Experiment No 1.5

Redix Sort :

```
#include <iostream>
using namespace std;
class radixsort
{
  int arr[30],n;
  public:
  void getdata();
  void showdata();
  void sortLogic();
};
void radixsort :: getdata()
{
  cout<<"How many elements you require : ";
  cin>>n;
  for(int i=0;i<n;i++)
  cin>>arr[i];
}
void radixsort :: showdata()
{
  cout<<"\nFinal Output\n";
  for(int i=0;i<n;i++)
  cout<<arr[i]<<" "<<endl;
}

void radixsort :: sortLogic()
{
  //for base 10 int temp;
  int bucket[10][20], buck_count[10], b[10];
int i,j,k,r,no_of_passes=0,divisor=1,largest,pass_no;
largest=arr[0];

for(i=1;i<n;i++) //Find the largest Number
{
  if(arr[i] > largest)
  largest=arr[i];
}

while(largest > 0) //Find number of digits in largest number
{
  no_of_passes++;
  largest /= 10;
}

for(pass_no=0; pass_no < no_of_passes; pass_no++)
{
  for(k=0; k<10; k++)
  buck_count[k]=0; //Initialize bucket count
  for(i=0;i<n;i++)
  {
     r=(arr[i]/divisor) % 10;
     bucket[r][buck_count[r]++]=arr[i];
  }
  i=0; //collect elements from bucket
  for(k=0; k<10; k++)
  {
     for(j=0; j<buck_count[k]; j++)
     arr[i++] = bucket[k][j];
  }
```

```
      divisor =divisor * 10;
   }
}

int main()
{
   radixsort obj;
   obj.getdata();
   obj.sortLogic();
   obj.showdata();
   return 0;
}
```

Output :

```
How many elements you require : 5
1
3
4
10
2

Final Output
1
```

2

3

4

10