

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи № 1 з дисципліни
«Основи програмування 2. Модульне програмування»

„Текстові файли”

Варіант 8

Виконав(ла)

Дацьо Іван Іванович
(шифр, прізвище, ім'я, по батькові)

Перевірив

Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 1

Текстові файли

Мета – вивчити особливості створення та обробки текстових даних.

Варіант 8

Задача

8. Створити текстовий файл. Кожен парний рядок вихідного файлу переписати в перший новий текстовий файл, кожен непарний - у другий. У файлі з парними рядками лексично впорядкувати рядки за алфавітом. У файлі з непарними рядками впорядкувати слова в кожному із перших N рядків. Вивести вміст вихідного і створеного файлів.

Код:

C++

```
1  #include <iostream>
2  #include "header.h"
3  using namespace std;
4
5  int main(){
6      int n = numberN();
7      cout << "In the end of your text enter ctr + D\n";
8      //Назви файлів
9      string inFileName = "inputText.txt";
10     string FileName1 = "text1.txt";
11     string FileName2 = "text2.txt";
12     string changed_FileName1 = "changed_text1.txt";
13     string changed_FileName2 = "changed_text2.txt";
14     //Вектор рядків
15     vector<string> lines = inp_lines();
16     text_to_file( name: inFileName , lines);
17     vector <string> lines1;
18     vector<string> lines2;
19     //Розділення на два вектори
20     splitting (lines, lines1 ,lines2);
21     text_to_file( name: FileName1, lines: lines1);
22     text_to_file( name: FileName2, lines: lines2);
23     //завдання з не парними рядками
24     odd_number(lines1);
25     text_to_file( name: changed_FileName1, lines: lines1);
26     even_number_task(lines2 , n);
```

```

26     even_number_task(lines2 , n);
27     //Запис змінених векторів у файл
28     text_to_file( name: changed_FileName2, lines: lines2);
29     //Вивід даних з файлів
30     cout <<"Original text :" << endl;
31     outFileText( name: inFileName);
32     cout <<"text 1 :" << endl;
33     outFileText( name: FileName1 );|
34     cout <<"text 2 :" << endl;
35     outFileText( name: FileName2 );
36     cout <<"Changed text 1 :" << endl;
37     outFileText( name: changed_FileName1 );
38     cout <<"Changed text 2 :" << endl;
39     outFileText( name: changed_FileName2 );
40     return 0;
41 }

```

```

1     #pragma once
2     #include <iostream>
3     #include <string>
4     #include <fstream>
5     #include <vector>
6     #include <algorithm>
7
8     using namespace std;
9     int numberN();
10    vector<string> inp_lines();
11    void text_to_file(string name ,vector<string> lines);
12    void splitting (vector<string>lines,vector<string> & lines1 , vector<string> & lines2);
13    void odd_number(vector<string> &lines1);
14    void even_number_task(vector<string> &lines2 , int n);
15    vector <string> splitF (string s);
16    vector <string> sort_words(vector <string> words );
17    string convertVectorToString(vector<string> words);
18    void outFileText(string name);
19
20
21

```

```

1      #include "header.h"
2      int numberN(){
3          int n;
4          cout << "enter N :";
5          cin >> n;
6          return n;
7      }
8      vector<string> inp_lines(){
9          int combination = 94;
10         string line ;
11         vector<string> lines;
12         while (int (line[0] != combination)){
13             getline( &cin , &line);
14             if (!line.empty()){
15                 lines.push_back(line);
16             }
17         }
18         lines.pop_back();
19         return lines;
20     }
21     void text_to_file(string name ,vector<string> lines){
22         ofstream infile ( name);
23         for ( int i = 0; i < lines.size(); i++){
24             infile << lines[i];
25             if( i != lines.size()-1 ){
26                 infile << "\n";

```

```

26             infile << "\n";
27         }
28     }
29     infile.close();
30 }
31 void splitting (vector<string> lines,vector<string> & lines1 , vector<string> & lines2){
32     for( int i = 0 ; i < lines.size(); i ++){
33         if (i % 2 == 0){
34             lines1.push_back(lines[i]);
35         }
36         else {
37             lines2.push_back(lines[i]);
38         }
39     }
40 }
41 void odd_number(vector<string> &lines1){
42     string tmp;
43     for (int i = 0; i < lines1.size() ; i ++){
44         for ( int j = i+1; j < lines1.size() ; j ++){
45             if (tolower( lines1[i][0] ) > tolower( lines1[j][0] ) && lines1[i][0] != ' '){
46                 tmp = lines1[i];
47                 lines1[i] = lines1[j];
48                 lines1[j] = tmp;
49             }
50             else if (tolower( lines1[i][0] ) == tolower( lines1[j][0] ) || lines1[j][0] == ' '){
51                 for (int k = 1; k < lines1[i].size() ; k ++){

```

```

51         for (int k = 1; k < lines1[i].size() ; k++){
52             if (tolower( @ lines1[i][k]) > tolower( @ lines1[j][k]) || lines1[j][0] != ' ' && tolower( @ lines1[j][k]) < tolower( @ lines1[i][0])){
53                 break;
54             }
55         }
56         tmp = lines1[i];
57         lines1[i] = lines1[j];
58         lines1[j] = tmp;
59     }
60 }
61 }
62 }
63 void even_number_task(vector<string> &lines2 , int n){
64     string tmp;
65     for (int i = 0 ; i < n; i++){
66         vector<string> words = splitF( @ lines2[i]);
67         words = sort_words(words);
68         lines2[i] = convertVectorToString(words);
69     }
70 }
71 }
72 }
73 vector<string> splitF (string s){

```

```

73 vector<string> splitF (string s){
74     vector<string> words;
75     string buffer = "";
76     for (int i = 0; i < s.length(); i++) {
77         if (s.at( @ i) != ' ' and i + 1 != s.length()) {
78             buffer += s.at( @ i);
79         } else {
80             if (i + 1 == s.length()) {
81                 buffer += s.at( @ i);
82             }
83             words.push_back(buffer);
84             buffer = "";
85         }
86     }
87
88     return words;
89 }
90 }
91 vector<string> sort_words(vector <string> words ){
92     string buffer;
93
94     string tmp;
95     for (int i = 0; i < words.size() ; i++){
96         for ( int j = i + 1; j < words.size() ; j++){
97             if (tolower( @ words[i][0]) > tolower( @ words[j][0]) && words[i] != " " || words[j] == " "){

```

```

98         for (int i = 0; i < words.size() ; i++){
99             for ( int j = i + 1; j < words.size() ; j++){
100                 if (tolower( @ words[i][0]) > tolower( @ words[j][0]) && words[i] != " " || words[j] == " "){
101                     tmp = words[i];
102                     words[i] = words[j];
103                     words[j] = tmp;
104                 }
105                 else if (tolower( @ words[i][0]) == tolower( @ words[j][0]) || words[j][0] == ' '){
106                     for (int k = 1; k < words[i].size() ; k++){
107                         if (tolower( @ words[i][k]) > tolower( @ words[j][k]) || words[j][0] != ' ' && tolower( @ words[j][k]) < tolower( @ words[i][0])){
108                             break;
109                         }
110                     }
111                     tmp = words[i];
112                     words[i] = words[j];
113                     words[j] = tmp;
114                 }
115             }
116         }
117         return words;
118     }
119 }
120 string convertVectorToString(vector<string> words) {
121     string s = "";
122     for (int i = 0; i < words.size(); i++) {
123         if (i + 1 != words.size()) {

```

```

116 string convertVectorToString(vector<string> words) {
117     string s = "";
118
119     for (int i = 0; i < words.size(); i++) {
120         if (i + 1 != words.size()) {
121             s += words[i] + " ";
122         }
123         else {
124             s += words[i];
125         }
126     }
127     return s;
128 }
129 void outFileText(string name){
130     ifstream outfile (name);
131     string temp;
132     cout << outfile.rdbuf()<<"\n";
133     outfile.close();
134 }

```

Python:

```

1 from fun import *
2
3 n = int(input("Enter n :"))
4 inFileName = "text"
5 FileName1 = "text-1"
6 FileName2 = "text-2"
7 ch_FileName1 = "changed-text-1"
8 ch_FileName2 = "changed-text-2"
9 lines = input_lines()
10 writeFile(lines, inFileName)
11 lines1 = []
12 lines2 = []
13 split_for_two(lines, lines1, lines2)
14 writeFile(lines1, FileName1)
15 writeFile(lines2, FileName2)
16 ch_lines1 = task_with1(lines1)
17 ch_lines2 = task_with2(lines2, n)
18 writeFile(ch_lines1, ch_FileName1)
19 writeFile(ch_lines2, ch_FileName2)
20 print("\nOriginal text :")
21 outText(inFileName)
22 print("\nText 1 :")
23 outText(FileName1)
24 print("\nText 2 :")
25 outText(FileName2)
26 print("\nChanged text 1 :")
27 outText(ch_FileName1)

```

```

26 print("\nChanged text 1 :")
27 outText(ch_FileName1)
28 print("\nChanged text 2 :")
29 outText(ch_FileName2)
30

```

```

1  def input_lines():
2      print("Enter lines(Press ctr + s to save work)")
3      lines = []
4      line = ""
5      comb = chr(1)
6      while line != comb:
7          line = input()
8          if line:
9              lines.append(line)
10         lines.pop(-1)
11     return lines

```

```

14  def writeFile(lines, name):
15      file = open(name, "w")
16      for i in range(0, len(lines)):
17          file.write(lines[i])
18          if i != len(lines) - 1:
19              file.write("\n")
20      file.close()

```

```

23  def split_for_two(lines, lines1, lines2):
24      for i in range(0, len(lines)):
25          if i % 2 != 0:
26              lines1.append(lines[i])
27          else:

```

```

28              lines2.append(lines[i])
29          else:
30              lines2.append(lines[i])
31
32  def task_with1(lines1):
33      for i in range(0, len(lines1)):
34          for j in range(i + 1, len(lines1)):
35              if lines1[i][0].lower() > lines1[j][0].lower():
36                  lines1[i], lines1[j] = lines1[j], lines1[i]
37              elif lines1[i][0].lower() == lines1[j][0].lower():
38                  for k in range(1, len(lines1)):
39                      if lines1[i][k].lower() > lines1[j][k].lower() or lines1[j][k] != ' ':
40                          break
41                  lines1[i], lines1[j] = lines1[j], lines1[i]
42      return lines1

```

```

44  def task_with2(lines2, n):
45      words = []
46      for m in range(0, n):
47          words = lines2[m].split()
48          for i in range(0, len(words)):
49              for j in range(i + 1, len(words)):
50                  if words[i][0].lower() > words[j][0].lower():
51                      words[i], words[j] = words[j], words[i]

```

```

44 def task_with2(lines2, n):
45     words = []
46     for m in range(0, n):
47         words = lines2[m].split()
48         for i in range(0, len(words)):
49             for j in range(i + 1, len(words)):
50                 if words[i][0].lower() > words[j][0].lower():
51                     words[i], words[j] = words[j], words[i]
52                 elif words[i][0].lower() == words[j][0].lower():
53                     for k in range(1, len(words)):
54                         if words[i][k].lower() > words[j][k].lower() or words[j][k] != ' ':
55                             break
56                     words[i], words[j] = words[j], words[i]
57             lines2[m] = " ".join(words)
58     return lines2
59
60
61 def outText(name):
62     file = open(name, "r")
63     while True:
64         line = file.readline()
65         if not line:
66             break
67         print(line.strip())
68     file.close()
69

```

Тестування :

C++


```
C:\c++\lab-2-1\cmake-build-debug\lab_2_1.exe
```

```
enter N :1
```

```
In the end of your text enter ctr + D
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
dklsajfksldjf kljas dfj laksjdf lkas
kfjlsdfklj asklf klas fdjklj askdf
fjdksajf kdjsaf lkdj jkals dfjklas d
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
^D
```

```
Original text :
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
dklsajfksldjf kljas dfj laksjdf lkas
kfjlsdfklj asklf klas fdjklj askdf
fjdksajf kdjsaf lkdj jkals dfjklas d
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
text 1 :
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
kfjlsdfklj asklf klas fdjklj askdf
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
text 2 :
```

```
dklsajfksldjf kljas dfj laksjdf lkas
fjdksajf kdjsaf lkdj jkals dfjklas d
Changed text 1 :
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
kfjlsdfklj asklf klas fdjklj askdf
```

```
text 1 :
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
kfjlsdfklj asklf klas fdjklj askdf
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
text 2 :
```

```
dklsajfksldjf kljas dfj laksjdf lkas
fjdksajf kdjsaf lkdj jkals dfjklas d
Changed text 1 :
```

```
dfjaksj fdlkja sdkfj asklf dkla fd
fjdaksfjka sdjkfhjkah sfdjh ajshkf j
kfjlsdfklj asklf klas fdjklj askdf
Changed text 2 :
```

```
dfj dklsajfksldjf kljas lkas laksjdf
fjdksajf kdjsaf lkdj jkals dfjklas d
```

```
Process finished with exit code 0
```

Python:

```
Enter lines(Press ctr + a to save work
Enter n :1
jfdskah fjkdsa jdkfh jasd
fd jsalfdlkaj fdksf klasdj fk
d fjksalf lkdasf lkj aksldf jklas fd
f asjkdf aksldfjklas dfkl jasdfjlskad
f sakjldjflk askldfj klsd jflfka jsdl
^A
```

Original text :

```
jfdskah fjkdsa jdkfh jasd
fd jsalfdlkaj fdksf klasdj fk
d fjksalf lkdasf lkj aksldf jklas fd
f asjkdf aksldfjklas dfkl jasdfjlskad
f sakjldjflk askldfj klsd jflfka jsdl
```

Text 1 :

```
fd jsalfdlkaj fdksf klasdj fk
f asjkdf aksldfjklas dfkl jasdfjlskad
```

Text 2 :

```
jfdskah fjkdsa jdkfh jasd
d fjksalf lkdasf lkj aksldf jklas fd
f sakjldjflk askldfj klsd jflfka jsdl
```

Text 2 :

```
jfdskah fjkdsa jdkfh jasd
d fjksalf lkdasf lkj aksldf jklas fd
f sakjldjflk askldfj klsd jflfka jsdl
```

Changed text 1 :

```
f asjkdf aksldfjklas dfkl jasdfjlskad
fd jsalfdlkaj fdksf klasdj fk
```

Changed text 2 :

```
fjkdsa jasd jdkfh jfdskah
d fjksalf lkdasf lkj aksldf jklas fd
f sakjldjflk askldfj klsd jflfka jsdl
```

Process finished with exit code 0

Висновок : При виконанні лабораторної роботи було вивчено особливості створення та обробки текстових даних. В результаті отримано програму , яка зчитує текст консолі , зберігає його у файл та виконує завдання варіанту . А саме для парних записує в перший текстовий файл та сортує рядки . А для не парних записує в другий текстовий файл та сортує лексично слова та виводить не змінені обидва тексти та змінені .