```cpp
// Programação Orientada a Objectos 2020/2021
// Ana Rita Videira - 5012012218

//Exame 1920 - Época Recurso - Exercicio 6


#include <cstdlib>
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <ctype.h>
#include <vector>

using namespace std;


class Excursao{

private:
    string data;
    const string destino;
    int lotacao;
    vector <int> inscritos;

public:

    // Criar objetos com todos os dados menos o de viajenates inscritos
    // Construtor
    Excursao( string d, string des, int lo): data(d), destino(des), lotacao(lo){};

    // Inscrever viajantes dados os seus passaportes: excursao3 << 123 << 789 << 123; // o segundo 123 é ignorado
    // Operador <<
    Excursao &operator<<(int p) {

        for (auto i : inscritos) {
            if (p == i)
                return *this;
        }

        inscritos.push_back(p);
        return *this;
    }


    // Transferir passageiros de uma excursão para outra: excursao3 >> excursao2;
    //Operador >>
    Excursao &operator>>(Excursao &e) {
        int lota = 0;

        for (int i = 0; i < e.lotacao && i < inscritos.size(); i++) {
            e.inscritos.push_back(inscritos[i]);
            lota++;
        }

        auto it = inscritos.begin();

        while (lota > 0) {
            it = inscritos.erase(it);
            lota--;
        }

        return *this;
    }


    //Atribuir excursões; excursao4 = excursao5;
    // Operador atribuicao
    Excursao &operator=(Excursao &p) {
        auto it = inscritos.begin();

        while (it != inscritos.end()) {
```

```
73              it = inscritos.erase(it);
74              it++;
75          }
76
77          data = p.data;
78          lotacao = p.lotacao;
79
80          p >> *this;
81
82          return *this;
83
84      }
85
86      // Aceder a um viajante específico e poder por outro no seu lugar: excursao2[1]
        = 329
87      //operador [ ]
88      int &operator[](int i) {
89          if (i > 0)
90              return inscritos[i];
91          else
92              cout << "Erro";
93      }
94
95
96     // Saber quantos passageiros estão neste momento no autocarro: int a = excursao2
97      //operador de conversao
98
99      operator int()const{
100         int contador = 0;
101         for(auto a : inscritos)
102             contador++;
103         return contador;
104     }
105
106
107     // Para testar resultados
108      string getAsString()const{
109         ostringstream os;
110         os << "Data: " << data << " Destino: " << destino << " Lotacao: " << lotacao
111         << " Inscritos: ";
112         for(auto a : inscritos)
113             os << " " << a;
114
115         return os.str();
116     }
117
118 };
119
120 // para usar o operador <<  para apresentar resultados
121 ostream &operator<<(ostream &os, const Excursao &e){
122     return os << e.getAsString();
123 };
124
125
126 int main(int argc, char** argv) {
127
128      const string destino = "Porto";
129     const string ola = "Lisboa";
130     int a;
131
132     Excursao excursao3("2020-10-01",destino,30);
133     Excursao excursao2("2021-11-11",ola,30);
134
135     excursao3 <<  123 << 456 << 789;
136
137     cout << excursao3 << endl;
138
139     a = excursao3;
140     cout << "A: " << a << endl;
141
142     cout << excursao3 << endl;
143     //excursao3[2] = 666;
144
```

```
145       //cout << excursao3 << endl;
146
147       //excursao2 = excursao3;
148
149       //cout << excursao2 << endl << excursao3;
150
151       //excursao3 << 123 << 789 << 12 << 33 << 22 << 44 ;
152       //cout << "Excursao 3: " << excursao3 << endl;
153
154
155       //excursao3 >> excursao2;
156       //cout << "Excursao 2: " << excursao2 << endl;
157
158       //cout << "Excursao 3: " << excursao3 << endl;
159
160
161       return 0;
162   }
163
164
```