```
#include <iostream>
#include <ctime>
#include <cstdlib>
Using namespace std;
Void Generar_Numeros_Aleatorios (int num, min, max,
  cantidad) {
  srand (time(0));
  for (int i=0; i<cantidad; i++) {
      num [i] = min + rand () %
  (max - min + 1);
   }
}
  void Ordenar_Por_Cubetas (int num, n, tamanio
    cubeta) {
     int min = num [0];
     int max = num [0];
//Encontrar el mínimo y el máximo
  for (int i=1; i<n; i++) {
  if (num[i] < min) {
      min = num [i];
   }
     else if (num [i] > max) {
       max = num [i];
   }
}
```

```
//Crear un vector de cubetas
int cubetas [tamanioCubeta][n/tamanioCubeta+1]={0};
//Distribuir los números en las cubetas
for (int i=0;i<n;i++){
int indice=(num[i]-min)/((double)(max-min)/tamanioCubeta);
cubetas[indice][cubetas[indice][0]+1]=num[i];
cubetas[indice][0]++;
}
//Ordenar los números en cada cubeta (usando cualquier algoritmo de ordenamiento)
for (int i=0;i<tamanioCubeta;i++)
{
for (int j=1;j<=cubetas[i][0];j++){
for (int K=j+1;K<=cubetas[i][0];K++){
if (cubetas[i][j]>cubetas[i][K]){
swap(cubetas[i][j],cubetas[i][K]);
}
}
}
}
```

```
//Fusionar las cubetas ordenadas de nuevo
en el arreglo original
  int indice=0;
   for (int j=1; j<=cubetas [i][0]; j++){
   num [indice++]= cubetas [i] [j];
   }
  }
}
 Void Imprimir_Numeros ( int num, n) {
   for (int i =0; i<n; i++){
     cout<<num[i]<<" ";
  }
   cout << endl;
  }
   int main (){
      const int n=10;
      int num[n];
      const int min=0;
      Const int max=100;
      Const int tamanioCubeta=5;
 Generar_Numeros_Aleatorios(num, min,
 max,n);
    cout<<"Numeros originales: ";
    Imprimir_Numeros (num,n);
```

```
Ordenar_Por_Cubetas (numeros, n, tamanioCubeta);
cout<<"Números ordenados: ";
  Imprimir_Numeros (num, n);
  return 0;
}
```