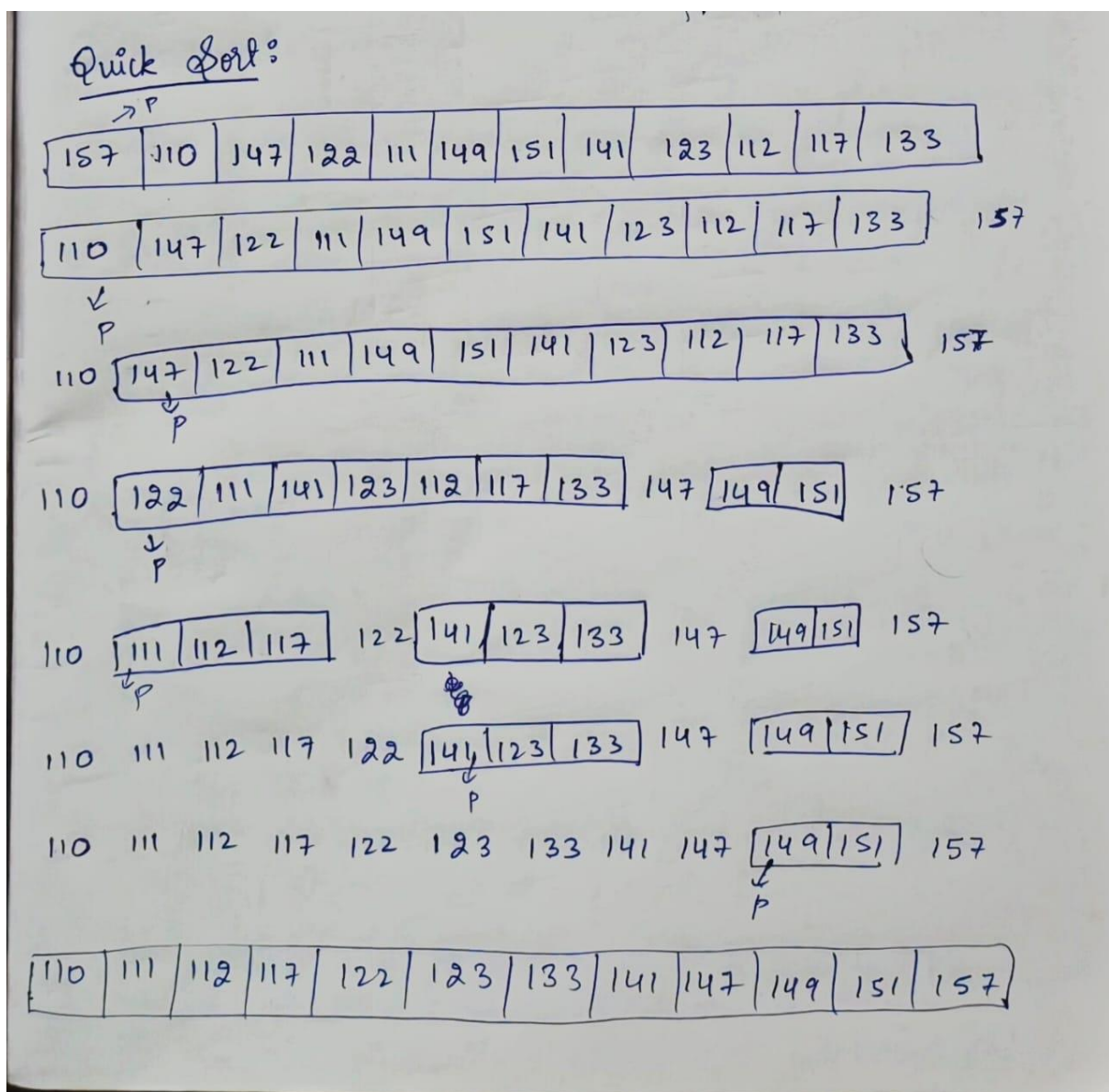


NAME : Mahalakshmi K

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:

```
1  #include <stdio.h>
2  int partition(int a[], int low, int high)
3  {
4      int pivot = a[low];
5      int i = low + 1;
6      int j = high;
7      int temp;
8      while (i <= j)
9      {
10         while (i <= high && a[i] <= pivot)
11             i++;
12         while (a[j] > pivot)
13             j--;
14         if (i < j)
15         {
16             temp = a[i];
17             a[i] = a[j];
18             a[j] = temp;
19         }
20     }
21     temp = a[low];
22     a[low] = a[j];
23     a[j] = temp;
24     return j;
25 }
26 void quickSort(int a[], int low, int high)
27 {
28     if (low < high)
29     {
30         int p = partition(a, low, high);
31         quickSort(a, low, p - 1);
32         quickSort(a, p + 1, high);
33     }
34 }
35 int main()
36 {
37     int a[] = {157,110,147,122,111,149,151,141,123,112,117,133};
38     int n = sizeof(a) / sizeof(a[0]);
39     int i;
40     quickSort(a, 0, n - 1);
41     printf("Sorted array:\n");
42     for (i = 0; i < n; i++)
43         printf("%d ", a[i]);
44     printf("%d ", a[i]);
45     return 0;
46 }
47
```

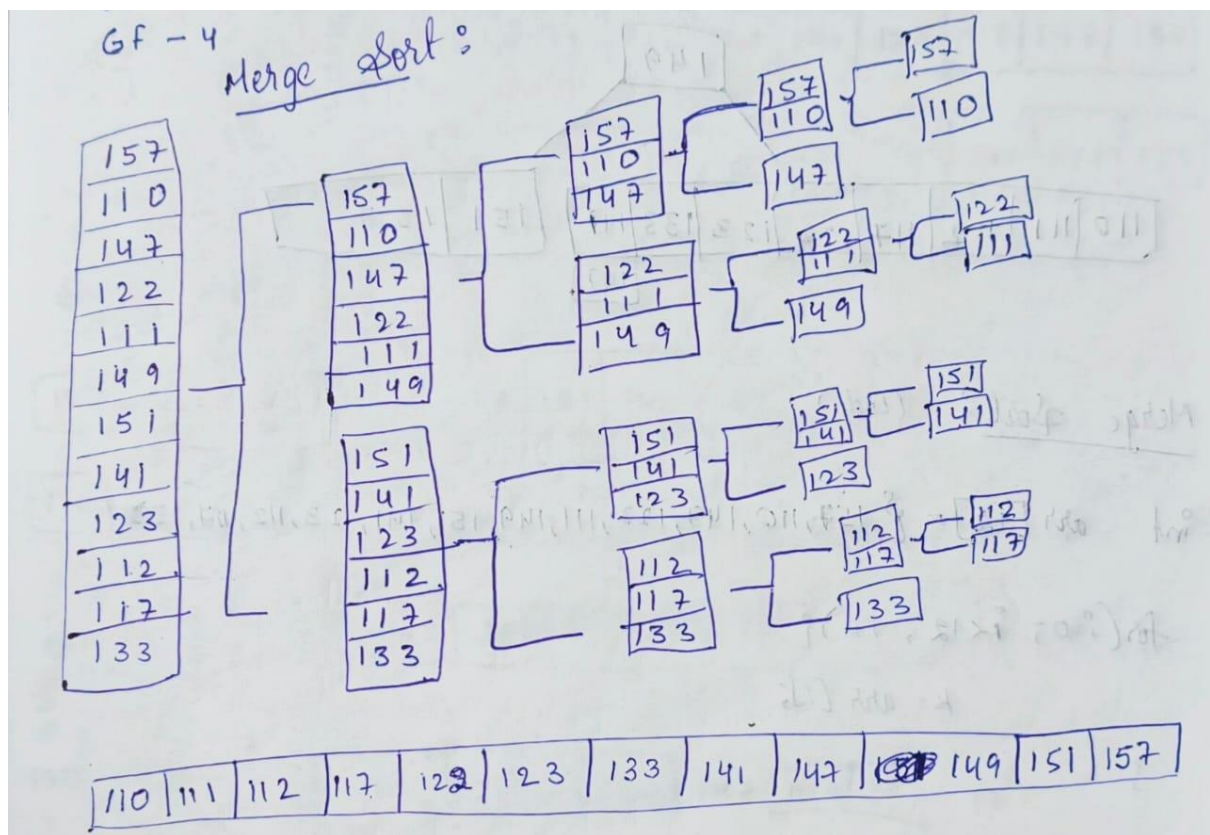
Output :

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 :

```

1  #include <stdio.h>
2  void merge(int a[], int low, int mid, int high)
3  {
4      int i = low;
5      int j = mid + 1;
6      int k = low;
7      int temp[100];
8      while (i <= mid && j <= high)
9      {
10         if (a[i] <= a[j])
11             temp[k++] = a[i++];
12         else
13             temp[k++] = a[j++];
14     }
15     while (i <= mid)
16         temp[k++] = a[i++];
17     while (j <= high)
18         temp[k++] = a[j++];
19     for (i = low; i <= high; i++)
20         a[i] = temp[i];
21 }
22 void mergeSort(int a[], int low, int high)
23 {
24     if (low < high)
25     {
26         int mid = (low + high) / 2;
27         mergeSort(a, low, mid);
28         mergeSort(a, mid + 1, high);
29         merge(a, low, mid, high);
30     }
31 }
32 int main()
33 {
34     int a[] = {157, 110, 147, 122, 111, 149, 151, 141, 123, 112, 117, 133};
35     int n = sizeof(a) / sizeof(a[0]);
36     int i;
37     mergeSort(a, 0, n - 1);
38     printf("Sorted array:\n");
39     for (i = 0; i < n; i++)
40         printf("%d ", a[i]);
41     return 0;
42 }

```

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 . . .

```