Gliwice,24.01.2018r.

Semester: I

Group: 1

Section: 3

Computer Programming Laboratory

Author: Agata Koclęga

E-mail: agatkoc633@student.polsl.pl

Tutor: dr inż. Piotr Fabian

1. Task topic

**Assignment 5**

Translator. Write a program that “translates” a text file by replacing words with corresponding words read from a separate dictionary. The dictionary should be read from a file once and later kept in memory.

1. Project analysis

The key to idea to this program is to getting the particular word from the text file out and organizing them in specific way into arrays. It all was about the right order. My aim was also to give the user opportunity to work with this program with different text files and dictionaries, the user can created by himself. It is very simple translator, because it translate text only one to one(one english word can be translated only for one polish word). All functions used in program are described with the comments in the source code and they were used to make the program the most simple and clear for the user, who has to interfere there to write in the proper path for the result’s place.

1. External specification

The program is intended to translate text file from language the original document is written in to the one, the user wanted it to be written in(I chose the translation from english to polish, so the exemplary use will be such). First of all user should prepare two files-one with the text destined to be translated and the second one with the proper dictionary. The file with dictionary should be adequately formatted- in one line must be placed the word in English and next to it, after space, should be the corresponding polish word. It’s important that translation of one word in english is also one word in polish. Moreover in the first line in dictionary should be given the number of English words we want to translate. The words which will be translated should be arranged in alphabetical order- firstly the words in English beginning with capital letters in alphabetical order and then the same operation with the english words starting with small letters.

In source code in bracket after „h=fopen” user should put the path where the result with translation is preferred to be placed in computer. Now the program is ready to run. In the first step user is asked to give the path of dictionary file, then the text file intended to be translated. After these two commands, in place the user wanted the result to be, the polish version of the input should be placed.

1. Internal specification

The source code of the program starts with the created function int mycmp, which task is to teams two words with themselves in the proper order. The next function in the code is int fpeek- it checks what is the following character in the text file, but does not load it in. In int main by means of fopen, the determined before file-in case of the first use of this function, the file with dictionary is being opened. Thanks to int size, fscanf reads in the number of pairs of the two corresponding words-english and polish one. After that there are created two 2D arrays with characters, for original words and their translations. In for loop, words are being read in to the program. In the next step the file with the text meant to be translated in is being read in(when the user gave it). FILE\*h with open creates and opens the file with the result of this program. With while loop the individual words from source file are read in and then ispunct, if there is present a punctuation mark, it will be read in thanks to this function to the result. Char bsearch is looking for the the given word from the file is meant to be translated in the file with the dictionary. Index determined the place, where the given word is in the dictionary. If space or comma is needed, it will be added in the end of the program thanks to the use of punctuation. In the very end we just deallocate the memory.

1. In this part a source code should be listed with appropriate comments. A source code with comments added to mail as an attachment( main.c file).
2. Testing

At the beginning the program was not working correctly-the words were not arranged in alphabetical order, the second problem was with the order of capital and small letters . After correction of these two problems, the program started working properly. I added screenshots of how it works in attachment.