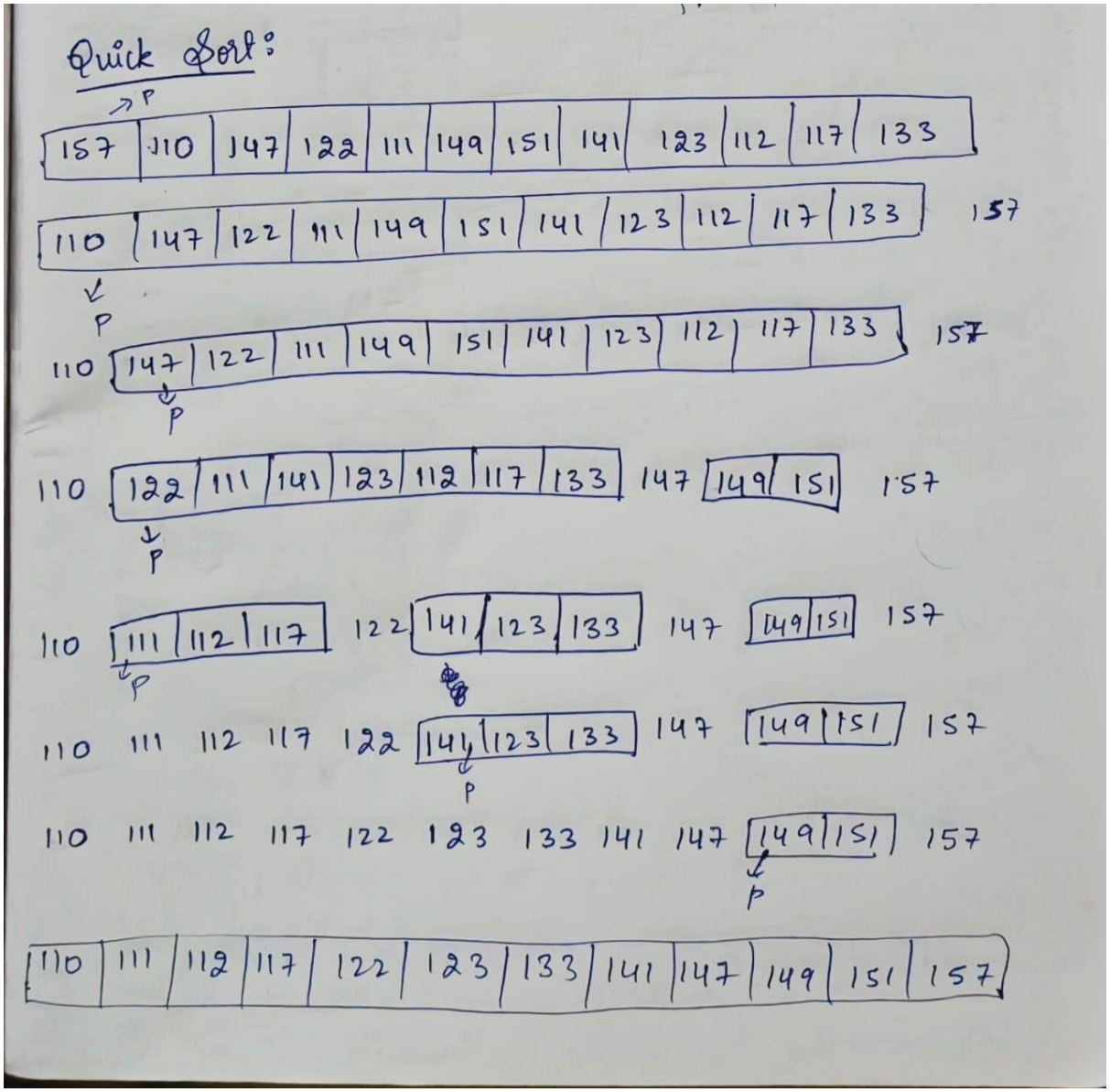
**NAME :** Mahalakshmi K

**ROLL NO :** CH.SC.U4CSE24160

## WEEK – 4

**Quick Sort :**




Logic :

```
void quickSort(int a[], int low, int high){
   if (low < high)    {
      int p = partition(a, low, high);
      quickSort(a, low, p - 1);
      quickSort(a, p + 1, high);
   }
}
int partition(int a[], int low, int high){
   int pivot = a[low];
   int i = low + 1;
   int j = high;
   int temp;
   while (i <= j)    {
      while (i <= high && a[i] <= pivot)
         i++;
      while (a[j] > pivot)
         j--;
      if (i < j) {
         temp = a[i];
         a[i] = a[j];
         a[j] = temp;
      }
   }
   temp = a[low];
   a[low] = a[j];
   a[j] = temp;
   return j;
}
```

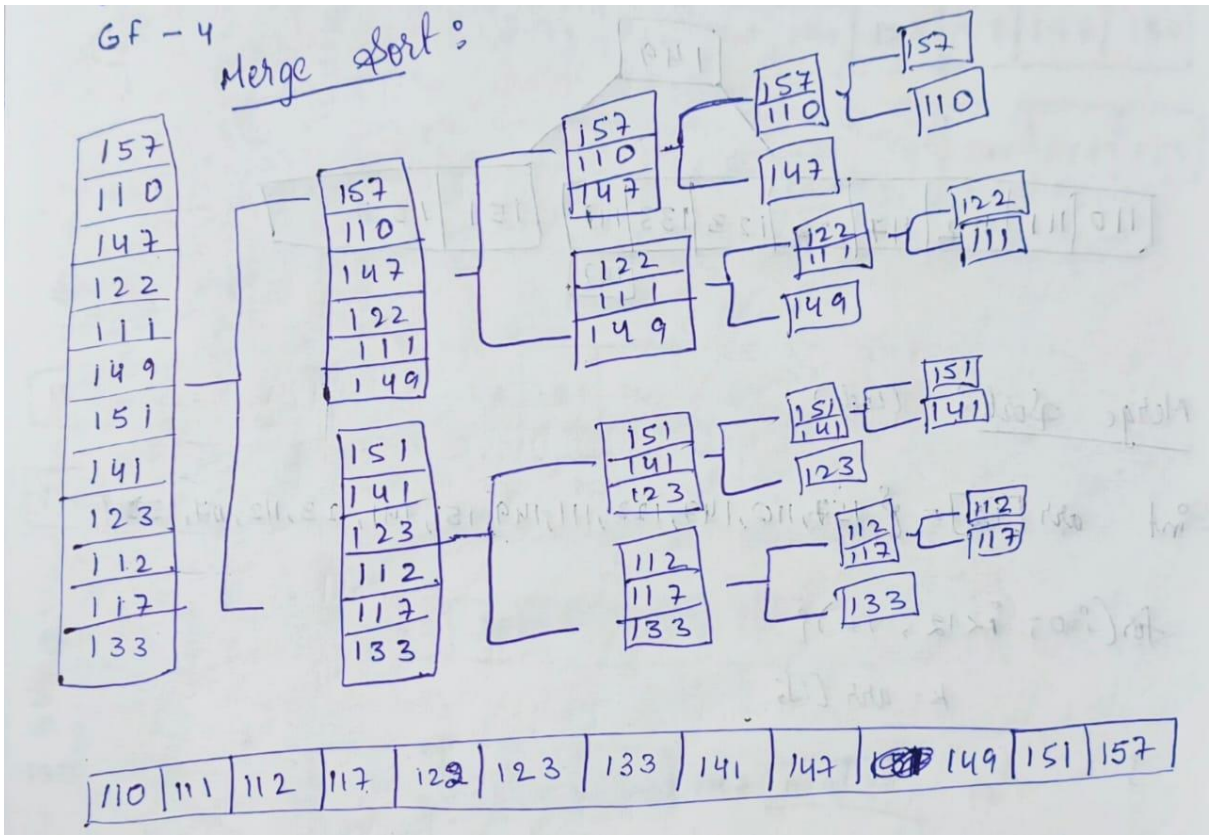
Code :

```c
#include <stdio.h>
int partition(int a[], int low, int high)
{
    int pivot = a[low];
    int i = low + 1;
    int j = high;
    int temp;
    while (i <= j)
    {
        while (i <= high && a[i] <= pivot)
            i++;
        while (a[j] > pivot)
            j--;
        if (i < j)
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    temp = a[low];
    a[low] = a[j];
    a[j] = temp;

    return j;
}
void quickSort(int a[], int low, int high)
{
    if (low < high)
    {
        int p = partition(a, low, high);
        quickSort(a, low, p - 1);
        quickSort(a, p + 1, high);
    }
}
int main()
{
    int a[] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int n = sizeof(a) / sizeof(a[0]);
    int i;
    quickSort(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
        printf("%d ", a[i]);
    return 0;
}

```

Output :

```
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157
--------------------------------
Process exited after 0.1601 seconds with return value 0
Press any key to continue . . .
```

## Merge Sort :




Logic :

```
void merge(int a[], int low, int mid, int high){
    int i = low;
    int j = mid + 1;
    int k = low;
    int temp[100];
```

```
    while (i <= mid && j <= high)    {
        if (a[i] <= a[j])
            temp[k++] = a[i++];
        else
            temp[k++] = a[j++];
    }
    while (i <= mid)
        temp[k++] = a[i++];
    while (j <= high)
        temp[k++] = a[j++];
    for (i = low; i <= high; i++)
        a[i] = temp[i];
}

void mergeSort(int a[], int low, int high){
    if (low < high)
    {
        int mid = (low + high) / 2;
        mergeSort(a, low, mid);
        mergeSort(a, mid + 1, high);
        merge(a, low, mid, high);
    }
}
```

Code :

```c
#include <stdio.h>
void merge(int a[], int low, int mid, int high)
{
    int i = low;
    int j = mid + 1;
    int k = low;
    int temp[100];
    while (i <= mid && j <= high)
    {
        if (a[i] <= a[j])
            temp[k++] = a[i++];
        else
            temp[k++] = a[j++];
    }
    while (i <= mid)
        temp[k++] = a[i++];

    while (j <= high)
        temp[k++] = a[j++];

    for (i = low; i <= high; i++)
        a[i] = temp[i];
}
void mergeSort(int a[], int low, int high)
{
    if (low < high)
    {
        int mid = (low + high) / 2;
        mergeSort(a, low, mid);
        mergeSort(a, mid + 1, high);
        merge(a, low, mid, high);
    }
}
int main()
{
    int a[] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int n = sizeof(a) / sizeof(a[0]);
    int i;
    mergeSort(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Output :

```
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157
--------------------------------
Process exited after 0.08793 seconds with return value 0
Press any key to continue . . .
```