

Tema 2

Cerințe generale:

1. Pe baza cunoștințelor obținute la curs și laborator, să se implementeze minim două cerințe din cele de mai jos.

Cerințe individuale:

1. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care conțin număr par de vocale și un număr divizibil cu 3 de consoane folosind tehnica map - reduce.

De exemplu:

```
input := [][]string{
{"aabbb", "ebep", "blablablaa", "hijk", "wswww"},
{"abba", "eeeppp", "cocor", "ppppppaa", "qwerty", "acasq"},
{"lalala", "lalal", "papapa", "papap"}} se va afisa 2.33
```

2. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor atât litere cât și cifre. Să se afle numărul mediu de cuvinte care sunt palindrom folosind tehnica map - reduce.

De exemplu:

```
input := [][]string{
{"a1551a", "parc", "ana", "minim", "1pcl3"},
{"calabalac", "tivit", "leu", "zece10", "ploaie", "9ana9"},
{"lalalal", "tema", "papa", "ger"}} se va afisa 2.33
```

3. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care sunt scrise în limba păsărească folosind tehnica map - reduce. Un cuvânt este scris în păsărească dacă după fiecare vocală este litera p.

De exemplu:

```
input := [][]string{
{"apap", "paprc", "apnap", "mipnipm", "copil"},
{"cepr", "program", "lepu", "zepcep", "golang", "tema"},
{"par", "impar", "papap", "gepr"}} se va afisa 3.0
```

4. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care încep și se termină cu o vocală folosind tehnica map - reduce.

De exemplu:

```
input := [][]string{
{"ana", "parc", "impare", "era", "copil"},
{"cer", "program", "leu", "alee", "golang", "info"},
{"inima", "impar", "apa", "eleve"}} se va afisa 2.66
```

5. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care sunt anagrame ale cuvântului *facultate* folosind tehnica map - reduce. Anagramă se numește un cuvânt sau o frază care se obține prin schimbarea ordinii literelor unui alt cuvânt sau ale unei alte fraze.

De exemplu:

```
input := [][]string{
{"acutatef", "parc", "cultateaf", "faculatet", "copil"},
{"cer", "tatefacul", "leu", "alee", "golang", "ultatefac"},
{"tefaculta", "impar", "apa", "eleve"}} se va afisa 2.00
```

6. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care încep și se termina cu litera mare și au un număr par de litere mici folosind tehnica map - reduce.

De exemplu:

```
input := [][]string{
{"AcasA", "CasA", "FacultatE", "SisTemE", "distribuite"},
{"GolanG", "map", "reduce", "Problema", "Tema", "ProieCt"},
{"LicentA", "semestru", "ALGORitM", "StuDent"}} se va afisa 1.66
```

7. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care conțin cel puțin două diacritice folosind tehnica map - reduce.

De exemplu:

```
input := [][]string{
{"țânțar", "carte", "ulcior", "copac", "plante"},
{"beci", "", "mlăștinos", "astronaut", "stele", "planete"},
{"floare", "somn", "șosetă", "scârțar"}} se va afisa 1.33
```

8. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care își au perechea pe care s-a aplicat un

cifru de substituție (prima litera din alfabet se schimbă cu ultima, a doua cu penultima etc.
) folosind tehnica map - reduce.

De exemplu:

Pentru cuvântul **c a r t e** perechea s-a pe care s-a aplicat cifrul de substituție este
x z i g w

Pentru cuvântul **f o a i e** perechea s-a pe care s-a aplicat cifrul de substituție este
u e z r v

Ceea ce înseamnă că, pentru

```
input := []string{
{"prajitura", "camion", "foaie", "liliac", "uezrv"},
{"carte", "trofeu", "xzigw", "laptop", "scris", "muzica"},
{"pictura", "telefon", "parapanta", "catel"}} se va afisa 0.66
```

9. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte ce își au perechea cu care rimează folosind tehnica map - reduce.

De exemplu:

```
input := []string{
{"stele", "mele", "borcan", "vajnic", "strașnic"},
{"crocodil", "garnisit", "mușețel", "făurit", "arhanghel", "noapte"},
{"lampă", "sine", "cine", "toriște"}} se va afisa 1.33
```

10. Se dă un vector de vectori, care conțin mai multe cuvinte ce pot avea în componența lor doar litere. Să se afle numărul mediu de cuvinte care au alternanța consoană - vocală folosind tehnica map - reduce.

De exemplu:

```
input := []string{
{"caracatiță", "ceva", "saar", "aaastrfb", ""},
{"aaabbbccc", "caporal", "ddanube", "jahfjksgfjhs", "ajsdas", "urs"},
{"scoica", "coral", "arac", "karnak"}} se va afisa 1.33
```

11. Se dă un vector de vectori, care conțin mai multe stringuri. Să se afle numărul mediu de cuvinte care contin litere mici, mari, simboluri si numere (stringuri ce sunt parole puternice).

```
input := []string{
{"sadsa1@A", "cevaA!4", "saar", "aaastrfb", ""},
{"aaabbbccc", "!Caporal1", "ddanube", "jahfjksgfjhs", "ajsdas", "urs"},
{"scoica", "Coral!@12", "arac", "karnak"}} se va afisa 1.33
```

12. Se dă un vector de vectori, care conțin mai multe stringuri. Fiecare string reprezintă o cale către un fisier sau director în formatul Unix (cu /). Sa se afle numărul mediu de cai absolute și relative.

```
input := [][]string{
  {"/dev/null", "/bin", "saar", "teme/scoala/2020", ""},
  {"proiect/tema", "/dev", "ddanube", "jahfjksgfjhs", "ajsdas", "urs"},
  {"scoica", "/teme/repos/git", "arac", "karnak"}} se va afisa 1.33
```

13. Se dă un vector de vectori, care conțin mai multe stringuri. Fiecare string reprezintă nume de familie românești. Sa se afle numărul mediu de nume ce se termină în "escu".

```
input := [][]string{
  {"Popescu", "Ionescu", "Pop", "aastrfb", ""},
  {"Nicolae", "Dumitrescu", "ddanube", "jahfjksgfjhs", "ajsdas", "urs"},
  {"Dumitru", "Angelescu", "arac", "karnak"}} se va afisa 1.33
```

14. Se dă un vector de vectori, care conțin mai multe numere naturale. Sa se afle numărul mediu de numere din sirul lui Fibonacci.

```
input := [][]string{
  {"12", "5", "6", "13", "7"},
  {"21", "20", "42", "43", "8", "38"},
  {"54", "55", "34", "100"}} se va afisa 1.66
```

15. Se dă un vector de vectori, care conțin mai multe numere naturale. Sa se afle numărul mediu de numere ce au fix 3 biți de 1 în reprezentarea binară.

```
input := [][]string{
  {"1", "13", "6", "7", "9"},
  {"19", "20", "43", "43", "21", "53"},
  {"54", "55", "28", "101"}} se va afisa 1.66
```

Barem: 1.5p

1. Aplicație funcțională respectând codul de la laborator 0.7p
2. Explicarea conceptului de map-reduce 0.3p
3. Întrebări din cod: 0.5p
4. Deadline:
 - a. 16.01.2024 (Grupele 464, 462)
 - b. 17.01.2024 (Grupele 461, 463)

Penalizare la prezentare-săptămână 14-1p