Merge Sort.

Problem1:

```
#include<stdlib.h>

void merge(int arr[], int l, int m, int r)

{

 int i, j, k;

 int n1 = m - l + 1;

 int n2 = r - m;

 int L[n1], R[n2];

 for (i = 0; i < n1; i++)

 {

 L[i] = arr[l + i];

 }

 for (j = 0; j < n2; j++)

 {

 R[j] = arr[m + 1+ j];

 }

 i = 0;

 j = 0;

 k = l;

 while (i < n1 && j < n2)

 {

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

 {

 arr[k] = L[i];

 i++;
 }
 else
 {
 arr[k] = R[j];
 j++;
 }
 k++;
 }
 while (i < n1)
 {
 arr[k] = L[i];
 i++;
 k++;
 }
 while (j < n2)
 {
 arr[k] = R[j];
 j++;
 k++;
 }
}
void mergeSort(int arr[], int l, int r)
{
 if (l < r)
 {
 int m = l+(r-l)/2;
 mergeSort(arr, l, m);
 mergeSort(arr, m+1, r);

 merge(arr, l, m, r);

 }

}

void printArray(int A[], int size)

{

 int i;

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

 {

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

 }

 printf("\n");

 int mid= size/2;

 printf("\nMedian: %d",A[mid]);

}

int main()

{

 int arr[1000], arr_size;

 printf("Enter Number of Elements in Array : ");

 scanf("%d", &arr_size);

 printf("Enter Elements of Array : \n");

 for(int i = 0; i < arr_size; i++)

 {

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

 }

 mergeSort(arr, 0, arr_size - 1);

 printf("\nSorted Array is : ");
 printArray(arr, arr_size);

 return 0;
}
```

Practice2:

```
#include<stdlib.h>
void merge(int arr[], int l, int m, int r)
{
 int i, j, k;
 int n1 = m - l + 1;
 int n2 = r - m;
 int L[n1], R[n2];
 for (i = 0; i < n1; i++)
 {
 L[i] = arr[l + i];
 }
 for (j = 0; j < n2; j++)
 {
 R[j] = arr[m + 1+ j];
 }
 i = 0;
 j = 0;
 k = l;
 while (i < n1 && j < n2)
 {
 if (L[i] <= R[j])
 {
```

```
 arr[k] = L[i];
 i++;
 }
 else
 {
 arr[k] = R[j];
 j++;
 }
 k++;
 }
 while (i < n1)
 {
 arr[k] = L[i];
 i++;
 k++;
 }
 while (j < n2)
 {
 arr[k] = R[j];
 j++;
 k++;
 }
}
void mergeSort(int arr[], int l, int r)
{
 if (l < r)
 {
```

```
 int m = l+(r-l)/2;

 mergeSort(arr, l, m);

 mergeSort(arr, m+1, r);

 merge(arr, l, m, r);

 }

}

void printArray(int A[], int size)

{

 int i;

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

 {

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

 }

 printf("\n");

}

void printsum(int A[],int size){

    int i,sum=0,digit_sum=0;

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

 {

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

  sum=sum+A[i];

 }

 printf("\nSum: %d",sum);


 while(sum){

    digit_sum+=(sum%10);

    sum=sum/10;
 }
 printf("\nDigit_Sum=%d",digit_sum);


}
int main()
{
 int arr[1000], arr_size;
 printf("Enter Number of Elements in Array : ");
 scanf("%d", &arr_size);
 printf("Enter Elements of Array : \n");
 for(int i = 0; i < arr_size; i++)
 {
 scanf("%d", &arr[i]);
 }
 mergeSort(arr, 0, arr_size - 1);
 printf("\nSorted Array is : ");
 printArray(arr, arr_size);
 printsum(arr,arr_size);


 return 0;
}
```

Selection Sort

Practice Problem2:

```
#include <stdio.h>
int main()
{
```

```
int a[100], n, i, j,t, position, swap, min_idx,temp;
printf("Enter Number of Elements n :");
scanf("%d", &n);
printf("\nTime:");
scanf("%d",&t);
printf("Enter %d Numbers n: \n", n);
for (i = 0; i < n; i++)
{
scanf("%d", &a[i]);
}
for (i = 0; i < t-1; i++)
{
min_idx = i;
for (j = i+1; j < t; j++)
{
if (a[j] < a[min_idx])
{
min_idx = j;
}
}
temp = a[min_idx];
a[min_idx] = a[i];
a[i] = temp;
}
printf("\nSorted Array:\n");
for(i = 0; i < n; i++)
{
 printf("%d ", a[i]);
 }
 return 0;
}
```