YILDIZ TECHNICAL UNIVERSITY

ELECTRICAL- ELECTRONICS

FACULTY

COMPUTER ENGINEERING

DEPARTMENT

KAMRAN BALAYEV 17011904

Find and Replace Application

In this assignment I will explain the code of Find and Replace application developed with Boyer-Moore Horspool algorithm.

Boyer-Moore-Horspool is an algorithm for finding substrings into strings. This algorithm compares each characters of substring to find a word or the same characters into the string. When characters do not match, the search jumps to the next matching position in the pattern by the value indicated in the Bad Match Table.

The first step is calculating the value of each letter of the substring to create the Bad Match Table. After this operation, matching will start. In the design of this algorithm, we need 2 function one of them is storing the shifting values of characters and other one is finding the wanted string. In addition to that, we will need a function for replacing the string.

Functions

- Shift table function:
  - Purpose of this function is creating shift table for the characters. Calculation formula of shift table is: length-index-1. It has one parameter which is finded text.
- Horspool function:
  - Purpose of this function is finding the string which user wants. It has one parameter which is findedtext. It returns -1 if the string is not available in the text. Otherwise, return the start position of string.
- Replace text function:

- Purpose of this function is replacing the finded text with the text which user wants (which is called replaceText). Function has 3 parameters:
  - First one is string wanted to be find
  - second one is string which will be replaced with finded string
  - the last one is the start position of finded string.

Detailed explanations and screenshots are available below:

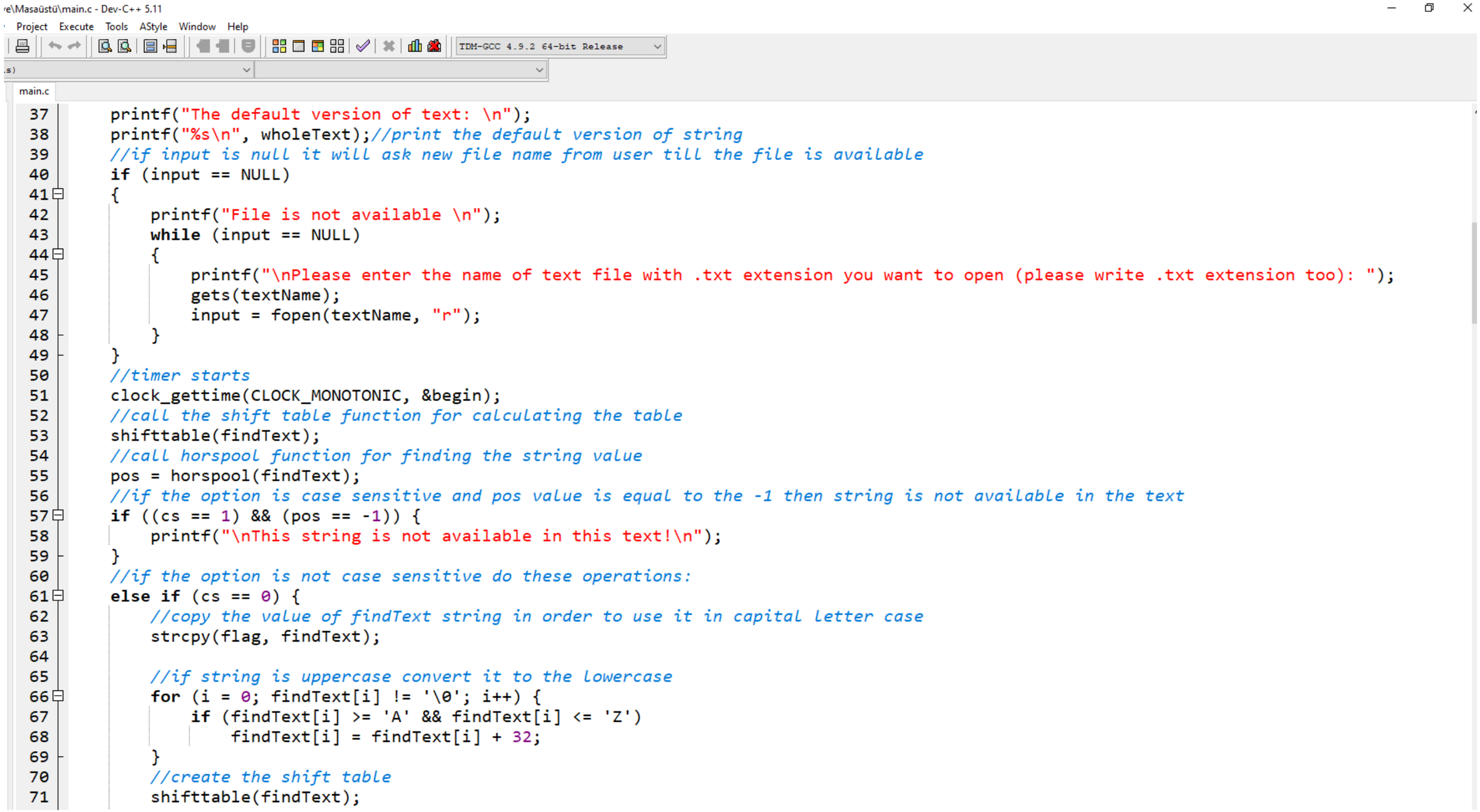
\Masaüstü\main.c - Dev-C++ 5.11

Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

main.c

```c
/*
    KAMRAN BALAYEV 17011904
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>//get empty spaces with scanf
#include <time.h>//for time calculation

struct timespec begin;/* These structs will be used in time calculation*/
struct timespec end;

void shifttable(char *);
int horspool(char *);
void replaceTextFunc(char*, char *, int);

int table[1000];//table of character shift values
char wholeText[1000];//this array stores the string read from file
int main()
{
    char findText[300], replaceText[300], textName[200], ch, i, flag[300];
    int cs, pos, counter = 0;//control for the case sensitive operation
    //get inputs from user
    printf("Please enter the text you want to find: ");
    scanf("%[^\n]s", findText);
    printf("\nPlease enter the text you want to replace with original one: ");
    getchar();//catch the new line
    scanf("%[^\n]s", replaceText);
    printf("\nPlease enter the name of text file with .txt extension you want to open (please write .txt extension too): ");
    getchar();//catch the new line
    gets(textName);
    printf("\nIf you want to search case sensitively please enter 1, otherwise enter 0: ");
    scanf("%d", &cs);
    //open and read file
    FILE *input = fopen(textName, "r");
    fgets(wholeText, 1000, input);//assign the string to the wholeText variable
```


e\Masaüstü\main.c - Dev-C++ 5.11
Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
main.c
printf("The default version of text: \n");
printf("%s\n", wholeText);//print the default version of string
//if input is null it will ask new file name from user till the file is available
if (input == NULL)
{
printf("File is not available \n");
while (input == NULL)
{
printf("\nPlease enter the name of text file with .txt extension you want to open (please write .txt extension too): ");
gets(textName);
input = fopen(textName, "r");
}
}
//timer starts
clock_gettime(CLOCK_MONOTONIC, &begin);
//call the shift table function for calculating the table
shifttable(findText);
//call horspool function for finding the string value
pos = horspool(findText);
//if the option is case sensitive and pos value is equal to the -1 then string is not available in the text
if ((cs == 1) && (pos == -1)) {
printf("\nThis string is not available in this text!\n");
}
//if the option is not case sensitive do these operations:
else if (cs == 0) {
//copy the value of findText string in order to use it in capital letter case
strcpy(flag, findText);
//if string is uppercase convert it to the lowercase
for (i = 0; findText[i] != '\0'; i++) {
if (findText[i] >= 'A' && findText[i] <= 'Z')
findText[i] = findText[i] + 32;
}
//create the shift table
shifttable(findText);

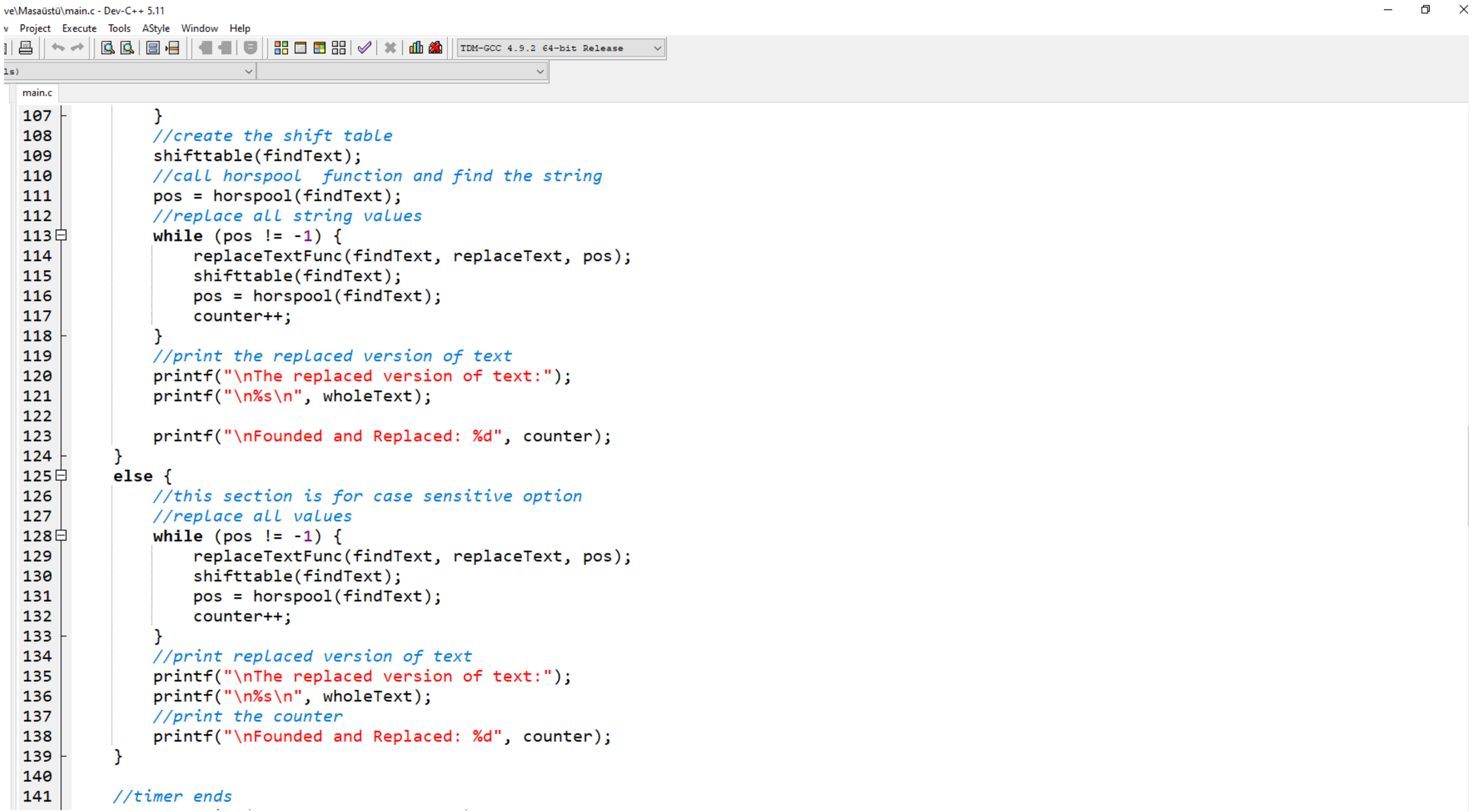
Masaüstü\main.c - Dev-C++ 5.11

Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

main.c

```
72          //call horspool  function and find the string
73          pos = horspool(findText);
74          //replace all string values
75          while (pos != -1) {
76              replaceTextFunc(findText, replaceText, pos);
77              shifttable(findText);
78              pos = horspool(findText);
79              counter++;
80          }
81          //if string is lowercase convert it to the uppercase
82          for (i = 0; findText[i] != '\0'; i++) {
83              if (findText[i] >= 'a' && findText[i] <= 'z')
84                  findText[i] = findText[i] - 32;
85          }
86          //create the shift table
87          shifttable(findText);
88          //call horspool  function and find the string
89          pos = horspool(findText);
90          //replace all string values
91          while (pos != -1) {
92              replaceTextFunc(findText, replaceText, pos);
93              shifttable(findText);
94              pos = horspool(findText);
95              counter++;
96          }
97
98          //use the previous value of findText which was stored in flag
99          strcpy(findText, flag);
100         //control the capital case letter with converting first character to the capital
101         if (findText[0] >= 'A' && findText[0] <= 'Z') {
102             findText[0] = findText[0] + 32;
103         }
104         //convert the first character of string to the lowercase
105         else if (findText[0] >= 'a' && findText[0] <= 'z') {
106             findText[0] = findText[0] - 32;
```


ve\Masaüstü\main.c - Dev-C++ 5.11
v Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
main.c
107 }
108 //create the shift table
109 shifttable(findText);
110 //call horspool function and find the string
111 pos = horspool(findText);
112 //replace all string values
113 while (pos != -1) {
114 replaceTextFunc(findText, replaceText, pos);
115 shifttable(findText);
116 pos = horspool(findText);
117 counter++;
118 }
119 //print the replaced version of text
120 printf("\nThe replaced version of text:");
121 printf("\n%s\n", wholeText);
122
123 printf("\nFounded and Replaced: %d", counter);
124 }
125 else {
126 //this section is for case sensitive option
127 //replace all values
128 while (pos != -1) {
129 replaceTextFunc(findText, replaceText, pos);
130 shifttable(findText);
131 pos = horspool(findText);
132 counter++;
133 }
134 //print replaced version of text
135 printf("\nThe replaced version of text:");
136 printf("\n%s\n", wholeText);
137 //print the counter
138 printf("\nFounded and Replaced: %d", counter);
139 }
140
141 //timer ends

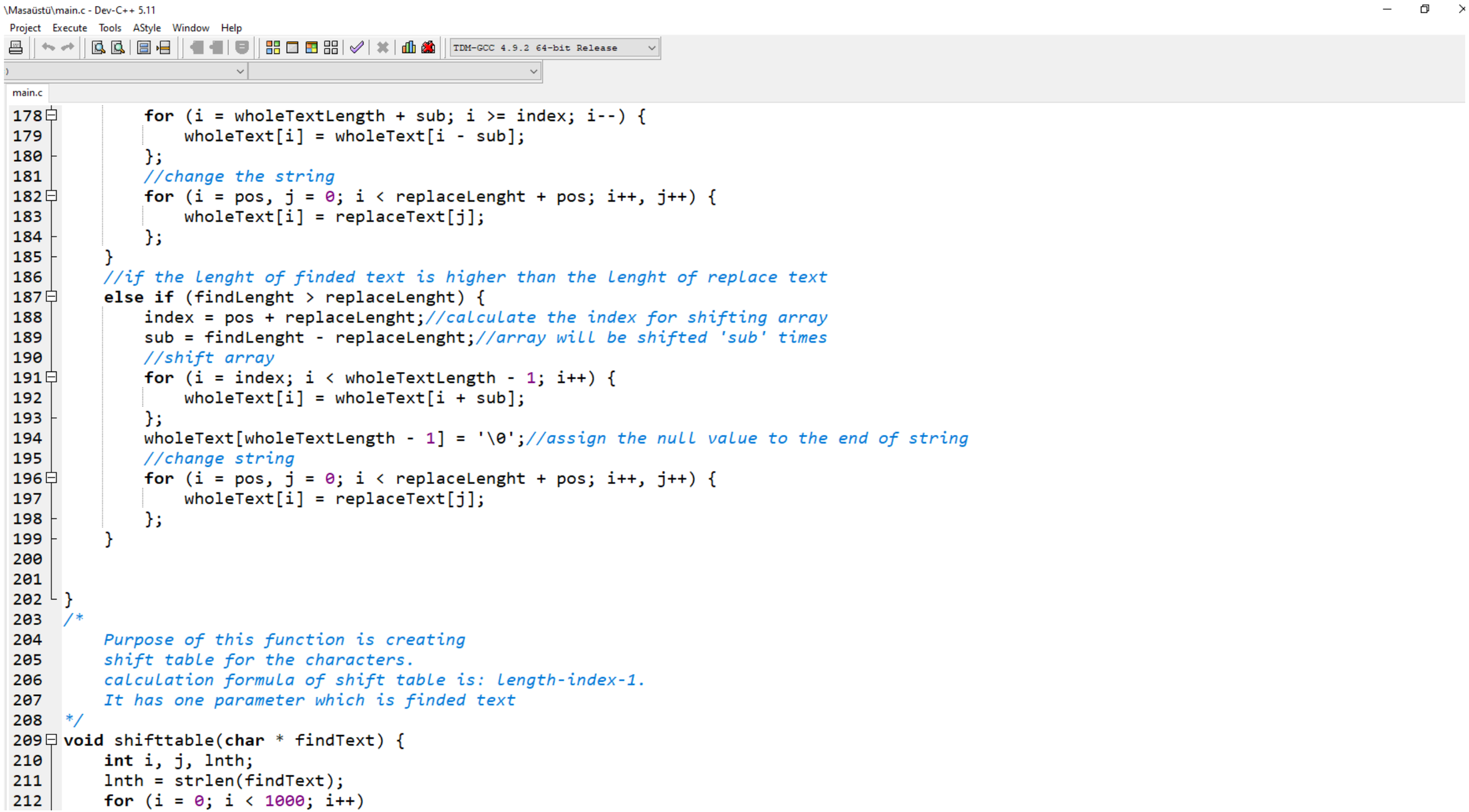
Masaüstü\main.c - Dev-C++ 5.11

Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

main.c

```
142     clock_gettime(CLOCK_MONOTONIC, &end);
143     printf("\nTime in nanosecond: %ld", (long int)(end.tv_sec-begin.tv_sec)*1000000000 + (end.tv_nsec-begin.tv_nsec) );
144
145     fclose(input);//close the file which was opened for reading
146
147     FILE *output = fopen(textName, "w+");//open file for writing the replaced version of text
148     fputs(wholeText,output);//put string to the file
149
150     fclose(output);//close the file which was opened for writing
151     return 0;
152 }
153 /*
154     Purpose of this function is replacing the finded text with the
155     text which user wants (which is called replaceText).
156     Function has 3 parameters:
157     First one is string wanted to be find ,
158     second one is string which will be replaced with finded string
159     and the last one is the start position of finded string.
160 */
161 void replaceTextFunc(char* findedText, char * replaceText, int pos) {
162     int i, index, j, sub;
163     int findLenght = strlen(findedText);//calculate the lenght of finded string
164     int replaceLenght = strlen(replaceText);//calculate the lenght of replace string
165     int wholeTextLength = strlen(wholeText);//calculate the whole text lenght
166     //if the length of string which will be replaced with finded one is equal to the lenght of finded text string
167     if (replaceLenght == findLenght) {
168         //change the string
169         for (i = pos, j = 0; i < replaceLenght + pos; i++, j++) {
170             wholeText[i] = replaceText[j];
171         };
172     }
173     //if the lenght of replace text is higher than the lenght of finded text
174     else if (replaceLenght > findLenght) {
175         index = pos + replaceLenght;//calculate the index for shifting array
176         sub = replaceLenght - findLenght;//array will be shifted 'sub' times
177         //shift array
```


\Masaüstü\main.c - Dev-C++ 5.11
Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
main.c
178 for (i = wholeTextLength + sub; i >= index; i--) {
179 wholeText[i] = wholeText[i - sub];
180 };
181 //change the string
182 for (i = pos, j = 0; i < replaceLenght + pos; i++, j++) {
183 wholeText[i] = replaceText[j];
184 };
185 }
186 //if the lenght of finded text is higher than the lenght of replace text
187 else if (findLenght > replaceLenght) {
188 index = pos + replaceLenght;//calculate the index for shifting array
189 sub = findLenght - replaceLenght;//array will be shifted 'sub' times
190 //shift array
191 for (i = index; i < wholeTextLength - 1; i++) {
192 wholeText[i] = wholeText[i + sub];
193 };
194 wholeText[wholeTextLength - 1] = '\0';//assign the null value to the end of string
195 //change string
196 for (i = pos, j = 0; i < replaceLenght + pos; i++, j++) {
197 wholeText[i] = replaceText[j];
198 };
199 }
200
201
202 }
203 /*
204 Purpose of this function is creating
205 shift table for the characters.
206 calculation formula of shift table is: length-index-1.
207 It has one parameter which is finded text
208 */
209 void shifttable(char * findText) {
210 int i, j, lnth;
211 lnth = strlen(findText);
212 for (i = 0; i < 1000; i++)

```
        table[i] = lnth;
    for (j = 0; j < lnth - 1; j++)
        table[findText[j]] = lnth - 1 - j;
}
/*
    Purpose of this function is finding the string which user wants.
    It has one parameter which is findedtext. It returns -1 if the string
    is not available in the text. Otherwise, return the start position of string.
*/
int horspool(char *findText) {
    int i, j, k, m, n;
    n = strlen(wholeText);
    m = strlen(findText);
    i = m - 1;
    while (i < n) {
        k = 0;
        while ((k < m) && (findText[m - 1 - k] == wholeText[i - k]))
            k++;
        if (k == m)
            return(i - m + 1);
        else
            i += table[wholeText[i]];
    }
    return -1;
}
```

C:\Users\balay\OneDrive\Masa³st³\main.exe

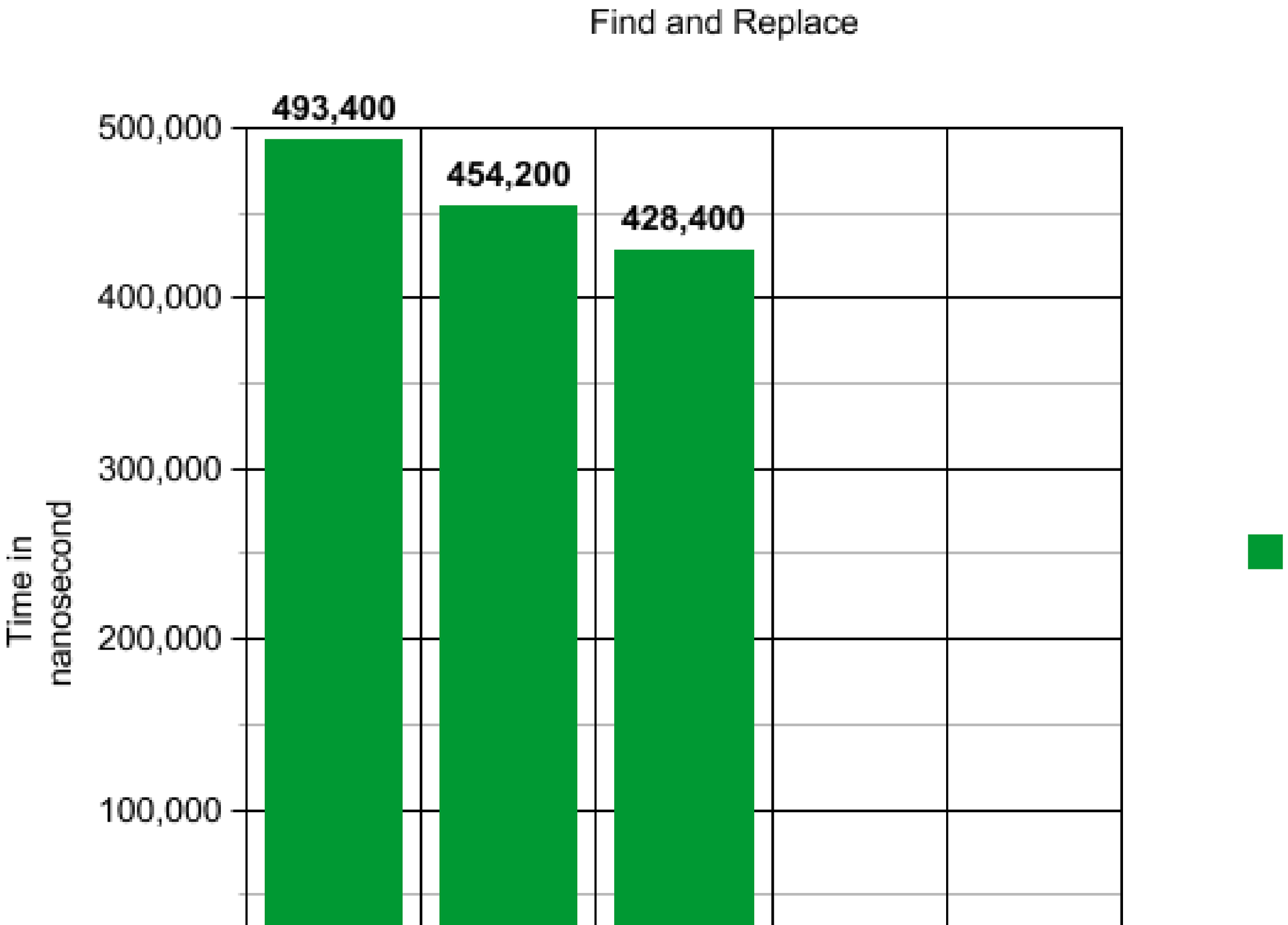
```
Please enter the text you want to find: algorithm

Please enter the text you want to replace with original one: method

Please enter the name of text file with .txt extension you want to open (please write .txt extension too): d.txt

If you want to search case sensitively please enter 1, otherwise enter 0: 1
The default version of text:
The Boyer-Moore Algorithm is considered the most efficient string matching algorithm.

The replaced version of text:
The Boyer-Moore Algorithm is considered the most efficient string matching method.

Founded and Replaced: 1
Time in nanosecond: 493400
```

C:\Users\balay\OneDrive\Masa³st³\main.exe

```
Please enter the text you want to find: algorithm

Please enter the text you want to replace with original one: method

Please enter the name of text file with .txt extension you want to open (please write .txt extension too): d.txt

If you want to search case sensitively please enter 1, otherwise enter 0: 0
The default version of text:
The Boyer-Moore Algorithm is considered the most efficient string matching algorithm.

The replaced version of text:
The Boyer-Moore method is considered the most efficient string matching method.

Founded and Replaced: 2
Time in nanosecond: 454200
--------------------------------
```

C:\Users\balay\OneDrive\Masa³st³\main.exe

```
Please enter the text you want to find: went to

Please enter the text you want to replace with original one: visited

Please enter the name of text file with .txt extension you want to open (please write .txt extension too): d.txt

If you want to search case sensitively please enter 1, otherwise enter 0: 1
The default version of text:
Wayne went to Wales to watch walruses.

The replaced version of text:
Wayne visited Wales to watch walruses.

Founded and Replaced: 1
Time in nanosecond: 428400
```


Find and Replace
493,400
454,200
428,400
500,000
400,000
300,000
200,000
100,000
Time in nanosecond