

## Estrutura de Dados 1 – Correção da questão 13 – P2

Nome: Pedro Igor Oliveira Silva Matrícula: 17/0062635

O código enviado como solução da questão:

```
1 #include <vector>
2
3 using namespace std;
4
5 vector<int> merge(const vector<int>& a, const vector<int>& b)
6 {
7     vector<int> *p;
8     vector<int> *s;
9     p = a;
10    r = b;
11    vector<int> r;
12    int i, m = 0, n = 0;
13
14    i = (p[m] >= s[n]) ? s[n] : p[m];
15
16    if (i == s[0]) {
17        ++n;
18    } else {
19        ++m;
20    }
21
22    r[0] = i;
23
24    for (int j = 1; j < (p.size() + s.size()); j++) {
25        if (p[m] >= r[j - 1]) {
26            i = p[m];
27        }
28
29        if (s[n] >= r[j - 1] && s[n] < p[m]) {
30            i = s[n];
31        }
32
33        if (p[m] == i && m < (p.size() - 1)) {
34            m++;
35        } else if (s[n] == i && n < (s.size() - 1)) {
36            n++;
37        }
38
39        r[j] = i;
40
41        if (n == (s.size() - 1) && (m == (p.size() - 1))) {
42            break;
43        }
44    }
45
46    return r;
47 }
```

Meu erro foi de sintaxe. Primeiramente, nas linhas 7 e 8, criei ponteiros para vector para receber os vector 'a' e 'b' o que era desnecessário e provocou erro de sintaxe. Na linha 10 utilizei um vector que ainda não havia sido declarado. Na linha 41 utilizei um '=' no lugar de um sinal de menos '-'. O compilador informa um erro de sintaxe na linha 35 ocasionado pela utilização de um 'else' com argumento, deveria ser utilizado um 'else if'.

Para solucionar removi os ponteiros para vector. Utilizei a própria estrutura da biblioteca vector para comparar posições ".at(i)". A lógica utilizada também foi alterada. Na nova lógica, primeiro verifico se um dos vector já está vazio, se sim, o vetor resultado é composto pelo outro vector, ou ao menos o restante dele. Se não, faço uma comparação entre a posição 'j' do vector 'a' e a posição 'k' do vector 'b'. Se 'a.at(j) > b.at(k)' o 'b' deve ser colocado no resultado e o 'k' incrementado. O mesmo acontece se 'b.at(k) > a.at(j)' porém o 'j' é incrementado e o 'a.at(j)' é adicionado ao final do vector.

Confira na próxima página a resolução pós-correção:

```

1  #include <vector>
2  #include <iostream>
3  #include<cassert>
4  using namespace std;
5  vector<int> merge(const vector<int> &a, const vector<int> &b){
6      vector<int> c;
7      int j = 0, k = 0;
8      for(int i = 0; i < (a.size()+b.size()); i++){
9          if(j >= a.size()){
10             c.push_back(b.at(k));
11             k++;
12         }
13         else if(k >= b.size()){
14             c.push_back(a.at(j));
15             j++;
16         }
17         else if(a.at(j) > b.at(k)){
18             c.push_back(b.at(k));
19             k++;
20         }
21         else{
22             c.push_back(a.at(j));
23             j++;
24         }
25     }
26     return c;
27 }

```

```

#include <vector>
#include <iostream>
using namespace std;
vector<int> merge(const vector<int> &a, const vector<int> &b){
    vector<int> c;
    int j = 0, k = 0;
    for(int i = 0; i < (a.size()+b.size()); i++){
        if(j >= a.size()){
            c.push_back(b.at(k));
            k++;
        }
        else if(k >= b.size()){
            c.push_back(a.at(j));
            j++;
        }
        else if(a.at(j) > b.at(k)){
            c.push_back(b.at(k));
            k++;
        }
        else{
            c.push_back(a.at(j));
            j++;
        }
    }
    return c;
}

```