

## Versiunea secventiala

~Proiect APD~

Safta Maria-Andreea

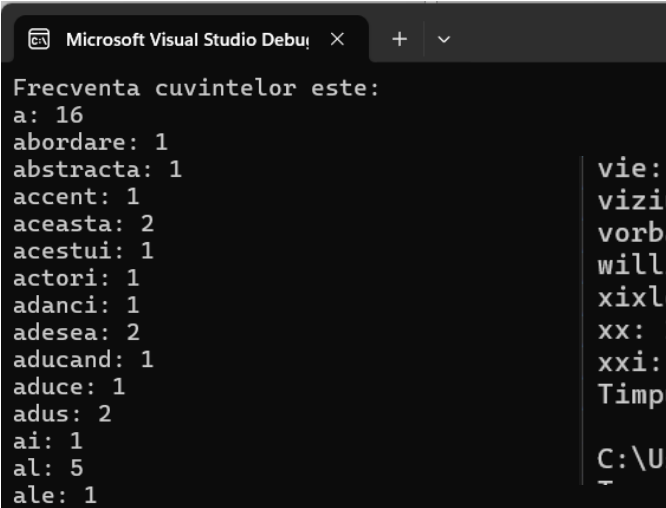
### Tema proiect : Text Processing

Pentru realizarea versiunii secventiale a acestei tema m-am folosit de functia de tokenizare si normalizare a unui text , iar ca limbaj de programare am folosit C++.

Programul citeste dintr-un fisier text , input.txt , continutul acestuia , care cu ajutorul functiei “normalizeText” transforma toate caracterele in litere mici pentru a nu se face diferente de capitalizare .

Functia “tokenize” prezinta un flux de string “std::istream” , cu ajutorul caruia citeste textul cuvant cu cuvant si apoi elimina caracterele de punctuatie “std::remove\_if”.Aceasta functie se foloseste de regula pentru o mai buna analiza a cuvintelor . Pentru normalizarea si tokenizarea textului se foloseste “countWordFrequencies”, care contorizeaza frecventa fiecarui cuvant , sorteaza cuvintele si le ordoneaza alfabetic . In functia “main” se realizeaza deschiderea fisierului input.txt , se verifica deschiderea acestuia , in cazul in care nu se deschide de va afisa un mesaj si in cazul in care se deschide va incepe sa citeasca continutul cu “std::stringstream” . De asemenea pentru masurarea timpului de executie si afisarea acestuia in milisecunde se va folosi biblioteca “#include <chrono>” .

Date experimentale:



```
Microsoft Visual Studio Debug Console
Frecventa cuvintelor este:
a: 16
abordare: 1
abstracta: 1
accent: 1
aceasta: 2
acestui: 1
actori: 1
adanci: 1
adesea: 2
aducand: 1
aduce: 1
adus: 2
ai: 1
al: 5
ale: 1
vie: 1
viziune: 1
vorba: 1
william: 1
xixlea: 1
xx: 1
xxi: 1
Timpul de executie: 64.6492 milisecunde
C:\Users\Andreea\Desktop\AN3_sem2\APD\TextPro
```

```
vie: 1
viziune: 1
vorba: 1
william: 1
xixlea: 1
xx: 1
xxi: 1
Timpul de executie: 66.8936 milisecunde

C:\Users\Andreea\Desktop\AN3_sem2\APD\TextP
To automatically close the console when deb
le when debugging stops
```