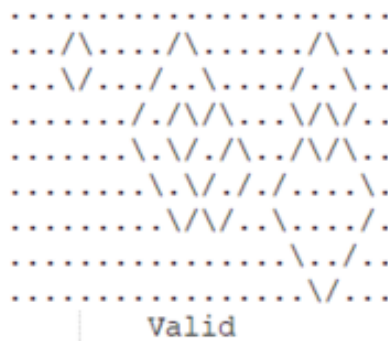


1. Se dă un fișier `diamonds.in` care conține pe prima linie două numere întregi, n și m ($1 \leq n \leq 50$; $1 \leq m \leq 50$), despărțite de un spațiu, reprezentând lungimea și lățimea secțiunii în care se caută diamante. Pe fiecare linie, se vor găsi m caractere. Singurele caractere posibile sunt '.'(punct), '/'(slash) sau '\' backslash. Scrieti un program care inlocuieste cu '.' toate slash-urile și backslash-urile care nu formează un diamant valid. Rezultatul se va scrie într-un fișier `diamonds.out`.

Observatii:

- Diamantele sunt formate din slash-uri și backslash-uri ('/' și '\') așezate într-o configurație simetrică pe verticala și orizontală (dublu simetric);
- Diamantele se pot intersecta, suprapune și pot conține alte diamante în interior;
- Diamantele conțin minimum două slash-uri '/' și două backslash-uri '\' ;
- Numărul slash-urilor este egal cu cel al backslash-urilor.

Exemplu:



<code>diamonds.in</code>	<code>diamonds.out</code>
6 8	.../\...
/./\.\	../...\
././\.\	././.\.
././\.\	.\./.\.
.\.\.\.	..\.\.\.
.\.\.\.	...\./...

2. Scrieți un program care identifică expresiile repetitive (“ticurile verbale”) întâlnite în fișierul text `verbal_tics.in`.

Observații:

- se vor considera expresii repetitive doar acelea care conțin cel puțin 2 cuvinte fără niciun cuvânt de legătură, respectiv 3 sau mai multe cuvinte dintre care cel puțin unul nu e de legătură;
- se consideră un cuvânt de legătură dacă are mai puțin de 3 litere;
- lungimea maximă a unei linii de text este de 512 caractere;
- `verbal_tics.in` poate conține mai multe “ticuri verbale”;
- `verbal_tics.out` trebuie să conțină toate “ticurile verbale”, fiecare pe câte un rând nou și în ordinea apariției lor

<pre>verbal_tics.in Si te-ai dus, dulce minune, S-a murit iubirea noastra - Floare albastra ! Floare albastra !... Totusi este trist in lume! Fara de iubirea noastra .</pre>	<pre>verbal_tics.out iubirea noastra Floare albastra</pre>
---	--

3. Reprezentați grafic în mod text un arbore definit de structura de mai jos. Fiecare linie din fișierul `input.in` respectă formatul: `<nod>;<nod parinte>`. Rădăcina arborelui este definită în prima linie a fișierului `input.in`. Nodurile pot avea orice nume de tip text.

Observații:

- arborele final conține pe fiecare nivel nodurile sortate alfabetic
- fiecare nod este afișat pe o linie nouă precedat de “|__” (un caracter pipe și 3 caractere underscore)
- calea completă către numele nodului va conține un număr de grupuri de caractere ce reflectă:
 - nivelul nodului în arbore
 - așezarea nodului de tip părinte în arbore (legătura nodului cu nodurile părinți)
- grupurile de caractere sunt de tipul “|...” (un caracter pipe și 3 caractere punct) sau “....” (4 caractere punct).

input.in	output.out
Romania;root	__ Romania
Bihor;Romania __ Bihor
Oradea;Bihor __ Oradea
Salonta;Bihor __ Salonta
Campiei;Salonta __ Campiei
Pietei;Salonta __ Pietei
Principala;Salonta __ Principala
Galati;Romania __ Galati
Hunedoara;Romania __ Hunedoara
Deva;Hunedoara __ Deva
Hateg;Hunedoara __ Hateg
Conac;Hateg __ Conac
Hategului;Hateg __ Hategului
Uzinei;Hateg __ Uzinei
Simeria;Hunedoara __ Simeria
Timis;Romania __ Timis
Deta;Timis __ Deta
Lugoj;Timis __ Lugoj
Florilor;Lugoj __ Florilor
Scolii;Lugoj __ Scolii
Timisoara;Timis __ Timisoara
Aradului;Timisoara __ Aradului
Martirilor;Timisoara __ Martirilor
Republicii;Timisoara __ Republicii
Sagului;Timisoara __ Sagului

