MERGE SORT

Exp. No.:

AIM:

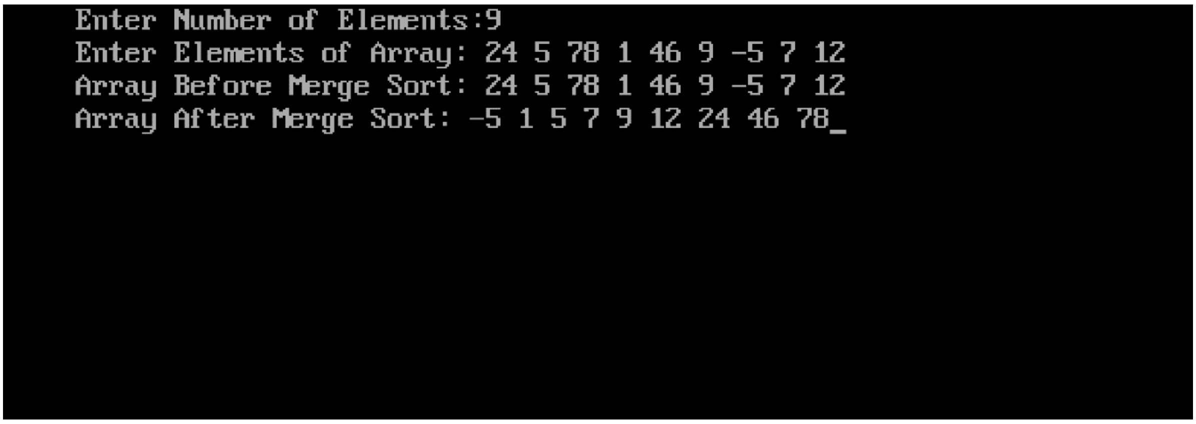
ALGORITHM:

PROGRAM:

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

int i;

void mergePart(int *a, int FI, int mid, int LI);

void mergeSort(int *a, int FI, int LI);

int main()
{
   int n, *a;
   printf("Enter Number of Elements:");
   scanf("%d", &n);
   a = (int *)malloc(n * sizeof(int));
   printf("Enter Elements of Array: ");
   for (i = 0; i < n; i++)
   {
      scanf("%d", &a[i]);
   }
   printf("Array Before Merge Sort:");
   for (i = 0; i < n; i++)
   {
      printf(" %d", a[i]);
   }
   mergeSort(a, 0, n - 1);
   printf("\nArray After Merge Sort:");
   for (i = 0; i < n; i++)
   {
      printf(" %d", a[i]);
   }
   getch();
   clrscr();
   return 0;
}
```

```
void mergePart(int *a, int FI, int mid, int LI)
{
    int j = mid + 1;
    int k = FI - 1;
    int *b;
    i = FI;
    b = (int *)malloc((LI + 1) * sizeof(int));
    while (i <= mid && j <= LI)
    {
        if (a[i] > a[j])
        {
            k++;
            b[k] = a[j];
            j++;
        }
        else
        {
            k++;
            b[k] = a[i];
            i++;
        }
    }
    if (i > mid)
    {
        while (j <= LI)
        {
            k++;
            b[k] = a[j];
            j++;
        }
    }
    else
    {
        while (i <= mid)
        {
            k++;
            b[k] = a[i];
            i++;
        }
    }
    for (i = FI; i <= LI; i++)
    {
        a[i] = b[i];
    }
}

void mergeSort(int *a, int FI, int LI)
{
    int mid;
    if (FI < LI)
    {
        mid = (LI + FI) / 2;
        mergeSort(a, FI, mid);
        mergeSort(a, mid + 1, LI);
        mergePart(a, FI, mid, LI);
    }
}
```

OUTPUT:



```
Enter Number of Elements:9
Enter Elements of Array: 24 5 78 1 46 9 -5 7 12
Array Before Merge Sort: 24 5 78 1 46 9 -5 7 12
Array After Merge Sort: -5 1 5 7 9 12 24 46 78_
```

RESULT: