**Cryptocurrency**

In ecosistemul unei noi monede virtuale exista mai multe entități ce întrețin dinamica retelei ce implica creare de tranzactii, validarea tranzacțiilor si in final salvarea acestora. Astfel:

* **Utilizatorii** monedei virtuale **creeaza tranzacții** ce implica un transfer din *wallet*-ul personal, catre un alt *wallet*, si le adaugă într-o coada pentru a fi procesate.
* **Supervizorii** preiau din coada propuneri de tranzactii si fac o serie de validari: exista fonduri suficiente pentru a creea tranzacția ( fonduri ale utilizatorului)? wallet-ul destinatar este valid ? după care introduc tranzacția într-o nouă coada pentru a fi adaugata in blockchain (istoricul tranzacțiilor)
* **Minerii** preiau tranzacțiile verificate de către supervizori si **le adaugă** într-un istoric (ordonat cronologic) al tranzacțiilor.

**Scrieți un program prin care simulati acțiunile actorilor implicați știind ca acestia actioneaza concurent.**

Cozile implicate in operații sunt limitate la o capacitate **X**, astfel incat daca limita este atinsă nu se pot adăuga noi elemente pana cand nu se eliberează un loc.

O tranzactie propusa contine: **cod wallet utilizator (id thread), valoare tranzactie, cod wallet destinatar (id thread)**

Procesarea de către supervizor a unei tranzacții propuse presupune validarea menționate mai sus:

* Pentru a valida ca utilizatorul are destui bani pentru a onora valoarea din tranzactie, supervizorul trebuie sa parcurgă tot istoricul tranzacțiilor care este stocat intr-un fisier cu o structura ce se va detalia mai jos. Daca suma tuturor tranzactiilor istorice asociate wallet-ului utilizatorului este o valoare mai mare decat valoare tranzactiei propuse, atunci tranzacția este acceptata si se adaugă în următoarea coada pentru a fi procesata de mineri. În cazul în care valoarea este mai mica, tranzactia este respinsa.
* Pentru a valida daca codul wallet-ului destinatar este corect, se va face o verificare bazata pe numarul total de utilizatori definiți in program, acest număr oferind ca si informație lista tuturor codurilor disponibile (30 utilizatori, id-urile sunt [0, 29])

Tranzactia acceptata contine: **cod wallet utilizator (id thread), valoare tranzactie, cod wallet destinatar (id thread), id supervizor**

**Minerul** preia din coada tranzacțiile acceptate si simuleaza procesarea tranzactiei prin scrierea informațiilor primite intr-un fisier denumit **blockchain.txt.** Acest fișier va fi folosit de către **Supervizori** pentru a face validarea tranzacției prin parcurgerea întregului istoric și agregand toate tranzactiile asociate *wallet-*ului asociat utilizatorului ce a inițiat tranzacția.

Acțiunile utilizatorilor, supervizorilor, minerilor se afișează pe ecran:

“Tranzactia initiata de wallet X a fost preluata de supervizorul Y”

“Tranzactia initiata de wallet X cu valoarea Y catre wallet Z a fost acceptata/respinsa de supervizorul T, sold total inainte de operatie: W”

“Minerul X a salvat tranzactia initiata de wallet X cu valoarea Y catre wallet Z, validata de supervizorul T”

Programul se termina cand toate tranzacțiile inițializate sunt acceptate/refuzate și adaugate in **blockchain.txt**.

Consideram

* N utilizatori
* M supervizori
* 1 miner
* Dimensiunea cozilor: 20
* Fiecare wallet incepe cu 100 unitati (se pot adauga cate o tranzactie in blockchain.txt pentru fiecare utilizator pentru a simula acest lucru)

-1,100,1,-1

-1,100,2,-1

….

-1,100,N,-1

Caz de testare:

* N=50
* M=4
* Utilizatorii adaugă tranzacții (fiecare utilizator o singura tranzactie, dupa care isi termina executia) - tranzacțiile se genereaza cu valori random (cod wallet utilizator este id-ul thread-ului asociat utilizatorului, valoare tranzacții: random **1-200**, cod wallet destinatar: random 0-70).
* Dacă cozile sunt pline, fie utilizatorii, fie supervizorii așteaptă până când sunt notificati ca exista un loc disponibil..
* Supervizorii preiau din coada tranzacții, verifica dacă exista fonduri suficiente si transmit tranzacția către procesarea in blockchain prin adaugarea ei coada corespunzătoare. La sfarsit notifică faptul ca tranzacția a fost procesata si există un loc disponibil in coada destinata tranzacțiilor create de utilizatori.
* Minerul adauga tranzactia acceptata in **blockchain.txt** si notifică faptul ca exista un loc disponibil in coada dedicată tranzacțiilor acceptate.

**Observatie:** pentru așteptare condiționată este necesar sa se folosească un mecanism de tip “wait-notify” (busy-waiting nu este acceptabil).