

```

1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 void Intercala(vector<int>& v, int ini, int mid1, int mid2, int fim){
6     //Calculando o tamanho dos 3 subvetores
7     int tam1 = mid1 - ini + 1;
8     int tam2 = mid2 - mid1;
9     int tam3 = fim - mid2;
10
11     vector<int> left (tam1);
12     vector<int> medium (tam2);
13     vector<int> right (tam3);
14
15     for(int i = 0; i < tam1; i++){
16         left[i] = v[i+ini];
17     }
18     for(int i = 0; i < tam2; i++){
19         medium[i] = v[i + mid1 + 1];
20     }
21     for(int i = 0; i < tam3; i++){
22         right[i] = v[i + mid2 + 1];
23     }
24     //Agora eu tenho 3 subvetores e preciso organiza-los em de forma crescente no vetor original.
25
26     int i = 0; //Para subvetor left
27     int j = 0; //Para o subvetor medium
28     int k = 0; //Para o subvetor right
29     int l = ini; //Para o vetor original
30
31     while(i < tam1 && j < tam2 && k < tam3){
32         if(left[i] <= medium[j] && left[i] <= right[k]){
33             v[l] = left[i];
34             i++;
35         }else if(medium[j] <= left[i] && medium[j] <= right[k]){
36             v[l] = medium[j];
37             j++;
38         }else{
39             v[l] = right[k];
40             k++;
41         }
42         l++;
43     }
44
45     if(i == tam1 && (j != tam2 || k != tam3)){
46         while(j < tam2 && k < tam3){
47             if(medium[j] <= right[k]){
48                 v[l] = medium[j];
49                 j++;
50             }else{
51                 v[l] = right[k];
52                 k++;
53             }
54             l++;
55         }
56     }else if(j == tam2 && (i != tam1 || k != tam3)){
57         while(i < tam1 && k < tam3){
58             if(left[i] <= right[k]){
59                 v[l] = left[i];
60                 i++;
61             }else{
62                 v[l] = right[k];
63                 k++;
64             }
65             l++;
66         }
67     }else if(k == tam3 && (i != tam1 || j != tam2)){
68         while(i < tam1 && j < tam2){
69             if(left[i] <= medium[j]){
70                 v[l] = left[i];
71                 i++;
72             }else{
73                 v[l] = medium[j];
74                 j++;
75             }
76             l++;
77         }
78     }
79
80     while(i < tam1){
81         v[l] = left[i];
82         i++;
83         l++;
84     }
85     while(j < tam2){
86         v[l] = medium[j];
87         j++;
88         l++;
89     }
90     while(k < tam3){
91         v[l] = right[k];
92         k++;
93         l++;
94     }
95 }
96
97 void TripleMerge(vector<int>& v, int ini, int fim){
98     if(ini < fim){
99         int n = fim - ini + 1; //calcula o tamanho do vetor
100         int razao = n / 3; //calcula o tamanho que cada subvetor tera
101
102         int mid1 = ini + razao - 1;
103         int mid2 = ini + 2 * razao - 1;
104
105         TripleMerge(v,ini,mid1); //Chamada recursiva para o primeiro subvetor
106         TripleMerge(v,mid1+1,mid2); //Chamada recursiva para o segundo subvetor
107         TripleMerge(v,mid2+1,fim); //Chamada recursiva para o terceiro subvetor
108
109         Intercala(v,ini,mid1,mid2,fim); //Intercalando 3 subvetores em um vetor ordenado
110     }
111 }
112
113
114 }
115
116 int main(){
117     vector<int> vetor(0);
118     for(int i = vetor.size()-1; i >= 0; i--){
119         vetor[i] = i;
120         cout << i << " ";
121     }
122
123     TripleMerge(vetor,0,vetor.size()-1);
124
125     for(int i = 0; i < vetor.size(); i++){
126         cout << vetor[i] << " ";
127     }
128 }

```