```
#include<stdio.h>
#include<string.h>
#include<iostream>
#include<fstream>
#include<vector>
#include<set>

using namespace std;

typedef vector<char*> Example;
vector<Example*> DataSet;
vector<char*> Attributes;

class comp_str {
public:
  bool operator() (char str1[], char str2[]) const {
    return (strcmp(str1,str2)<0);
  }
};

vector<set<char*, comp_str> *> Values;

void ReadData(const char fstri[]) {

  ifstream ifile(fstri);
  char line[10000];
  char *s; char *cline;

  // legge intestazione
  ifile.getline(line,10000);
  cout << "attributes: " << endl;
  s=strtok(line,",");
  while (s!=NULL) {
    cout << s << endl;
    char *snew = new char[strlen(s)+1];
    strcpy(snew,s);
    Attributes.push_back(snew);
    Values.push_back(new set<char*, comp_str>);
    s=strtok(NULL,",");
  }

  // legge dati
  ifile.getline(line,10000);
  while (strcmp(line,"")) {
    //cout << line << endl;

    Example* e = new Example;

    int iattr=0;
    s=strtok(line,",");
    while (s!=NULL) {
      //cout << s << endl;
      char *snew = new char[strlen(s)+1];
      strcpy(snew,s);
      set<char*>::iterator it=Values[iattr]->find(snew);
      if (it==Values[iattr]->end())
        Values[iattr]->insert(snew);
      e->push_back(snew);
      s=strtok(NULL,",");
      iattr++;
    }
```

```
    DataSet.push_back(e);

    ifile.getline(line,10000);
  }
}

double frequenza(int f, char *fv) {
  int cnt=0;
  for (int i=0; i<DataSet.size(); i++)
    if (!strcmp((*DataSet[i])[f],fv)) cnt++;
  return cnt;
}

double AND(int f1, char* f1v, int f2, char* f2v) {
  int cnt=0;
  for (int i=0; i<DataSet.size(); i++)
    if (!strcmp((*DataSet[i])[f1],f1v) && !strcmp((*DataSet[i])[f2],f2v)) cnt++;
  return cnt;
}

double OR(int f1, char* f1v, int f2, char* f2v) {
  int cnt=0;
  for (int i=0; i<DataSet.size(); i++)
    if (!strcmp((*DataSet[i])[f1],f1v) || !strcmp((*DataSet[i])[f2],f2v)) cnt++;
  return cnt;
}

main() {
  ReadData("contact-lenses.csv");

  int nf=Attributes.size();

  for (int i=0; i<nf; i++) {
    cout << Values[i]->size() << endl;
  }

  // analisi della popolarita di una coppia (att:val)
  cout << "Analisi features: " << endl;
  for (int f=0; f<nf; f++) {
    cout << Attributes[f] << endl;
    for (set<char*>::iterator it=Values[f]->begin(); it!=Values[f]->end(); it++)
      cout << " " <<  *it << " " << frequenza(f,*it) << endl;
  }

  // analisi della correlazione di coppie (att1:val1) (att2:val2)
  cout << "Analisi correlazioni AND: " << endl;
  for (int f1=0; f1<nf; f1++)
    for (int f2=f1; f2<nf; f2++) {
      cout << Attributes[f1] << " vs " << Attributes[f2] << endl;
      //<< " : " << correla(f1,f2) << endl;
      for (set<char*>::iterator it1=Values[f1]->begin(); it1!=Values[f1]->end(); it1++)
        for (set<char*>::iterator it2=Values[f2]->begin(); it2!=Values[f2]->end(); it2++)
          cout << " " << *it1 << " vs " << *it2 << " " << AND(f1,*it1, f2, *it2) << endl;
    }

  cout << "Analisi correlazioni OR: " << endl;
  for (int f1=0; f1<nf; f1++)
    for (int f2=f1; f2<nf; f2++) {
      cout << Attributes[f1] << " vs " << Attributes[f2] << endl;
      //<< " : " << correla(f1,f2) << endl;
```

```
      for (set<char*>::iterator it1=Values[f1]->begin(); it1!=Values[f1]->end(); it1++)
        for (set<char*>::iterator it2=Values[f2]->begin(); it2!=Values[f2]->end(); it2++)
          cout << " " << *it1 << " vs " << *it2 << " " << OR(f1,*it1, f2, *it2) << endl;
    }
}
```