

Problema: Busy Schedule

1 - Objetivo

Deve-se organizar os horários do menor para o maior e depois imprimir.

2 - Análise

Primeiramente são vários casos de testes, portanto deve-se continuar lendo a entrada até que seja diferente de zero.

Ao observar cada entrada basta apenas organizar os dados de forma que possam ser tratados, separados e comparados.

Os principais valores que definem a sua posição em um vetor ordenado são os valores da hora e a classificação se é p.m. ou a.m. Entretanto existe um caso especial devido ao valor 12 na hora, na realidade o valor não é 12 e sim 0.

3 - Solução

O código abaixo utiliza o método de ordenação “insertion sort” que possui complexidade quadrática. Ao inserir um valor novo no vetor são realizadas as comparações necessárias abordada na análise, e se for necessário são trocadas de posição no vetor.

A seguir está a proposta da solução na linguagem C++:

```
#include <bits/stdc++.h>

using namespace std;

double getNumber(string s) {
    string n1 = "", n2 = "";
    for (int i = 0; i < s.size(); i++) {
        if(s[i] == ':') {
            n2+= s.substr(i+1, s.size()-1);
            break;
        } else
            n1+= s[i];
    }
}
```

```

double nm1 = stoi(n1);
nm1 = nm1 == 12 ? 0 : nm1;
double nm2 = stoi(n2);
nm2/=100;
return nm1+nm2;
}

int main() {
    int T; cin >> T;
    while(T!=0) {
        vector<pair<string, string>> v(T);
        cin >> v[0].first >> v[0].second;
        for (int i = 1; i < T; i++) {
            cin >> v[i].first >> v[i].second;
            int c = i;
            while(c > 0) {
                if(v[c].second == v[c-1].second) {
                    if(getNumber(v[c].first) <= getNumber(v[c-1].first))
                        swap(v[c], v[c-1]);
                } else if(v[c].second == "a.m.") {
                    swap(v[c], v[c-1]);
                } else
                    break;
                c--;
            }
        }
        for (int i = 0; i < T; i++)
            cout << v[i].first << " " << v[i].second << '\n';
        cout << '\n';
        cin >> T;
    }
    return 0;
}

```