**Name:** Purvi Singhvi

**Enrollement No.:** 180099

**PROGRAM-6**

**Aim:** Write an algorithm and program to sort n numbers using Merge sort technique.

**Algorithm:**

MERGE(A, p, q, r)

1. n1= q-p+1

2. n2= r-q

3. Let L[1….. n1+1] and R[1…...n2+1] are new arrays

4. for (i=1 to n1) do

5. L[i] = A[p+i-1]

6. END for

7. for(j=1 to n2)do

8. R[j]=A[q+j]

9. END for

10. L[n1+1]= infinity

11. R[n2+1]=inifinity

12. for(k=p to r)do

13. if(L[i] <= R[j])

14. A[k] = L[i]

15. i=i+1

16. Else

17. A[k]=R[j]

18. j=j+1

19. End if

MERGE SORT(A,p,r)

1. if(p<r)

2. q= floor((p+q)/2)

3. MERGE SORT(A, p, q)

4. MERGE SORT(A, q+1, r)

5. MERGE(A, p, q, r)

6. End if

7. STOP

**ii)** Without using recursion

**Source Code:**

#include <stdio.h>

#include<conio.h>

#define MAX 30

void main()

{

int arr[MAX],temp[MAX],i,j,k,n,size,l1,h1,l2,h2;

clrscr();

printf("Enter the number of elements : ");

scanf("%d",&n);

printf("\n Enter elements:");

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

{

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

}

for(size=1; size < n; size=size\*2 )

{

l1=0;

k=0;

while( l1+size < n)

{

h1=l1+size-1;

l2=h1+1;

h2=l2+size-1;

if( h2>=n )

h2=n-1;

i=l1;

j=l2;

while(i<=h1 && j<=h2 )

{

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

temp[k++]=arr[i++];

else

temp[k++]=arr[j++];

}

while(i<=h1)

temp[k++]=arr[i++];

while(j<=h2)

temp[k++]=arr[j++];

l1=h2+1;

}

for(i=l1; k<n; i++)

temp[k++]=arr[i];

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

arr[i]=temp[i];

}

printf("Sorted list is :\n");

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

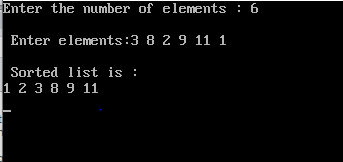
printf("%d ", arr[i]);

printf("\n");

getch();

}

**Output:**



**Complexity:**

Best case: nlog(n)

Worst case: nlog(n)

Average case: nlog(n)