

Execução

Usei a função bubble sort como método de ordenação para um vetor de caracteres e procurei implementar a intercalação e a ordenação por meio de funções, mas primeiro eu leio a fita de entrada e faço a distribuição de acordo com a quantidade de fitas e com a quantidade de memória disponível usando um loop para que continue fazendo isso ate que tudo do arquivo de entrada tenha sido lido, usa o método de ordenação para ordenar o que tiver dentro das fitas e depois intercala as fitas colocando o resultado já intercalado e ordenado dentro de um arquivo de saída, coloquei uma verificação no final do código para verificar se ele esta ordenado corretamente ou não para testar o resultado.

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <algorithm>
5
6  using namespace std;
7
8  // Função de Bubble Sort para ordenar um vetor de caracteres
9  void bubbleSort(vector<char>& arr) {
10     int n = arr.size();
11     for (int i = 0; i < n - 1; i++) {
12         for (int j = 0; j < n - i - 1; j++) {
13             if (arr[j] > arr[j + 1]) {
14                 swap(arr[j], arr[j + 1]);
15             }
16         }
17     }
18 }
19
20 // Função para intercalar arquivos e ordenar usando Bubble Sort
21 void intercalarOrdenar(const string& arquivoSaida01, const string& arquivoSaida02, const string& arquivoSaida03, const string& arquivoResultado)
22 {
23     ifstream input1(arquivoSaida01);
24     ifstream input2(arquivoSaida02);
25     ifstream input3(arquivoSaida03);
26     ofstream output(arquivoResultado);
27
28     char char1, char2, char3;
29
30     // Leitura inicial dos caracteres dos arquivos
31     input1.get(char1);
32     input2.get(char2);
```

```

92         getline(result, linhaAtual);
93         if (linhaAtual < linhaAnterior) {
94             ordenado = false;
95             break;
96         }
97         linhaAnterior = linhaAtual;
98     }
99
100     result.close();
101
102     if (ordenado) {
103         cout << "A ordenacao nao esta em ordem alfabetica." << endl;
104     } else {
105         cout << "A ordenacao esta em ordem alfabetica." << endl;
106     }
107
108     return 0;
109 }
110
62         input2.get(char2);
63     }
64     if (!input3.eof() && char3 == minChar) {
65         input3.get(char3);
66     }
67 }
68
69 // Fecha os arquivos
70 input1.close();
71 input2.close();
72 input3.close();
73 output.close();
74 }
75
76 int main() {
77     const string arquivoSaida01 = "Fita01.txt";
78     const string arquivoSaida02 = "Fita02.txt";
79     const string arquivoSaida03 = "Fita.txt"; // Nome do arquivo alterado para Fita.txt
80     const string arquivoResultado = "Resultado.txt";
81
82     // INTERCALA E ORDENA OS BLOCOS
83     intercalarOrdenar(arquivoSaida01, arquivoSaida02, arquivoSaida03, arquivoResultado);
84
85     // Teste para verificar se a ordenação está em ordem alfabética
86     ifstream result(arquivoResultado);
87     string linhaAnterior = "";
88     bool ordenado = true;
89
90     while (!result.eof()) {
91         string linhaAtual;

```

```

32     input3.get(char3);
33
34     while (!input1.eof() || !input2.eof() || !input3.eof()) {
35         vector<char> data;
36
37         // Preenche o vetor de caracteres a partir dos caracteres lidos
38         if (!input1.eof()) {
39             data.push_back(char1);
40         }
41         if (!input2.eof()) {
42             data.push_back(char2);
43         }
44         if (!input3.eof()) {
45             data.push_back(char3);
46         }
47
48         // Ordena os caracteres usando Bubble Sort
49         bubbleSort(data);
50
51         // Escolhe o menor caractere dentre os lidos
52         char minChar = data[0];
53
54         // Escreve o caractere no arquivo de saída
55         output << minChar;
56
57         // Lê o próximo caractere do arquivo correspondente
58         if (!input1.eof() && char1 == minChar) {
59             input1.get(char1);
60         }
61         if (!input2.eof() && char2 == minChar) {

```

Resultados

Fita de entrada

```

1   joaninha
2
3
4

```

Fita.txt

```
1  ajo
2
3
4
```

Fita1.txt

```
1  aahijnnos
2
3
4
```

Arquivo de Saída

```
1  aahijnno
2
3
```

Dificuldades

Tive muita dificuldade no código e fiz o possível para implementar o que foi pedido mas reparei que a separação dos blocos não funcionou corretamente, uma das dificuldades que tive foi compreender o conceito de intercalação balanceada e a separação dos blocos a lógica do código se mostrou muito complicada.