



Fillit

Can you feel it ?

Team pedago pedago@staff.42.fr

*Résumé: este istoria unei piese de **Tetris**, a unui mic patrat si a unui programator ce intra intr-un bar...*

Table des matières

I	Preambul	2
II	Introducere	3
III	Obiective	4
IV	Instrtuctiuni generale	5
V	Parte obligatorie	6
V.1	Intrarea in program	6
V.2	Cel mai mic patrat	9
V.3	Rezultatul programului	10
VI	Livrare si corectare-perechi	12

Chapitre I

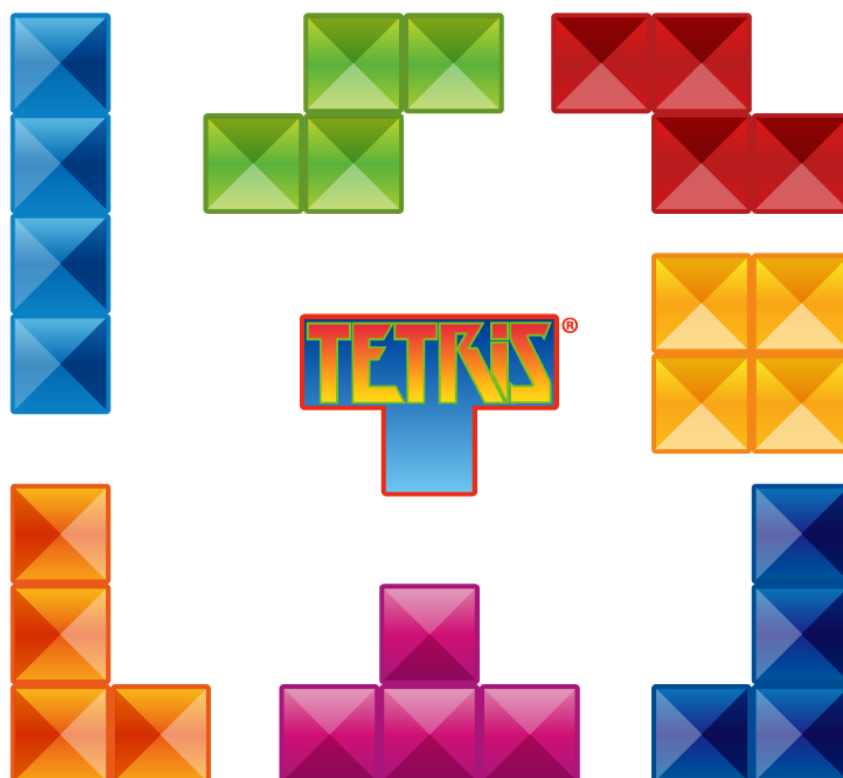
Preambul

Chapitre II

Introducere

Fillit este un proiect ce va permite sa descoperiti si/sau sa va familiarizati cu o problematica recurenta din programare : cautarea unei solutii optime dintr-un numar foarte mare de posibilitati rezonabile. In cazul acestui proiect e vorba de aranjarea Tetriminos intre ele si determinarea celui mai mic patrat posibil ce le poate contine.

Un Tetriminos este o figura geometrica formata din 4 blocuri pe care le cunoasteti datorita celebrului joc Tetris.



Chapitre III

Obiective

Scopul **Fillit** nu este de a rescrie jocul **Tetris**, dar este o varianta a jocului in aceeași logica. Programul va lua ca parametru un fisier ce descrie o lista de **Tetriminos** ce vor trebui apoi aranjate între ele pentru a forma cel mai mic patrat posibil. Bineinteles, scopul este de a gasi cel mai mic patrat, in timpul cel mai scurt, in ciuda unui numar mare de aranjamente ce crește într-un mod exploziv cu fiecare piesa suplimentara.

Va trebui deci sa reflectati asupra structurii datelor si a algoritmului folosit, astfel ca rulara programului sa se termine inaintea anului 3000.



Chapitre IV

Instruțiuni generale

- Proiectul trebuie scris în limbajul C și conform standardului de cod (Norme).
- Funcțiile permise sunt : `exit`, `open`, `close`, `write`, `read`, `malloc` și `free`.
- Makefile va compila fișierele livrate și va avea cel puțin comenzile : `all`, `clean`, `fclean` și `re`.
- Trebuie să compilați executabilul folosind flag-urile `Wall`, `Wextra` și `Werror`. Orice alt flag și mai ales optimizarea sunt interzise.
- Executabilul trebuie să se numească `fillit` și să fie localizat în directorul rădăcină.

Chapitre V

Parte obligatorie

V.1 Intrarea in program

Executabilul trebuie sa ia ca parametru un (singur) fisier ce descrie lista de **Tetriminos** de aranjat. Acest fisier este format foarte exact : fiecare descriere a unui **Tetriminos** este realizat pe 4 linii iar doua **Tetriminos** sunt **separate de o linie goala**.

Daca numarul parametrilor executabilului este diferit de 1, programul trebuie sa afiseze **usage** si sa se opreasca in mod corespunzator. Daca nu stiti ce este un **usage**, lansati comanda **cp** fara argument in shell pentru a va face o idee.

Descrierea unei **Tetriminos** trebuie sa respecte urmatoarele reguli :

- Exact 4 linii a 4 caractere urmate de un retur de linie.
- O **Tetriminos** este o piesa clasica de **Tetris** compusa din 4 blocuri.
- Fiecare caracter trebuie sa fie un diez '#' cand caseta corespunde unuia din cele 4 blocuri ale unei **Tetriminos**, fie un punct '.' cand caseta e goala.
- Fiecare bloc al unei **Tetriminos** trebuie sa fie in contact cu cel putin cu un alt bloc, pe una din cele 4 laturi ale sale.

Cateva exemple de descriere de **Tetriminos** valide :

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

Cateva exemple de descriere de **Tetriminos** gresite :

```
####    ..#    ##...    #.    ....    ..##    ####    ,,,    .HH.
..#     ..#    ##...    ##    ....    ....    ####    ####    HH..
....    .#..    ....    #.    ....    ....    ####    ,,,    ....
....    #...    ....    ....    ....    ##..    ####    ,,,    ....
```

Fiecare **Tetriminos** ce ocupa doar 4 casete din cele 16 disponibile este posibil sa descrie acelasi **Tetriminos** de fiecare data in mod diferit. Totusi, rotatia unei **Tetriminos** descrie o **Tetriminos** diferita de original in cadrul acestui proiect. Asta inseamna ca orice rotatie a unei **Tetriminos** nu e posibila decat una fata de altele.

Aceste **Tetriminos** sunt perfect echivalente din toate punctele de vedere :

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

Aceste 5 **Tetriminos** sunt 5 **Tetriminos** perfect diferite din toate punctele de vedere :

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

Pentru a termina, iata un exemplu de fisier descriptiv valid pe care programul vostru va trebui sa-l accepte spre rezolvare,

```
$> cat -e valid_sample.fillit
...#$
...#$
...#$
...#$
$
...$
...$
...$
...$
####$
$
.###$
...#$
...$
...$
...$
$
...$
.###$
.##$.
...$
...$
$>
```


De asemenea un exemplu de fisier descriptiv gresit pe care programul vostru trebuie sa-l respinga din mai multe motive :

```
$> cat -e invalid_sample.filllit
```

[illegible]

V.2 Cel mai mic patrat

Scopul acestu proiect este de a potrivi **Tetriminos** intre ele pentru a forma cel mai mic patrat posibil, stiind ca patratele pot avea gauri atunci cand piesele date nu se potrivesc perfect.

Fiecare **Tetriminos**, desi e reprezentat pe o grila de 16 carouri, e definita doar pe spatiile pline (de '#'). Cele 12 '.' ramase vor fi ignorate de aranjarea **Tetriminos** intre ele.

Tetriminos sunt aranjate in ordinea in care apar in fisier. Dintre diferitele solutii posibile pentru realizarea celui mai mic patrat, va fi retinuta solutia in care fiecare **Tetriminos** este dispusa cel mai sus, iar apoi cel mai la stanga posibil, in momentul plasarii acesteia.

Exemplu :

Considerand doua **Tetriminos** consecutive (cele '#' vor fi inlocuite cu cifre pentru a simplifica citirea rezultatelor) :

```
1...  ....
1...  ....
1... ET ..22
1...  ..22
```

Cel mai mic patrat e format din aceste 2 piese facute din 4 carouri, dar exista mai multe versiuni pe care le puteti vedea mai jos :

a)	b)	c)	d)	e)	f)
122.	1.22	1...	1...	1...	1...
122.	1.22	122.	1.22	1...	1...
1...	1...	122.	1.22	122.	1.22
1...	1...	1...	1...	122.	1.22
g)	h)	i)	j)	k)	l)
.122	.1..	.1..	221.	..1.	..1.
.122	.122	.1..	221.	221.	..1.
.1..	.122	.122	..1.	221.	221.
.1..	.1..	.122	..1.	..1.	221.
m)	n)	o)	p)	q)	r)
22.1	.221	...1	...1	...1	...1
22.1	.221	22.1	.221	...1	...1
...1	...1	22.1	.221	22.1	.221
...1	...1	...1	...1	22.1	.221

Conform regulilor de joc, solutia corecta este a).

V.3 Rezultatul programului

Programul trebuie sa afiseze cel mai mic patrat la iesirea standard. Pentru a identifica fiecare **Tetriminos** in patratul solutie, veti atribui o litera mare (incepand cu 'A') acestei **Tetriminos** in ordinea in care ele apar in fisierul descriptiv. Nu vor fi niciodata mai mult de 26 **Tetriminos** intr-un fisier descriptiv.

Daca la parcurgerea fisierul descriptiv apare cel putin o eroare, programul trebuie sa afiseze **error** la iesirea standard si sa se opreasca in mod corespunzator.

Exemplu :

```
$> cat sample.fillit | cat -e
....$
##..$
.#..$
.#..$
$
....$
####$
....$
....$
$
#...$
###.$
....$
....$
$
....$
##..$
.##.$
....$
$> ./fillit sample.fillit | cat -e
DDAA$
CDDA$
CCCA$
BBBB$
$>
```

Alt exemplu :

```
$> cat sample.fillit | cat -e
....$
....$
####$
....$
$
....$
..$.
.##$
.##$
$> ./fillit sample.fillit | cat -e
error$
$>
```

Ultimul exemplu :

```
$> cat sample.fillit | cat -e
...#$
...#$
...#$
...#$
...#$
$
....$
....$
....$
####$
$
.###$
...#$
....$
....$
$
....$
..##$
.##.$
....$
$
....$
.##.$
.##.$
....$
$
....$
....$
##.$.
.##.$
$
##.$.
.#.$.
.#.$.
....$
$
....$
###.$
.#.$.
....$
$> ./fillit sample.fillit | cat -e
ABBBB.$
ACCCEE$
AFFCEE$
A.FFGG$
HHHDDG$
.HDD.G$
$>
```

Chapitre VI

Livrare si corectare-perechi

Livrati proiectul in repository-ul **GiT** ca de obicei. Doar rezultatul muncii voastre transmis acolo va fi evaluat.

Activitatea va fi evaluata de o moulinette (si in plus de o corectare/evaluare-pereche conform cursului). Aceasta moulinette va avea un timeout arbitrar ce va opri executia programului daca acesta foloseste prea mult timp pentru a gasi solutia unui test. Acest test va fi considerat gresit in acest caz.