lab5

SOURCE

ex 1

```
#include <iostream>
#include <fstream>
#include <iomanip>

int main()
{
    int line_number = 0;
    std::ifstream even_lines("./A02.txt");
    std::ifstream odd_lines("./A01.txt");
    std::ofstream output("merged.txt");
    std::string even_line, odd_line;

    if(even_lines.is_open() && odd_lines.is_open())
    {
        while(std::getline(even_lines, even_line)
            && std::getline(odd_lines,odd_line))
        {
            output << std::setfill('0') << std::setw(3)
                << line_number << " " << odd_line << std::endl
                << std::setfill('0') << std::setw(3)
                << line_number+1 << " " << even_line << std::endl;
            line_number+=2;
        }
    }

}
```

ex2

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>

unsigned int edit_distance(const std::string& s1, const std::string& s2)
{
    const std::size_t len1 = s1.size(), len2 = s2.size();
    std::vector<std::vector<unsigned int>> d(len1 + 1, std::vector<unsigned
int>(len2 + 1));

    d[0][0] = 0;
    for(unsigned int i = 1; i <= len1; ++i) d[i][0] = i;
```

```
    for(unsigned int i = 1; i <= len2; ++i) d[0][i] = i;

    for(unsigned int i = 1; i <= len1; ++i)
        for(unsigned int j = 1; j <= len2; ++j)
                        // note that std::min({arg1, arg2, arg3}) works only
in C++11,
                        // for C++98 use std::min(std::min(arg1, arg2), arg3)
                        d[i][j] = std::min({ d[i - 1][j] + 1, d[i][j - 1] +
1, d[i - 1][j - 1] + (s1[i - 1] == s2[j - 1] ? 0 : 1) });
    return d[len1][len2];
}

std::vector<std::string> concatenate_text_files(std::ifstream f1,
std::ifstream f2)
{
    std::string s1, s2;
    std::vector<std::string> o;
    if(f1.is_open() && f1.is_open())
    {
        while (std::getline(f1,s1) && std::getline(f2, s2))
        {
            o.push_back(s1);
            o.push_back(s2);
        }
        return o;
    }
    return o;
}

std::vector<std::string> read_to_vector(std::ifstream f)
{
    std::string s;
    std::vector<std::string> v_s;
    if(f.is_open())
    {
        while(std::getline(f,s))
        {
            v_s.push_back(s);
        }
        return v_s;
    }
    return v_s;
}

void vec_to_txt(std::string file_name, std::vector<std::string> vec)
{
    std::ofstream output(file_name);
    for(int line = 0; line < vec.size(); line++)
    {
        output << vec[line] << "\n";
    }

}
```

```
void print_lev_distance(std::vector<std::string> s1,
std::vector<std::string> s2, std::string msg)
{
    std::cout<<msg<<"\n";
    for(int line = 0; line < s1.size() && line < s2.size(); line++)
    {
        if(s1[line] != s2[line])
        {
            std::cout<<s1[line]<< " " <<edit_distance(s1[line], s2[line])
<<"\n";
        }
    }
    std::cout << "\n";
}

int sum_of_lev_distance(std::vector<std::string> s1,
std::vector<std::string> s2)
{
    int sum_lev = 0;
    for(int line = 0; line < s1.size() && line < s2.size(); line++)
    {
        if(s1[line] != s2[line])
        {
            sum_lev += edit_distance(s1[line], s2[line]);
        }
    }
    return sum_lev;
}

int main()
{
    std::vector<std::string> merged_lines =
        read_to_vector(std::ifstream("./merged.txt"));
    std::vector<std::string> correct_lines =
        read_to_vector(std::ifstream("./A03.txt"));
    std::vector<std::string> A0102 =
        concatenate_text_files(std::ifstream("./A01.txt"),
std::ifstream("./A02.txt"));
    std::vector<std::string> A0201 =
         concatenate_text_files(std::ifstream("./A02.txt"),
std::ifstream("./A01.txt"));
    print_lev_distance(correct_lines, merged_lines, "A03.txt vs
merged.txt");
    std::cout<<"sum of the Levenshtein distances for A03.txt and merged.txt
"
        <<sum_of_lev_distance(merged_lines,correct_lines) << "\n";
    std::cout<<"sum of the Levenshtein distances for A03.txt and A0102.txt
"
        <<sum_of_lev_distance(A0102,correct_lines) << "\n";
    std::cout<<"sum of the Levenshtein distances for A03.txt and A0201.txt
"
        <<sum_of_lev_distance(A0201,correct_lines) << "\n";

}
```

```
output:
A03.txt vs merged.txt
Est Europa nunc unita 4
Et unita maneat 4
Una in diversitate 4
Pacem mundi augeat 4
Semper regnant in Europa 4
Fides et iustitia 4
Et libertas populorum 4
In maiore patria 4
Cives, floreat Europa 4
Opus magnum vocat vos 4
Stellae signa sunt in caelo 4
Aureae, quae iungant nos 4

sum of the Levenshtein distances for A03.txt and merged.txt 48
sum of the Levenshtein distances for A03.txt and A0102.txt 0
sum of the Levenshtein distances for A03.txt and A0201.txt 204
```