



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Лабораторна робота №1  
Імперативне програмування  
**Мультипарадигменне програмування**

Виконав  
студент групи ІТ-03:

Куксюк В.І.

Перевірив:

Очеретяний О. К.

Київ 2021

### **Мета роботи:**

Практична робота складається із трьох завдань, які самі по собі є досить простими. Але, оскільки задача - зрозуміти, як писали код наші славні пращури у 1950-х, ми введемо кілька обмежень:

Заборонено використовувати функції

Заборонено використовувати цикли

Для виконання потрібно взяти мову, що підтримує конструкцію GOTO

### **Завдання:**

### **[Код на GitHub](#)**

### **Алгоритм task1**

Алгоритм зчитує stopwords за допомогою `os.Open()` та `scanner.Scan()` порядково, після чого йде посимвольна обробка. У випадку коли символ є пробілом, комою чи крапкою зчитані символи являються словом та добавляються до масиву `stopwords`. Коли всі рядки з файлу прочитано файл замінюється на текст та виконується те ж саме, тільки відбувається перевірка на велику літеру за допомогою двох масивів з великими та малими літерами, що відповідають одне одному. Якщо слово не було використано до цього, воно добавляється в масив, якщо воно вже є, то відповідний йому запис в масиві `wordsAmount` інкрементується. Після завершення зчитування масив частоти слів сортується бальбушковим сортуванням, відповідно масив `words` також відповідно йому сортується. Після чого відображається потрібна кількість слів минуючи `stopwords`.

## Результат виконання завдання #1

```
PS D:\KPI\Course_2\Multi-paradigm programming\lab1> go run task1.go
1 ) sed - 120
2 ) ut - 107
3 ) in - 99
4 ) et - 93
5 ) at - 84
6 ) nunc - 80
7 ) eget - 79
8 ) vitae - 73
9 ) id - 72
10 ) enim - 69
11 ) arcu - 68
12 ) quis - 66
13 ) eu - 65
14 ) viverra - 64
15 ) non - 61
16 ) risus - 60
17 ) aliquam - 60
18 ) egestas - 59
19 ) ac - 59
20 ) massa - 59
21 ) pellentesque - 58
22 ) tincidunt - 57
23 ) mauris - 53
24 ) tortor - 52
25 ) diam - 51
PS D:\KPI\Course_2\Multi-paradigm programming\lab1> █
```

## Результат виконання завдання #2

### Алгоритм task2

Відмінність від минулого алгоритму в тому, що в масиви, що знаходяться в масиві заносяться дані про те, на якій сторінці є це слово. Сорткування відбувається за Алфавітом, а не за кількістю повторів.

```
PS D:\KPI\Course_2\Multi-paradigm programming\lab1> go run task2.go
1) a - 1 2 3
2) ac - 1 2 3
3) accumsan - 1 2
4) adipiscing - 1 2 3
5) aenean - 1 2 3
6) aliqua - 1
7) aliquam - 1 2 3
8) aliquet - 1 2 3
9) ante - 1 2 3
10) arcu - 1 2 3
11) at - 1 2 3
12) auctor - 1 2 3
13) augue - 1 2 3
14) bibendum - 1 2 3
15) blandit - 1 2
16) commodo - 1 2 3
17) condimentum - 1 2
18) congue - 1 2
19) consectetur - 1 2 3
20) consequat - 1 2
21) convallis - 1 2 3
22) cras - 1 2 3
23) cum - 1
24) curabitur - 1 2 3
25) cursus - 1 2 3
26) dapibus - 1
27) diam - 1 2 3
28) dictum - 1 2 3
29) dictumst - 1 2 3
30) dignissim - 1 2 3
```

**Висновок:**

Під час виконання даної роботи я ознайомився з основами імперативного програмування та конструкцією GOTO.