

Міністерство освіти і науки України  
Національний технічний університет України «КПІ ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії  
Програмування інтелектуальних інформаційних систем

**ЗВІТ**  
до лабораторних робіт

**Виконав**  
**студент**

ІП-01 Паршиков Анатолій Олегович  
(№ групи, прізвище, ім'я, по батькові )

**Прийняв**

ас. Очеретяний О. К.  
(посада, прізвище, ім'я, по батькові )

Київ 2021

## Умов (Завдання 1)

Обчислювальна задача тут тривіальна: для текстового файлу ми хочемо відобразити N (наприклад, 25) найчастіших слів і відповідну частоту їх повторення, упорядковано за зменшенням. Слід обов'язково нормалізувати використання великих літер і ігнорувати стоп-слова, як «the», «for» тощо. Щоб все було просто, ми не піклуємося про порядок слів з однаковою частотою повторень. Ця обчислювальна задача відома як *termfrequency*.

Ось такий вигляд матимуть ввід і відповідно вивід результату програми:

Input:

WhitetigerslivemostlyinIndia

WildlionslivemostlyinAfrica

Output:

live - 2

mostly - 2

africa - 1

india - 1

lions - 1

tigers - 1

white - 1

wild - 1

## Опис алгоритму (Завдання 1)

1. Початок
2. Ініціалізувати всі потрібні змінні та задати статичний масив стоп-слів
3. Відкрити файл
4. Зчитувати по одному символу
5. Перевірка на велику літеру, якщо літера велика, то зробити її маленькою, якщо символ – не літера, нічого не робити.
6. Додати цей символ до тимчасової змінної, де зберігається слово.
7. Якщо символ – пробіл, або кінець строки, то перевірити чи є в тимчасовій змінній стоп-слово.
8. Якщо стоп-слово, то зчитати символ (крок 4), якщо ні – додати це слово до масиву слів.
9. Перевірити чи переповнений масив слів, якщо так, то розширити його, якщо ні, то зчитати символ (крок 4)
10. Якщо не зчитали всі символи у файлі, то зчитати символ (крок 4).
11. Підрахувати кількість повторень кожного слова в масиві, та записати йому відповідну пару у вигляді кількості повторень.
12. Відсортувати масив за спаданням кількості повторень.
13. Записати у файл пари з масиву.
14. Кінець

## Реалізація (Завдання 1)

```
using System.IO;

namespace MP1._1
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] stopWord = new[] { "i", "me", "my", "myself", "we",
"our", "ours", "ourselves", "you", "your",
"yours", "yourself", "yourselves", "he", "him", "his",
"himself", "she", "her", "hers", "herself",
"it", "its", "itself", "they", "them", "their", "theirs",
"themselves", "what", "which", "who", "whom",
"this", "that", "these", "those", "am", "is", "are", "was",
"were", "be", "been", "being", "have", "has",
"had", "having", "do", "does", "did", "doing", "a", "an",
"the", "and", "but", "if", "or", "because",
"as", "until", "while", "of", "at", "by", "for", "with",
"about", "against", "between", "into", "through",
"during", "before", "after", "above", "below", "to", "from",
"up", "down", "in", "out", "on", "off", "over",
"under", "again", "further", "then", "once", "here", "there",
"when", "where", "why", "how", "all", "any",
"both", "each", "few", "more", "most", "other", "some",
"such", "no", "nor", "not", "only", "own", "same",
"so", "than", "too", "very", "s", "t", "can", "will", "just",
```

```

"don", "should", "now" };

string[] words = new string[10];
int lengthWords = 10;
int currentLengthWords = 0;
string path = "engl.txt";
using (StreamReader reader = new StreamReader(path))
{
    string word = "";
    ReadChar:
        char temp = (char)reader.Read();

        if (temp >= 'A' && temp <= 'Z')
            temp += (char)32; //remove capital letters

        if (temp >= 'a' && temp <= 'z')
        {
            word = word + temp;
            goto ReadChar;
        }

        if (temp == ' ' || temp == '\n') //is end of the word
        {
            if (word == "")
            {
                goto ReadChar;
            }
            int indexStopWord = 0;
            CheckStopWord:
            if (indexStopWord != stopWord.Length)
            {
                if (stopWord[indexStopWord] == word)
                {
                    word = "";
                    goto ReadChar;
                }

                indexStopWord++;
                goto CheckStopWord;
            }

            words[currentLengthWords] = word;
            word = "";
            currentLengthWords++;
        }

        if (lengthWords*0.8 <= currentLengthWords) //expand array
        {
            string[] tempWords = words;
            int currentIndex = 0;
            lengthWords *= 2;
            words = new string[lengthWords];
            CopyArray1:
                words[currentIndex] = tempWords[currentIndex];
                currentIndex++;
                if (currentIndex != currentLengthWords)
                {
                    goto CopyArray1;
                }
                goto ReadChar;
        }

        if (!reader.EndOfStream)

```

```

        {
            goto ReadChar;
        }
    }

    (string, int)[] numberOfWords = new (string, int)[10];

    numberOfWords[0] = (words[0], 1);
    int currentLengthNumberOfWords = 1;
    int lengthNumberOfWords = 10;
    int wordIndex = 1;
    int i = 0;
    CheckWithAllSavedWords:
        if (words[wordIndex] != numberOfWords[i].Item1 && wordIndex
!= currentLengthWords)
        {
            i++;
            if (currentLengthNumberOfWords == i)
            {
                numberOfWords[currentLengthNumberOfWords] =
(words[wordIndex], 1);
                currentLengthNumberOfWords++;
                i = 0;
                wordIndex++;
            }

            if (lengthNumberOfWords*0.8 <=
currentLengthNumberOfWords) //expand array
            {
                (string, int)[] tempWords = numberOfWords;
                int currentIndex = 0;
                lengthNumberOfWords *= 2;
                numberOfWords = new (string,
int)[lengthNumberOfWords];
                CopyArray2:
                numberOfWords[currentIndex] =
tempWords[currentIndex];
                currentIndex++;
                if (currentIndex != currentLengthNumberOfWords)
                {
                    goto CopyArray2;
                }
            }
            goto CheckWithAllSavedWords;
        }
    if(words[wordIndex] == numberOfWords[i].Item1)
    {
        numberOfWords[i].Item2++;
        i = 0;
        wordIndex++;
        goto CheckWithAllSavedWords;
    }

    int index1 = 0;
    int index2 = 0;
    Sort:
    if (numberOfWords[index1].Item2 <
numberOfWords[index2].Item2)
    {
        (string, int) temp = numberOfWords[index1];
        numberOfWords[index1] = numberOfWords[index2];
        numberOfWords[index2] = temp;
    }

```

```

        index2++;
        if (index2 == currentLengthNumberOfWords)
        {
            index1++;
            index2 = index1;
            goto Sort;
        }
        if(index1 != currentLengthNumberOfWords)
            goto Sort;

        using StreamWriter writer = new StreamWriter("output.txt");
        int count = 0;
        OutputNumberOfWord:
        if (currentLengthNumberOfWords != count)
        {
            writer.WriteLine($"{numberOfWords[count].Item1} -
{numberOfWords[count].Item2} repeat");
            count++;
            goto OutputNumberOfWord;
        }
    }
}

```

## Результат роботи(Завдання 1)

engl.txt – Блокнот

Файл Правка Формат Вид Справка

After my summer camp ended my whole family decided to visit my grandparent's village It is situated away from the city and is full of lush and greenery love spending time with my grandparents so I was very excited to visit their villag

We stayed there for over two weeks and that throughout that time was at my best behavior used to get up at o'clock in the morning to take a stroll with my grandfather Then all the family members would get together to have breakfast full of healthy things and fresh fruits from the farm nearby

In the evening we would sit by our grandmother's bed and listen to stories Then my siblings and used to go to the terrace and play games We would also gaze at their stars all night as we did not get that in the city

In short this is how spent my summer vacation It was definitely one of the best times of my lives I got to learn new things and spend valuable time with my grandparents I feel blessed to have a loving family with whom I can enjoy my summer vacation

\*output.txt – Блокнот

Файл Правка Формат Вид Справка

summer - 3 repeat  
family - 3 repeat  
grandparents - 3 repeat  
time - 3 repeat  
get - 3 repeat  
would - 3 repeat  
full - 2 repeat  
visit - 2 repeat  
best - 2 repeat  
used - 2 repeat  
village - 2 repeat  
city - 2 repeat  
things - 2 repeat  
lush - 1 repeat  
greenery - 1 repeat  
love - 1 repeat  
spending - 1 repeat  
whole - 1 repeat  
excited - 1 repeat  
stayed - 1 repeat  
two - 1 repeat  
weeks - 1 repeat  
throughout - 1 repeat  
camp - 1 repeat  
behavior - 1 repeat  
situated - 1 repeat

## Умова (Завдання 2)

Тепер, нам потрібно виконати задачу, що називається словниковим індексуванням. Для текстового файлу виведіть усі слова в алфавітному порядку разом із номерами сторінок, на яких ці слова знаходяться. Ігноруйте всі слова, які зустрічаються більше 100 разів. Припустимо, що сторінка являє собою послідовність із 45 рядків. Наприклад, якщо взяти книгу *Pride and Prejudice*, перші кілька записів індексу будуть:

```
abatement - 89
abhorrence - 101, 145, 152, 241, 274, 281
abhorrent - 253
abide - 158, 292
```

## Опис алгоритму (Завдання 2)

1. Початок
2. Ініціалізувати всі потрібні змінні та задати статичний масив стоп-слів
3. Відкрити файл
4. Зчитувати по одному символу
5. Перевірка на велику літеру, якщо літера велика, то зробити її маленькою, якщо символ – не літера, нічого не робити.
6. Додати цей символ до тимчасової змінної, де зберігається слово.
7. Якщо символ – пробіл, або кінець строки, то перевірити чи є в тимчасовій змінній стоп-слово, а також перевірити, якщо це кінець строки, то додати збільшити тимчасову змінну на один (щоб знати який рядок).
8. Якщо слово має більш, ніж 101 повторення, то нічого не робити, якщо менше, то додати його до масиву, разом з номер сторінки
9. Якщо стоп-слово, то зчитати символ (крок 4), якщо ні – додати це слово до масиву слів.
10. Перевірити чи переповнений масив слів, якщо так, то розширити його, якщо ні, то зчитати символ (крок 4)
11. Якщо не зчитали всі символи у файлі, то зчитати символ (крок 4).
12. Підрахувати кількість повторень кожного слова в масиві, та записати йому відповідну пару у вигляді кількості повторень.
13. Відсортувати масив в алфавітному порядку.
14. Записати у файл слово, та сторінки, на яких воно знаходиться.
15. Кінець

## Реалізація (Завдання 2)

```
using System.IO;

namespace MP1._1
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] stopWord = new[] { "i", "me", "my", "myself", "we",
"our", "ours", "ourselves", "you", "your",
"yours", "yourself", "yourselves", "he", "him", "his",
"himself", "she", "her", "hers", "herself",
"it", "its", "itself", "they", "them", "their", "theirs",
"themselves", "what", "which", "who", "whom",
"this", "that", "these", "those", "am", "is", "are", "was",
"were", "be", "been", "being", "have", "has",
"had", "having", "do", "does", "did", "doing", "a", "an",
"the", "and", "but", "if", "or", "because",
"as", "until", "while", "of", "at", "by", "for", "with",
"about", "against", "between", "into", "through",
"during", "before", "after", "above", "below", "to", "from",
"up", "down", "in", "out", "on", "off", "over",
"under", "again", "further", "then", "once", "here", "there",
```



```

"when", "where", "why", "how", "all", "any",
    "both", "each", "few", "more", "most", "other", "some",
"such", "no", "nor", "not", "only", "own", "same",
    "so", "than", "too", "very", "s", "t", "can", "will", "just",
"don", "should", "now" };
    (string, int[], int)[] InfoWords = new (string, int[], int)[10];
    int currentLengthInfoWords = 0;
    int lengthInfoWords = 10;
    int numberOfLine = 0;

    string path = "input.txt";
    string word = "";
    using (StreamReader reader = new StreamReader(path))
    {
        ReadChar:
        char temp = (char)reader.Read();

        if (temp >= 'A' && temp <= 'Z')
            temp += (char)32; //remove capital letters

        if (temp >= 'a' && temp <= 'z')
        {
            word = word + temp;
            goto ReadChar;
        }

        if (temp == ' ' || temp == '\n') //is end of the word
        {
            if (temp == '\n')
            {
                numberOfLine++;
            }
            if (word == "")
            {
                goto ReadChar;
            }
            int indexStopWord = 0;
            CheckStopWord:
            if (indexStopWord != stopWord.Length)
            {
                if (stopWord[indexStopWord] == word)
                {
                    word = "";
                    goto ReadChar;
                }

                indexStopWord++;
                goto CheckStopWord;
            }

            int count = 0;
            SearchWord:
            if (count != currentLengthInfoWords)
            {
                if (InfoWords[count].Item1 == word)
                {
                    if (InfoWords[count].Item3 <= 100)
                    {
InfoWords[count].Item2[InfoWords[count].Item3] = numberOfLine / 45 + 1;
                        InfoWords[count].Item3++;
                    }
                }

                word = "";

```

```

        goto ReadChar;
    }
    count++;
    goto SearchWord;
}

int[] pages = new int[101];
pages[0] = numberOfLine / 45 + 1;

InfoWords[currentLengthInfoWords] = (word, pages, 1);
word = "";
currentLengthInfoWords++;
}

if (lengthInfoWords*0.8 <= currentLengthInfoWords)
//expand array
{
    (string, int[], int)[] tempInfoWords = InfoWords;
    int currentIndex = 0;
    lengthInfoWords *= 2;
    InfoWords = new (string, int[],
int) [lengthInfoWords];
    CopyArray1:
        InfoWords[currentIndex] =
tempInfoWords[currentIndex];
        currentIndex++;
        if (currentIndex != currentLengthInfoWords)
        {
            goto CopyArray1;
        }
    }
    if (!reader.EndOfStream)
        goto ReadChar;

}

int index1 = 0;
int index2 = 0;
Sort:
int charIndex = 0;
int lengthCharIndex;
bool swap=false;

if
(InfoWords[index1].Item1.Length<InfoWords[index2].Item1.Length)
{
    lengthCharIndex = InfoWords[index1].Item1.Length;
}
else
{
    lengthCharIndex = InfoWords[index2].Item1.Length;
    swap=true;
}

CompareTwoWorld:
    if (InfoWords[index1].Item1[charIndex] >
InfoWords[index2].Item1[charIndex])
    {
        swap = true;
    }
    else if (InfoWords[index1].Item1[charIndex] <
InfoWords[index2].Item1[charIndex])
    {
        swap = false;
    }
}

```

```

    }
    else
    {
        charIndex++;
        if (charIndex < lengthCharIndex)
        {
            goto CompareTwoWorld;
        }
    }

    if (swap)
    {
        (string, int[], int) temp = InfoWords[index1];
        InfoWords[index1] = InfoWords[index2];
        InfoWords[index2] = temp;
    }

    index2++;
    if (index2 == currentLengthInfoWords)
    {
        index1++;
        index2 = index1;
    }
    if(index1 != currentLengthInfoWords)
        goto Sort;

    using StreamWriter writer = new StreamWriter("output.txt");
    int countOutput = 0;
    int countWord = 0;
    OutputWord:
    if (currentLengthInfoWords != countOutput)
    {
        if (InfoWords[countOutput].Item3 == 101)
        {
            countOutput++;
            goto OutputWord;
        }
        writer.Write($"{InfoWords[countOutput].Item1} - ");

        OutputPage:
        if (countWord != InfoWords[countOutput].Item3 - 1)
        {
            writer.Write($"{InfoWords[countOutput].Item2[countWord]},");

            countWord++;
            goto OutputPage;
        }

        writer.Write($"{InfoWords[countOutput].Item2[InfoWords[countOutput].Item3 - 1]}");

        writer.WriteLine();

        countWord = 0;
        countOutput++;
        goto OutputWord;
    }
}
}
}

```

## Результат роботи (Завдання 2)

\*input.txt – Блокнот

ФайлПравкаФорматВидСправка

The Project Gutenberg eBook of Pride and Prejudice, by Jane Austen

This eBook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at [www.gutenberg.org](http://www.gutenberg.org). If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Pride and Prejudice

Author: Jane Austen

Release Date: June, 1998 [eBook #1342]  
[Most recently updated: August 23, 2021]

Language: English

Character set encoding: UTF-8

Produced by: Anonymous Volunteers and David Widger

\*\*\* START OF THE PROJECT GUTENBERG EBOOK PRIDE AND PREJUDICE \*\*\*

\*output.txt – Блокнот

ФайлПравкаФорматВидСправка

abatement - 99

abhorrence - 111, 160, 167, 263, 299, 306

abhorrent - 276

abide - 174, 318

abiding - 177

abilities - 72, 74, 107, 155, 171, 194

able - 19, 37, 58, 78, 84, 86, 88, 91, 98, 101, 107, 109, 110, 120, 126, 130,

ablution - 119

abode - 59, 60, 66, 110, 122, 130, 176, 260

abominable - 32, 51, 71, 71, 122, 161

abominably - 48, 133, 269, 299

abominate - 263, 296

abound - 101

abroad - 194, 196, 233, 288

abrupt - 203

abruptly - 41, 155

abruptness - 198, 198

absence - 54, 56, 64, 77, 78, 78, 90, 99, 99, 100, 106, 106, 110, 111, 127, 150, 17

absent - 31, 199, 225, 229

absolute - 78, 227, 253, 308

absolutely - 17, 25, 32, 92, 94, 125, 147, 166, 167, 171, 190, 203, 242, 260, 269,

absurd - 61, 163, 171, 296, 302

absurdities - 127, 217

absurdity - 189

abundant - 227

abundantly - 67, 85, 125