Tema 1 - TetriBit

Responsabili

- Andrei Lecu
- Ovidiu Dancila
- Data publicare: **15 Octombrie 2018**
- Deadline: **5 Noiembrie 2018, ora 23:55**

Update:

- *27 Octombrie 2018*: A fost actualizat deadline-ul temei (Post Forum). **Atentie!** Nu se mai accepta teme trimise dupa data de **5 Noiembrie 2018**, ora **23:55**.

Obiective

- să se realizeze un program urmând anumite cerințe
- să se respecte formate stricte de intrare/ieşire
- să se însușească cunoștințele din primele trei laboratoare
- să înțeleagă și să utilizeze operatorii pe biți

Introducere

Gigel nu este inițiat încă în tainele programării, dar vrea să învețe și să devină expert. Într-o zi i-a venit o idee care este foarte posibil să îi schimbe viața: lui îi plac foarte mult jocurile video și ar putea învăța foarte ușor programare îmbinând utilul cu plăcutul. Astfel, s-a gândit să implementeze cunoscutul joc Tetris folosind cunoștințele dobândite în primele 3 laboratore de la cursul de Programarea Calculatoarelor (cursul lui preferat de până acum).

Cerință

Harta

După ce a învățat în cadrul laboratorului operatorii pe biți, Gigel s-a gândit că se poate folosi de reprezentarea în baza 2 a unui număr pentru a desena harta jocului Tetris astfel:

- Biții 1 sunt zone ocupate
- Biții 0 sunt zone libere

De asemenea, pentru a avea o harta mai mare la dispoziție, a ales să folosească numere de 64 de biți (de exemplu: **unsigned long long** sau **uint64_t**) dimensiunea hărții fiind de 8 linii a câte 8 coloane (8 x 8). În plus, pentru o afișare mai frumoasă a decis să înlocuiască în afișare biții de **1** folosind caracterul **”#“** și biții de **0** folosind caracterul **”.”**(punct).
Astfel, pentru numărul **“284 803 830 071 167”** a cărui reprezentare în bază 2 este:

```
00000000 00000001 00000011 00000111 00001111 00011111 00111111 01111111
```

Se poate crea harta:

```
........

.......#

......##

.....###

....####

...#####

..######

.#######
```

OBSERVAȚIE: În configurația inițială a hărții nu pot exista linii complete, adică nu pot apărea linii de forma:

```
########
```

în care toți cei 8 biți sunt setați pe **1**.

Implementare

Gigel s-a gândit să își organizeze implementarea prin stabilirea de la început a tuturor funcționalităților pe care jocul lui este capabil să le realizeze.

Afișare hartă

Primul pas pe care acesta trebuie să îl efectueze este să afișeze harta primită la intrare.

Piese

La fel ca harta, piesele sunt primite ca input sub forma unui număr decimal.

Tipuri de piese:

```
1. ...#....

2.             |    ...#....

3. ...##...    |    ...#....
```

```
4. ...#....    |    ...#....

5. ..##....    |    ...##...

6. ...##...

7. ...##...
```

Piesele pot apărea în orice poziție și pot să fie rotite în orice fel. De exemplu, piesele de tipul 3 pot să apară și sub forma:

```
.##.....

.#......
```

Mișcarea pieselor

Fiecare sesiune de joc conține un număr de mutări (numărul de mutări poate să fie și 0). Fiecare mutare conține piesa curentă și transformările care trebuie aplicate acesteia - câte o transformare pentru fiecare linie a hărții. Astfel, o mutare apare în input cu următorul format:

```
6168 -1 2 -2 0 0 1 1 -3
```

Unde primul număr reprezintă codificarea în decimal a piesei și următoarele 8 numere sunt transformările care trebuie aplicate asupra piesei. Numărul specific fiecărei transformări reprezintă numărul de poziții cu care piesa trebuie deplasată:

- Un număr negativ sugerează o deplasare la stânga
- Un număr pozitiv sugerează o deplasare la dreapta

OBSERVAȚIE 1: Numărul specific fiecărei transformări poate să fie mai mare decât numărul maxim de deplasări posibile ale piesei. În acest caz, trebuie deplasată piesa doar cu numărul maxim de mutări posibile. De exemplu, pentru piesa **6168**, care arată asftel:

```
...##...

...##...
```

o transformare cu **-6** produce rezultatul:

```
##......

##......
```

pentru că numărul maxim posibil de deplasări în stânga este 3.

De asemenea, pentru piesa:

```
....#...
```

O transformare cu **-9**, ținând cont că linia pe care trebuie să se deplaseze arată astfel:

```
#.#.....
```

Va produce următorul rezultat pe linia mai sus menționată:

```
#.#.#... -> #.##....
```

Deci piesa nu poate să treacă peste o zonă deja ocupată, ci trebuie să rămână în ultimul loc disponibil (în cazul curent piesa efecutează o singură deplasare la stanga).

OBSERVAȚIE 2: Numărul de transformări este întotdeauna egal cu 8. Dacă, din motivul unei coliziuni, nu se pot efectua toate transformările, atunci celelalte transformări trebuie ignorate.

OBSERVAȚIE 3: Pentru consecvență, se alege ca ordinea operațiilor să fie următoarea:

1. Pasul 1: se face mutarea piesei pe linia de jos.
2. Pasul 2: se face deplasarea piesei în cadrul liniei curente.

Coliziune

În cazul în care, în drumul spre ultima linie, o piesă întâlnește un obstacol, aceasta trebuie să nu mai execute următoarele transformări și să se oprească în poziția rămasă.

De exemplu, pentru harta:

```
........

........

........

........

#####...

#####...

#####...

#####...
```

și piesa:

```
....##..

....##..
```

cu mutarea:

```
3084 0 0 0 0 2 0 0 0 0
```

execuția mutării se va opri dupa cea de-a 4-a mutare (a 5-a mutare nu poate fi efectuată din cauză că apare o coliziune și din cauza ordinii operațiilor de mutare a piesei), harta finală fiind:

```
........

........

....##..

....##..

#####...

#####...

#####...

#####...
```

Eliminarea liniilor complete

În cazul în care, după o mutare care își termină execuția (ajunge la ultima linie sau se oprește din cauza unei coliziuni), o linie de pe hartă devine completă (toți cei 8 biți sunt 1), atunci acea linie trebuie eliminată si toate liniile de deasupra coborâte cu o poziție. De asemenea, dacă mai multe linii devin complete după o mutare, toate trebuie eliminate.

De exemplu, pentru harta:

```
........

........

##...#..

#######.

#######.

#####.##

#####.##

#####.##
```

și piesa:

```
#.......

#.......
```

cu mutarea:

```
32896 0 10 0 0 0 0 0 0
```

secvența de afișări este:

```
........

........

##...#..

#######.

#######.

#####.##

#####.##

#####.##


#.......

........

##...#..

#######.

#######.

#####.##

#####.##

#####.##


.......#

.......#

##...#..
```

```
#######.
#######.
#####.##
#####.##
#####.##

........
.......#
##...#.#
#######.
#######.
#####.##
#####.##
#####.##

........
........
##...#.#
########
#######.
#####.##
#####.##
#####.##

........
........
##...#..
########
```

```
########

#####.##

#####.##

#####.##


........

........

........

........

##...#..

#####.##

#####.##

#####.##


GAME OVER!

Score:7.89
```

Terminarea jocului

Un joc poate să se încheie în 2 cazuri:

1. Toate mutările au fost efectuate
2. Piesa nu are loc în întregime pe hartă. Adică apare o coliziune înainte ca aceasta să fie afișată complet pe hartă.

De exemplu: Pentru harta:

```
........

#..#####

####..##

#.######

##.#####
```

```
####.###
#.######
####.###
```

și piesa

```
...##...
...##...
```

cu mutarea

```
6168 1 1 1 1 1 1 1 1
```

Secventa de rulare este:

```
........
#..#####
####..##
#.######
##.#####
####.###
#.######
####.###


....##..
#..#####
####..##
#.######
##.#####
####.###
#.######
####.###
```

```
....##..

#..#####

####..##

#.######

##.#####

####.###

#.######

####.###


GAME OVER!

Score:4.87
```

OBSERVAȚIE: În cazul în care piesa nu are loc în întregime pe hartă și jocul trebuie încheiat, afișarea variantei finale a hărții se face de 2 ori (după cum se observă și în exemplul de mai sus).

Scor

Pentru că un joc de tetris fără scor nu are farmec, Gigel s-a gandit să inventeze o formula de calcul a scorului pentru harta finala. Astfel, după terminarea mutărilor, el calculează scorul folosind următoarea formulă:

$$score = zerosNumber - \sqrt{+1.25_{completedLines}})score = zerosNumber + 1.25^{completed Lines})$$

Unde:

- zerosNumber = numarul de biți care au valoarea 0 din harta finala
- completedLines = numărul de linii complete care au fost eliminate

De asemenea, pentru a evita rezultatele lungi, el s-a hotarât să afișeze scorul cu o precizie de 2 zecimale (rotunjit).

Format input

```
H

M
```

```
P1 T1_1 T1_2 T1_3 T1_4 T1_5 T1_6 T1_7 T1_8

P2 T2_1 T2_2 T2_3 T2_4 T2_5 T2_6 T2_7 T2_8

...

PM TM_1 TM_2 TM_3 TM_4 TM_5 TM_6 TM_7 TM_8
```

unde:

- Prima linie conține numărul H în decimal (pe 64 de biți) care reprezintă codificarea hărții
- A doua linie conține numărul M de mutări ce trebuie efectuate
- Următoarele M linii conțin detaliile mutărilor: Piesa și cele 8 numere specifice transformărilor care trebuie aplicate piesei în cei (maxim) 8 pași de rulare

Citirea se va face de la stdin (tastatură), dar, pentru a facilita testarea, se recomandă scrierea input-ului într-un fișier și rularea executabilului folosind redirectarea din fișierul creat. Exemplu:

```
./tema1 <in/test0.in
```

Format output

- Primul pas este afișarea hărții primite ca input.
- Pentru fiecare mutare se va afișa harta după fiecare transformare. În cazul în care ultima transformare posibilă (nu neapărat cea de-a 8-a) produce o operație de eliminare de linii complete, atunci trebuie efectuată afișarea atât înainte de eliminarea liniilor, cât și după.
- Între 2 afișări de hărți trebuie să se lase o linie liberă (\n)
- După terminarea tuturor mutărilor, se afișează următorul text (pe 2 linii separate):

```
GAME OVER!

Score:S
```

unde S este scorul calculat folosind formula de mai sus.

Pentru că testarea corectitudinii temei se face automat, este necesar să se respecte cu strictețe formatul de output. **NU** trebuie să existe afișări în plus.

Punctaj

- **[20p]** Teste care nu conțin nicio mutare - doar afișare de hartă și calcul de scor.
- **[15p]** Mutări simple de piese, fără coliziuni.

- **[10p]** Mutări care implica și coliziuni.
- **[10p]** Transformări ale pieselor care depășesc limitele rândului.
- **[10p]** Mutări din care rezultă completarea unor linii și efectuarea eliminării acestora.
- **[25p]** Teste combinate și mai complexe.
- **[10p]** Fișier README în care să se descrie implementarea + Coding Style.

TOTAL: **100p**

Trimitere temă

Tema va fi trimisă folosind vmchecker, cursul **Programarea Calculatoarelor (CB & CD)**. Găsiți checker-ul aici.
Formatul arhivei va fi următorul:

1. fișier(ele) `.c` (și fișiere `.h` - dacă este cazul).
2. Un fișier **Makefile** (detalii aici) care să conțină următoarele reguli:
   a. **build**: creează executabilul aferent (numele executabilului: **tema1**)
   b. **run**: rulează executabilul aferent
   c. **clean**: șterge fișierele obiect/executabile create.
3. Un fișier README în care vă descrieți rezolvarea temei.

Este necesară scrierea numelui si a grupei atât în fișierul README, cât și în toate fișierele sursă pe care le adăugați în arhiva temei.

1. Arhiva trebuie să fie de tipul **zip**.
2. Inputul se va fi citit de la **stdin (tastatura)**, iar output-ul va fi afișat la **stdout (ecran)**. Testarea se face cu ajutorul redirectării acestora din linia de comandă/bash.

Observații

- Nu folosiți variabile globale.
- Fiți consistenți în ceea ce privește Coding Style-ul.
- Puteți folosi noțiuni care nu au fost prezentate în primele 3 laboratoare (vectori, funcții, etc.) cu condiția să menționați în fișierul README utilitatea acestora și motivul pentru care le-ați folosit.
- Pentru întrebări și nelămuriri, creați un nou thread pe forum-ul asociat temei: Forum Tema 1

Listă depunctări

- o temă care nu compilează și nu a rulat pe **vmchecker** nu va fi luată în considerare
- o temă care nu rezolvă cerința și trece testele prin alte mijloace nu va fi luată în considerare

- o temă în care memorarea hărții se face folosind tablouri (uni-dimensionale sau bi-dimensionale) și/ sau nu folosește operatori pe biți pentru realizarea cerințelor nu va fi luată în considerare.
- [-1.0]: warning-uri la compilare (este obligatorie folosirea în fișierul **Makefile** a flag-ului de compilare **-Wall**pentru regula **build**)
- [-1.0]: numele variabilelor nu sunt sugestive
- [-1.0]: linii mai lungi de 80 de caractere
- [-5.0]: abordare ineficientă
  - în cadrul cursului de programare nu avem ca obiectiv rezolvarea în cel mai eficient mod posibil a programelor; totuși, ne dorim ca abordarea să nu fie una ineficientă, de genul să nu folosiți instrucțiuni repetitive acolo unde clar era cazul, etc.