

merge sort

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
void split (int l, int r);
void combine (int l, int m, int r);
void main ()
{
```

```
    int a[15000], n, i, j, ch, temp;
    clock_t start, end;
    while (1)
{
```

fixing ("In 1. For manual entry of N value
and array elements").

fixing ("In 2. To display time taken for
sorting number of elements N in the range
600 to 14,500").

fixing ("In 3. To crit").

fixing ("In Enter your choice").

scanf ("%d", &n),

getchar (ch)

case 1: fixing ("Enter the no of elements").

scanf ("%d", &n);

fixing ("In Enter array elements").

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

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

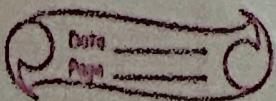
start = clock();

split (0, 0, n - 1);

end = clock();

fixing ("In sorted array").

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



freunif C" • (char", a[i]);
freunif C" In Time taken to sort n numbers is .1.
secs, n ((double)(end-start)) /CLOCKS_PER_SEC);

break;

case a: n = 300;

while (n <= 14500) {

for (i=0, i<n, i++)

a[i] = random(1000);

a[i] = n-i;

}

start = clock();

split(a, 0, n-1);

for (j=0; j < 500000, j++) { tmp = 88/500; }

end = clock();

freunif C" (n time taken to sort n numbers
is .1. secs, n ((double)(end-start)) /
CLOCKS_PER_SEC);

n = n + 100;

} break;

case a: 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);

,

void combine (int arr[], int low, int mid, int high)

{
 int c[15000], i, j, k; (arr, mid, high)
 i = low = k = 0;

j = mid + 1;

while (i <= mid && j <= high)

{

if (arr[i] < arr[j])

c[k] = arr[i]; (i = 0 = l) o
 ++k;

++i;
 }

else

{

c[k] = arr[j]; (j = 0 = r) o

++k;

++j;
 }

if (i > mid)

{

while (j <= high)

{

c[k] = arr[j]; (j = 0 = r) o
 ++k;

++j;
 }

{

if (j > high)

{

while (i <= mid)

{

```

3. {
    c[i]=a[i];
    for(i=low; i<high; i++)
    {
        a[i]=c[i];
    }
}

```

Output:

1. For manual entry of N value and 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 : 1

Enter the no of elements : 5

Enter array elements : 44 33 22 11

Sorted array : 11 22 33 44

Time taken to sort 5 numbers is 0.00000 secs

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

Enter your choice : 2

Time taken to sort 500 numbers is 0.047000 secs

Time taken to sort 1000 numbers is 0.031000 secs

Time taken to sort 1500 numbers is 0.047000 secs

Time taken to sort 2000 numbers is 0.031000 secs

Time taken to sort 2500 numbers is 0.047000 secs

Time taken to sort 5500 numbers is 0.04700 sec
Time taken to sort 6500 numbers is 0.03100 sec
Time taken to sort 7500 numbers is 0.04700 sec
Time taken to sort 8500 numbers is 0.03100 sec
Time taken to sort 9500 numbers is 0.03200 sec
Time taken to sort 10500 numbers is 0.04700 sec
Time taken to sort 11500 numbers is 0.03100 sec
Time taken to sort 12500 number is 0.04700 sec
Time taken to sort 13500 numbers is 0.03100 sec
Time taken to sort 14500 number is 0.03100 sec

Selection sort and Merge sort

