      case 1: printf("\n Enter the number of elements:");
              scanf("%d", &n);
              printf("\n Enter array elements:");
              for(i=0; i<n; i++)
              {
                scanf("%d", &a[i]);
              }
               start=clock();
               split(a, 0, n-1);
               end=clock();
               printf("\n sorted array is:");
               for(i=0; i<n; i++)
               printf("%d \t", a[i]);
    printf("\n Timetaken to sort %d numbers %f secs", n,
          ((double)(end-start))/CLOCKS_PER_SEC));
           break;

    Case 2:
          n=500;
          while(n<=14500){
          for(i=0; i<n; i++)
            { //a[i]=random(1000);
              a[i]=n-i;
            }
          start=clock();
          split(a, 0, n-1);
        // Dummy loop to create delay
          for(j=0; j<5000000; j++) {temp = 38/600;}
            end=clock();
        printf("\n time taken to sort %d numbers is %f secs", n,
    ((double)(end-start))/CLOCKS_PER_SEC));
          n=n+1000;
```

```
        break;
    case 3: exit(0);
    }
    getchar();
    }
}
void split(int a[], int low, int high)
{
    int mid;
    if(low< high)
    {
    mid=(low + high)/2;
    split(a, low, mid);
    split(a, mid+1, high);
    combine(a, low, mid, high);
    }
}
void combine(int a[], int low, int mid, int high)
{
    int c[15000], i, j, k;
    i=k=low;
    j=mid+1;
    while(i<=mid && j<=high)
    {
        if(a[i]<a[j])
        {
            c[k]=a[i];
            ++k;
            ++i;
        }
        else
        {
            c[k]=a[j];
            ++k;
            ++j;
        }
    }
    if(i>mid)
    {
        while(j<=high)
        {
            c[k]=a[j];
            ++k;
            ++j;
        }
    }
    if(j>high)
    {
        while(i<=mid)
        {
            c[k]=a[i];
            ++k;
            ++i;
        }
    }
```

```
for(i=low ; i<=high; i++)
{
  a[i] = c[i];
}
}
```

O/P:

Enter the number of elements: 4
Enter array elements: 44 33 22 11
Sorted array is: 11 22 33 44
Time taken to sort 4 numbers is 0.000000 secs

1. For manual entry of N value & array elements
2. To display time taken for sorting number of elements N in the range 500 to 14500
3. To exit.

Enter your choice 2

Time taken to sort 500 numbers is 0.000000 Secs

1500	0.015
2500	0
3500	0.016
4500	0
5500	0.016
6500	0
7500	0.15
8500	0
9500	0.016
10500	0
11500	0.015
12500	0
13500	0.16
14500	0

30/5