Tema 2 – Structuri de date (seria CB)
Turnurile din Hanoi

Responsabili temă:	Cosmin-Dumitru Oprea, Florin Iancu, Costin Busioc
Data publicării:	**4.04.2018**
Termenul de predare:	**24.04.2018 ora 23:55** Se acceptă teme trimise cu penalizare de 10 puncte / zi (din maxim 100 puncte) până la data de 27.04.2018 ora 23:55

1. Introducere

Turnurile din Hanoi este un joc matematic sau puzzle. Este format din trei tije și un număr variabil de discuri, de diferite mărimi, care pot fi poziționate pe oricare din cele 3 tije. Jocul începe având discurile așezate în stivă pe prima tijă, în ordinea descrescătoare a mărimii lor de la bază spre vârf (astfel încât să formeze un turn). Scopul jocului este acela de a muta întreaga stivă de pe tija inițială pe o tijă finală, respectând următoarele reguli:

- La un moment dat se poate muta doar un singur disc.
- Fiecare mutare constă în luarea celui mai de sus disc de pe o tijă și glisarea lui pe o altă tijă, chiar și deasupra altor discuri, dacă ele care sunt deja prezente pe acea tijă.
- Un disc mai mare **nu** poate fi poziționat deasupra unui disc mai mic.

Numărul minim de mișcări pentru a rezolva jocul este $\mathbf{2^n - 1}$, unde n este numărul de discuri.

O simulare a jocului se poate vedea în următorul link:
https://www.puzzle.ro/ro/play_toh.htm

2. Cerință

Se cere rezolvarea unui astfel de joc folosind numărul minim de mutări, adică $\mathbf{2^n - 1}$.
Observație. Mutările pentru rezolvarea jocului în număr minim de pași sunt deterministe. Există o singură soluție de rezolvare în număr minim de mutări pentru fiecare sistem Hanoi.

3. Descrierea operațiilor și a datelor de intrare

Implementarea temei va consta în efectuarea unui set de operații descrise în fișierul de intrare.
Definim un **sistem Hanoi** ca fiind un grup de 3 stive: A, B, C.
Scopul jocului este mutarea tuturor discurilor de pe tija A pe tija C.
Un disc Hanoi este descris prin două proprietăți: diametru și culoare.
În temă vor fi mai multe sisteme Hanoi. În cadrul unuia, toate discurile au aceeași culoare.

Pe prima linie din fișierul de intrare se află un număr **N** - numărul de operații. Urmează apoi **N** linii, fiecare conținând câte o operație, după cum urmează:

add [culoare_disc] [dimensiune_disc]

- culoarea unui disc este un șir de caractere `a..z` (litere mici ale alfabetului englezesc)
- toate discurile de aceeași culoare vor fi adăugate în stiva A a sistemului Hanoi în ordine descrescătoare a diametrului, de sus în jos.
- după inserarea oricărui disc într-o stivă aceasta trebuie să rămână ordonată
-

play [culoare] [număr_mutări]

- operația cere simularea următoarelor **[număr_mutări]** mutări ale jocului în sistemul Hanoi de culoare **[culoare]**
- dacă **[număr_mutări]** este mai mare decât numărul de pași necesari pentru terminarea jocului, simularea se va opri și nu se vor mai face alte mutări.

show [culoare]

- operația cere afișarea stării curente a sistemului Hanoi de culoare **[culoare]**
- afișarea se va face pe 3 linii astfel:

```
A_[culoare]: diametru_disc_1 diametru_disc_2 … diametru_disc_n
B_[culoare]: diametru_disc_1 diametru_disc_2 … diametru_disc_n
C_[culoare]: diametru_disc_1 diametru_disc_2 … diametru_disc_n
```

Observație: `disc_1` este cel de la baza stivei, iar discul n este cel din vârful stivei.

show_moves [culoare] [număr_mutări]

- operația cere afișarea următoarelor **[număr_mutări]** mutări ce trebuie executate în sistemul Hanoi de culoare **[culoare]** pentru a termina jocul. Dacă numărul total de mutări existente în sistem este mai mic decât **[număr_mutări]** se vor afișa doar cele existente.
- afișarea se face pe o singură linie. Un exemplu de afișare este mai jos:

```
M_[culoare]: A->B A->C B->C
```

! Urmăriți cu atenție exemplele de mai jos și explicația detaliată de la testul 3.

4. Restricții și precizări:

- 1 <= N <= 100
- Numărul de discuri de aceeași culoare <= 16
- o operație de tip **play** NU va fi urmată de o operație de tip **add** pentru același sistem Hanoi
- datele de intrare nu trebuiesc validate; se garantează că vor fi corecte
- programul va fi rulat astfel: **./tema2 in_file out_file**
- comenzile se citesc din fișierul **in_file**, iar rezultatele se scriu în fișierul **out_file**
- stivele și cozile vor fi implementate ca liste **generice** simplu înlănțuite**.**
- **nu** aveți voie cu variabile globale
- **pentru sortări, afișări** se vor folosi doar stive/cozi auxiliare (**nu se permite iterarea prin stive/cozi**)

● **Sugestie:** un mod util de rezolvare a cerinței, este pornirea cu o simulare inițială a întregului set de pași pentru fiecare culoare și păstrarea mutărilor într-o coadă.

● **20%** din testele de intrare vor respecta următoarele condiții:
  - vor conține doar operații de tip **add, show**
  - discurile de aceeași culoare vor fi în ordine descrescătoare a diametrului
● alte **20%** din testele de intrare vor respecta următoarele condiții:
  - vor conține doar operații de tip **add, show**

5. Exemplu

Intrare	Ieșire
8 add red 3 add blue 4 add red 2 add blue 2 add red 1 add blue 1 show red show blue	A_red: 3 2 1 B_red: C_red: A_blue: 4 2 1 B_blue: C_blue:
8 add red 2 add red 1 add blue 2 add red 3 add blue 4 add blue 1 show red show blue	A_red: 3 2 1 B_red: C_red: A_blue: 4 2 1 B_blue: C_blue:
7 add red 2 add red 1 add red 3 show red play red 3 show_moves red 4 play red 4	A_red: 3 2 1 B_red: C_red: M_red: A->C B->A B->C A->C

Explicație: Se construiește o tijă cu 3 discuri, apoi se simulează toate cele $7(2^n - 1)$ mutări aferente stivelor de culoare roșie, care vor fi păstrate într-o coadă. La prima operație **play** se vor executa primele 3 mutări din coadă(evident primele 3 mutări se vor elimina din coadă). La operația **show_moves** se vor afișa următoarele 4 mutări din coadă, fără a fi eliminate. La următoarea operație **play** se vor executa ultimele 4 mutări din coadă și jocul se va termina (nemaifiind alte mutări în coadă).

18 add red 3 add blue 4 add red 2 add blue 2 add red 1 add blue 1 show red show blue show_moves red 3 show_moves blue 4 play red 3 show_moves red 4 play red 4 show_moves red 3 play blue 6 show_moves blue 1 play blue 2 show_moves blue 2	A_red: 3 2 1 B_red: C_red: A_blue: 4 2 1 B_blue: C_blue: M_red: A->C A->B C->B M_blue: A->C A->B C->B A->C M_red: A->C B->A B->C A->C M_red: M_blue: A->C M_blue:

6. Notare

- **85 de puncte** obținute pe testele de pe vmchecker
- **10 puncte**: coding style; se pot acordă depunctări pentru:
  - warninguri la compilare (trebuie compilat cu -Wall)
  - linii mai lungi de 80 de caractere
  - folosirea incorectă de pointeri
  - neverificarea codurilor de eroare
  - neeliberarea resurselor folosite (trebuie eliberarea memoriei alocate, închiderea fișierelor etc)
  - alte situații nespecificate aici, dar considerate inadecvate
- **5 puncte**: README - va conține detaliile de implementare a temei, precum și **punctajul obținut la teste** (la rularea pe calculatorul propriu)
- **bonus: 20 de puncte** pentru soluțiile care nu au memory leak-uri (bonusul se acordă dacă testul a trecut cu succes)
- temele care nu compilează, nu rulează sau obțin punctaj 0 pe vmchecker, indiferent de motive, vor primi punctaj 0.

7. Reguli de trimitere a temelor

- temele vor trebui încărcate atât pe vmchecker (în secțiunea Structuri de Date **seria CB**) cât și pe cs.curs.pub.ro, în secțiunea aferentă temei 2.

- arhiva cu rezolvarea temei trebuie să fie **.zip** și să conțină:

- fișiere surse (fiecare fișier sursă va trebui sa înceapă cu un comentariu de forma: **/* NUME Prenume - grupa */**
- fișier **`README`**, denumit obligatoriu astfel, care să conțină detalii despre implementarea temei
- fișier **`Makefile`**, denumit obligatoriu astfel, cu două reguli:
  build, care va compila sursele și va obține executabilul cu numele **tema2**
  clean, care va șterge executabilele și alte fișiere obiect generate
- arhiva trebuie să conțină doar fișierele sursa (inclusiv Makefile si README), **nu** se acceptă fișiere executabile sau obiect
- dacă arhiva nu respectă specificațiile de mai sus nu va fi acceptată la upload și tema nu va fi luata în considerare.