Ps5 Source Code: Makefile

```
CC= g++
CFLAGS= -g -O3 -Wall -Werror -std=c++0x
DEPS= -lsfml-system

all:    ED

ED: main.o ED.o
    $(CC) main.o ED.o -o ED $(DEPS)

main.o: main.cpp ED.hpp
    $(CC) -c main.cpp ED.hpp $(CFLAGS)

ED.o:   ED.cpp ED.hpp
    $(CC) -c ED.cpp ED.hpp $(CFLAGS)

clean:
    rm *.o
    rm *.gch
    rm ED
```

Ps5 Source Code: main.cpp

```
#include "ED.hpp"

int main(int argc, char* argv[]){
  sf::Clock clock;
  sf::Time time;

  std::string a;
  std::string b;
  std::cin >> a >> b;

  ED obj(a, b);
  int dist = obj.OptDistance();
  std::string align = obj.Alignment();
  std::cout << "Edit distance = " << dist << std::endl;
  std::cout << align << std::endl;

  time = clock.getElapsedTime();
  std::cout << "Time : " << time.asSeconds() << std::endl;
  std::cout << "Edit distance = " << dist << std::endl;

```

```
21    return 0;
22  }
```

Ps5 Source Code: ED.hpp

```
1   #ifndef ED_HPP
2   #define ED_HPP
3
4   #include <iostream>
5   #include <sstream>
6   #include <stdexcept>
7   #include <string>
8   #include <vector>
9   #include <SFML/System.hpp>
10
11  class ED{
12   public:
13      ED(std::string a, std::string b);
14      static int penalty(char a, char b){
15          if(a == b){
16          return 0;
17          }
18          else{
19          return 1;
20          }
21      }
22      static int min(int a, int b, int c){
23          int minimum = 999999;
24          if (a < minimum){
25          minimum = a;
26          }
27          if (b < minimum){
28          minimum = b;
29          }
30          if (c < minimum){
31          minimum = c;
32          }
33          return minimum;
34      }
35      int OptDistance();
36      std::string Alignment();
37   private:
38      std::string A;
39      std::string B;
```

```
40      std::vector<std::vector<int> > matrix;
41  };
42
43  #endif
```

Ps5 Source Code: ED.cpp

```
1   #include"ED.hpp"
2
3   ED::ED(std::string a, std::string b){
4       A = a;
5       B = b;
6       A = A + ' ';
7       B = B + ' ';
8   }
9
10  int ED::OptDistance()
11  {
12    int i = A.length();
13    int j = B.length();
14    int k;
15    int l;
16      for(k = 0; k <= j; k++)
17    {
18      std::vector<int> temp;
19      matrix.push_back(temp);
20
21      for(l = 0; l <= i; l++)
22      {
23        matrix.at(k).push_back(0);
24      }
25    }
26    for(k = 0; k <= j; k++){
27      matrix[k][i] = (2 * j) - (2 * k);
28    }
29    for(l = 0; l <= i; l++){
30      matrix[j][l] = (2 * i) - (2 * l);
31    }
32
33      for(k = j - 1; k >= 0; k--){
34      for(l = i - 1; l >= 0; l--){
35        matrix[k][l] = min(matrix[k+1][l] + 2, matrix[k][l+1] + 2, matrix[k+1][l+1
36      }
37    }
```

```
38
39    return matrix[0][0];
40  }
41
42
43  std::string ED::Alignment(){
44  std::string temp;
45  int j = 0;
46  int i = 0;
47  int counter = 0;
48  int counterB = 0;
49  int path;
50  temp += A[counter];
51       temp += ' ';
52       temp += B[counterB];
53       temp += ' ';
54       if(A[counter] == B[counterB])
55         temp += '0';
56       else
57         temp += '1';
58       temp += '\n';
59       counter++;
60       counterB++;
61  while((unsigned)i < A.length() && (unsigned)j < B.length()){
62      path = min(matrix[i+1][j], matrix[i][j+1], matrix[i+1][j+1]);
63      if (path == matrix[i+1][j+1]){
64       temp += A[counter];
65       temp += ' ';
66       temp += B[counterB];
67       temp += ' ';
68       if(A[counter] == B[counterB])
69         temp += '0';
70       else
71         temp += '1';
72       temp += '\n';
73       i++;
74       j++;
75       counter++;
76       counterB++;
77       path = -1;
78      }
79      else if (path == matrix[i][j+1]){
80       temp += A[counter];
81       temp += ' ';
82       temp += '-';
83       temp += ' ';
```

```
      temp += '2';
      temp += '\n';
      j++;
      counter++;
      path = -1;
     }
     else if (path == matrix[i+1][j]){
      temp += A[counter];
      temp += ' ';
      temp += '-';
      temp += ' ';
      temp += '2';
      temp += '\n';
      i++;
      counter++;
      path = -1;
     }

}
  return temp;
}
```