```
//  File:  tuple.h
//
//Massachusetts Institute of Technology
//16.412J/6.834J Cognitive Robotics
//
//Russian Doll Search
//
//Problem Set #2
//Due: in class Wed, 3/9/05
//
//Lawrence Bush, Brian Bairstow
//{ bushl2, bairstow }@mit.edu
// _________________________________________
//
//
//  tuple.h -   Contains a tuple class.
//          Stores a tuple read in from file.
//          A tuple is an assignment to a set of variables.
//          More precisely, it is an ordered set of values
//          assigned to the ordered set of variables.
//
//          In this class, there is a set of vaiables that
//          have an assignment.
//

#ifndef _tuple_h_
#define _tuple_h_
#include <string>
#include <vector>
#include <algorithm>
#include "variable.h"
using namespace std;

// tuple class
//
//
class tuple {


public:
    tuple() {}  // default constructor
    // Constructor - assigns all variable attributes
    tuple( variables c_vars_in, int non_default_value_in )
        : c_vars(c_vars_in), non_default_value(non_default_value_in)
    {
    }






    double get_non_default_value()  const {return non_default_value;}

    // assessor operator, returns player k
    variable operator[](int k) const
    {
        return c_vars[k];
    }


    int size() { return c_vars.size(); } // returns the number of variable in variables

    // print draft list container
    void print(std::ostream & out) {
        out << "Non-Default Tuple Cost:  " << non_default_value << endl;
        out << "---------------------------------------------------"<<endl;
        c_vars.print(out);
    }


    double evaluate(variables & ca_in, vector<unsigned long long int> & operations){

        if(c_vars.matches(ca_in, operations)){

            return non_default_value;

        } else {
            return -1;
        }
    }


private:
    variables c_vars;
    double non_default_value;
    int constraint_arity;

};




#endif
```