

Tema 1 aplica?ie  
client-server

Cerin?e generale:

2. Metoda de procesare a datelor pe server trebuie s? fie f?cut? concurrent? -

utiliza?i rutine. 3. Exist? un fi?ier de configurare ?n care exist? parametrii ini?iali ai programului.

Ex: c?te elemente are array-ul de date pe care îl poate trimite clientul, de c?te ori se apelez? sau rutin? go. Decide?i voi ce anume pute?i include ?n acest fi?ier.

? Client <Nume> Conectat. ? Client <Nume> a facut request cu date: <date>. ? Server a primit requestul. ? Server proceseaza datele. ? Server trimitre <raspuns> c?tre client. ? Client <Nume> a primit r?spunsul: <raspuns>. Cerin?e individuale:

Serverul returneaz? c?tre client un array cu cuvinte unde cuvantul i din lista output este alc?uit din caracterele de pe pozi?ia i ?n fiecare din cuvintele din array-ul input.

Exemplu: casa, masa, trei, tanc, 4321 => cmitt4, aara3, ssen2, aaic1 Pentru pozitia 0 avem cmitt4 pentru ca sunt alese in ordinea caracterelor de pozitia 0 din fiecare string, deci c din casa, m din masa etc. 2. Clientul trimitre catre server un array de strings . Un string poate con?ine atât

caracter, c?t ?i cifre, amestecate. Serverul returneaz? c?tre num?rul clientului de numere care sunt p?trate perfecte. Exemplu: abd4g5, 1sdf6fd, fd2fdsf5 => 2 p?trate perfecte: 16 din 1sdf6fd, 25 din fd2fdsf5.

Serverul de r?spundere c?tre client cu suma numerelor format prin inversarea fiec?ruui element din array-ul ini?ial. Exemplu: 12, 13, 14 => 21, 31, 41 cu suma 93

4. Clientul trimite către server un array având pe primele două poziții limitate întregi a, b

ale unui interval și un set de valori întregi. Serverul returnează media aritmetică a numerelor citite, pentru care suma cifrelor aparțin intervalului [a,b]. Exemplu:  
Pentru a = 2, b = 10, n = 5 și valorile 11, 39, 32, 80, 84 server-ul va returna 41. 5.  
Clientul trimite către server un array de strings. .

Serverul elimina toate stringurile ce nu reprezintă numere binare, iar pentru cele ce reprezintă numere binare va face conversia în baza 10 și va returna clientului.

Exemplu: 2dasdas, 12, dasdas, 1010, 101 => 10, 3 6. Clientul trimite către server un array de strings. Un string poate conține doar

caractere. Serverul returnează client codificarea Caesar (  
[https://en.wikipedia.org/wiki/Caesar\\_cipher](https://en.wikipedia.org/wiki/Caesar_cipher)) a elementelor elementare, utilizând LEFT sau RIGHT shift cu k pozitioni. Exemplu: abcdef => xyzabc folosit LEFT shift cu 3 caractere. abcdef => bcdefg utilizând RIGHT shift cu 1 caracter. 7. Clientul trimite către server un text codificat după regulamentul următor: în față respectiv?

caracterul este scris un număr de numere consecutive ale acestuia. Serverul returnează textul decodificat. Numărul ce apare în față unui caracter va fi mai mic sau cel mult egal cu 20. Exemplu: Pentru ?1G11o1L? se va afișa ?Goooooooooooo L?. 8. Clientul trimite către server un array de numere naturale .

Serverul returnează numărul total de cifre ale tuturor numerelor prime din set.

Exemplu: Pentru: 23, 17, 15, 3, 18 => 5 cifre (nr 23, 17, 3) 9. Clientul trimite către server un array de strings. Un string poate conține atât

caractere, cât și cifre, amestecate. Serverul respondă către client cu numărul de cuvinte care au un număr de cuvinte aflate pe poziții par în cuvânt. Exemplu: mama, iris, bunica, ala => 2 cuvinte: iris, ala. 10. Clientul trimite către server un array de strings. Un string poate conține atât

caractere, cât și cifre, amestecate. Serverul returnează cmmdc pentru toate numerele (se calculează listele divizorilor pentru fiecare număr și se calculează intersecția listelor).

Exemplu: 24, 16, 8, aaa, bbb => 8.

Serverul respondă către client cu suma numerelor array-ului format prin permutarea la dreapta a k cifre ale fiecărui element din array-ul inițial.

Exemplu: 1234, 3456, 4567 => pentru k = 2:  $3412 + 5634 + 6745 = 15791 \cdot 12$ .

Clientul trimite către server un array de numere naturale .

Serverul returnează către client suma elementelor array-ului format din dublarea primei cifre a numerelor. Exemplu: 23, 43, 26, 74 => suma de returnat =  $223 + 443 + 226 + 774$ . Clientul trimite către server un array de întregi ce sunt pe primele două poziții limitate

înregi a, b ale unui interval, restul array-ului reprezentând componentele a n numere complexe (elementele din array luate două câte două imagini din partea reală și cea a unui număr complex). Server-ul returnează în ordine crescătoare valorile modulelor ce nu apară în intervalul [a, b]. - a se folosi pachetul cmplx (<https://golang.org/pkg/math/cmplx/#Abs>). Exemplu: 3,10,3,4,5,6: intervalul [3,10] și numerele complex:  $3+4i$ ,  $5+6i$ .

Serverul returnează către un array de stringuri cu toate stringurile care pot fi parole valide (contin litere mici, mari, cifre și simboluri). Puteti folosi regex.

Exemplu: Ceva12!@, asa212, dasdas => Ceva12!@ 15. Clientul trimite către server un array de caractere(pot să fie strings de

dimensiune 1) care pot să fie litere mici, mari, cifre sau simboluri . Serverul returnează către client un număr N de parole, N fiind aleator aleasă între 1 și 10. Fiecare cuvânt este de dimensiune M, unde M este între 5 și 15 și este alcătuit din caracterele primite de la client, având minim o literă mică, o literă mare, un simbol și o cifră. Exemplu: a, b, e, 3, !, A => b A3ba, ee3!Bab3, aabb A! Serverul întoarce 3 parole (3 aleator din intervalul 1-10) de dimensiune 5, 8, respectiv 6 (alese aleator din intervalul 5-15). Barem: 1.5p 1. Aplicări cu un singur client-server 0.2p 2. Mai multă client: 0.5p 3. Citire din fișier: 0.2p 4. Mesaje client-server: 0.2p 5. Arhitectura generală a aplicării: 0.4p 6. Termen de prezentare

o. 20.11.2025 (Grupa 1) b.

20.11.2025 (Grupa 2)