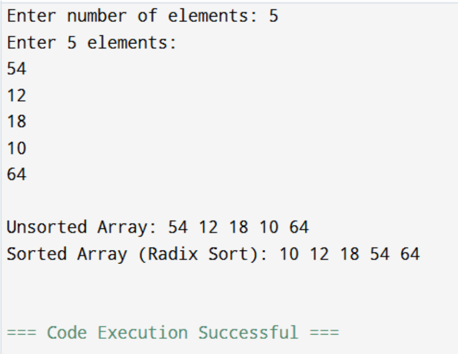
Experiment 32: Radix Sort

Code:

```
#include <stdio.h>
int getMax(int arr[], int n) {
    int max = arr[0];
    for (int i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}
void countingSort(int arr[], int n, int place) {
    int output[n];
    int count[10] = {0};
    for (int i = 0; i < n; i++) {
        int digit = (arr[i] / place) % 10;
        count[digit]++;
    }
    for (int i = 1; i < 10; i++)
        count[i] += count[i - 1];
    for (int i = n - 1; i >= 0; i--) {
        int digit = (arr[i] / place) % 10;
        output[count[digit] - 1] = arr[i];
        count[digit]--;
    }
    for (int i = 0; i < n; i++)
        arr[i] = output[i];
}
```

```
void radixSort(int arr[], int n) {
    int max = getMax(arr, n);
    for (int place = 1; max / place > 0; place *= 10)
        countingSort(arr, n, place);
}
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    int arr[50], n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nUnsorted Array: ");
    printArray(arr, n);
    radixSort(arr, n);
    printf("Sorted Array (Radix Sort): ");
    printArray(arr, n);
    return 0;
}
```

Output:



```
Enter number of elements: 5
Enter 5 elements:
54
12
18
10
64

Unsorted Array: 54 12 18 10 64
Sorted Array (Radix Sort): 10 12 18 54 64


=== Code Execution Successful ===
```