UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO – DEC

DISCIPLINA: LINGUAGENS DE PROGRAMAÇÃO 2
PROFESSOR ANTONIO CARLOS SOBIERANSKI
a.sobieranski@ufsc.br

ENUNCIADO TRABALHO T1 – "FILE MANIPULATOR HACK"

Uso de *std::vector* ou *std::pair* para desenvolvimento do problema:

Obviamente o exemplo abaixo é ruim em termos de *armazenamento (redundancy)*.
Ideal ter *functions* que retornem & ou *, ou passar o *std::pair* ou *std::vector* para a *function*.

```
vector<string> file1;
vector<string> file2;
vector<string> file3;

file1.push_back("pizza");
file1.push_back("house");
file1.push_back("car");
file1.push_back("dog");
file1.push_back("love");

file2.push_back("aaa");
file2.push_back("bbb");
file2.push_back("ccc");

file3.push_back("zaza");
file3.push_back("zipzip");
file3.push_back("free");
file3.push_back("kaltkalt");
file3.push_back("hausss");
file3.push_back("asdf");
```

```
vector<string> fileNames;
fileNames.push_back("File1");
fileNames.push_back("File2");
fileNames.push_back("File3");

vector< vector<string> > AllFiles;
AllFiles.push_back(file1);
AllFiles.push_back(file2);
AllFiles.push_back(file3);


for(size_t i=0; i<AllFiles.size(); i++) //number of files inserted
{
    cout << "Filename is :" << fileNames.at(i) << endl;

    for(size_t j=0; j<AllFiles.at(i).size(); j++) //number of word in file index #i
    {
        cout << AllFiles.at(i).at(j) << endl;
    }
    cout << endl;
}
```

```
pair<string, vector<string>> pair1;
pair<string, vector<string>> pair2;
pair<string, vector<string>> pair3;

pair1.first = "File 1";
pair1.second = file1;

pair2.first = "File 2";
pair2.second = file2;

pair3.first = "File 3";
pair3.second = file3;

vector< pair<string, vector<string>>> AllPairs;
AllPairs.push_back(pair1);
AllPairs.push_back(pair2);
AllPairs.push_back(pair3);


for(size_t i=0; i<AllPairs.size(); i++)
{
    cout << "Filename is: " << AllPairs.at(i).first << endl;

    for(size_t j=0; j<AllPairs.at(i).second.size(); j++)
    {
        cout << AllPairs.at(i).second.at(j) << endl;
    }
    cout << endl;
}
```