



odovzdávať riešenia? Lebo KSPáci prerábajú celú stránku (stará stránka zostarla a pokazila sa). Prerábacie práce idú pomalšie ako sme si pôvodne mysleli a teda niektoré veci meškajú. Odovzdávanie príkladov na novej stránke už funguje, ale mnohé časti stránky stále nie sú v použiteľnom stave, čo teda znamená, že odovzdávať nemôžete.

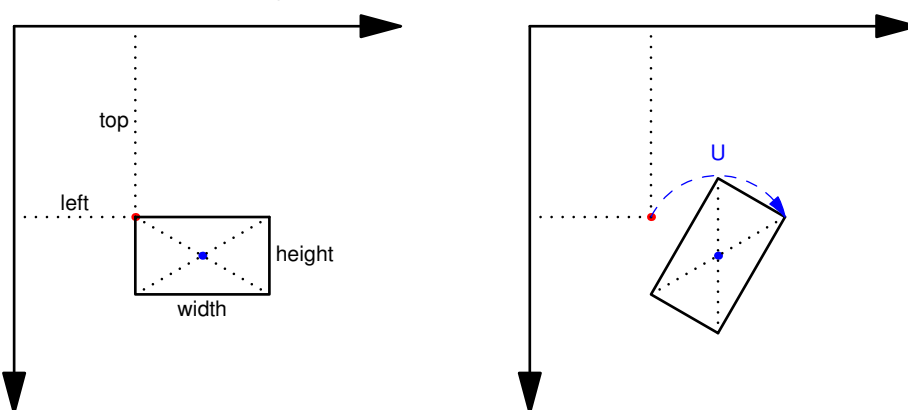
Jednou z vecí, ktorá na novej stránke chýba je dizajn. No a viete ako to chodí, na robenie dizajnu treba vedieť písať v **CSS**, ktoré väčšina vedúcich nevláda a tí čo ho ovládajú musia robiť iné veci.

Ale vy, účastníci, ste veľmi šikovné deti a určite nám s tým CSSkom trochu pomôžete. Aby stránka nebola nudná, ale akčná a ohromujúca, potrebujeme, aby sa všetky tlačidlá mohli otáčať a zväčšovať.

## Úloha

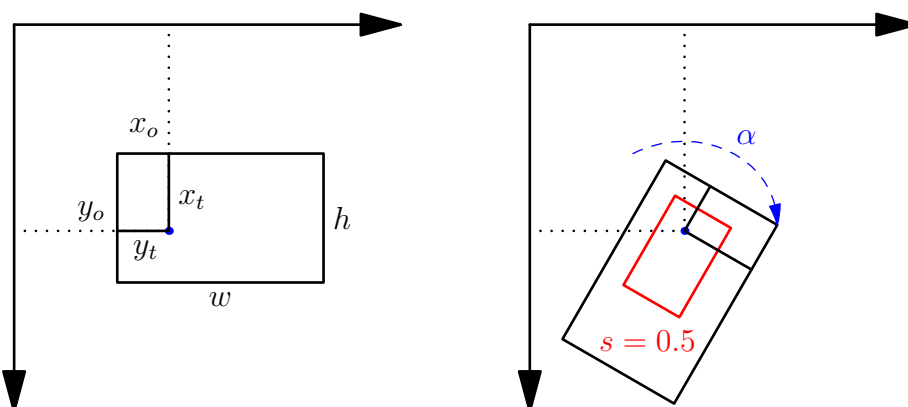
Tlačidlo si pre jednoduchosť predstavíme ako obdĺžnik. Obyčajné nezväčšené tlačidlo vieme nastaviť tak, že mu nastavíme rozmery a pozíciu. V CSS na to slúžia atribúty **width** (šírka), **height** (výška), **left** (vzdialenosť od ľavého okraja obrazovky) a **top** (vzdialenosť od horného okraja obrazovky).

Keď chceme aby bolo tlačidlo väčšie či menšie, jednoducho mu správne nastavíme **width** a **height**. Ak chceme tlačidlo otočiť, použijeme na to **transform: rotate(Udeg)**, čo otočí tlačidlo okolo jeho stredu o uhol  $U$  stupňov v smere hodinových ručičiek. Dôležité je, že táto transformácia sa udeje až po tom, ako sa tlačidlo umiestni podľa **left**, **top**, **width** a **height**.



Nám však otáčanie okolo stredu nestačilo, a potrebujeme tlačidlá otáčať podľa rôznych bodov. Popis tlačidla si predstavujeme takto:

1. Najprv tlačidlu určíme rozmery  $(w, h)$  a pozíciu ťažiska  $(x_t, y_t)$  – jeho vzdialenosti od ľavého a horného okraja obdĺžnika.
2. Následne určíme pozíciu ťažiska na obrazovke,  $(x_o, y_o)$ .
3. Potom tlačidlo otočíme okolo ťažiska v smere hodinových ručičiek o uhol  $\alpha$ .
4. Napokon tlačidlo zväčšíme okolo ťažiska  $s$  násobne.



Na vstupe dostanete popis tlačidiel podľa našej predstavy a vašou úlohou je vygenerovať CSS kód, ktorý zobrazí tlačidlá správne.

## Formát vstupu

Na prvom riadku vstupu je číslo  $n$  udávajúce počet tlačidiel. Nasleduje  $n$  riadkov, z ktorých každý popisuje jedno tlačidlo. Popis tlačidla pozostáva z 8 čísel,  $w, h, x_t, y_t, x_o, y_o, \alpha, s$ , pričom prvých 7 čísel je celých a

posledné číslo je reálne, s presne jednou cifrou za desatinnou čiarkou. Čísla su udávané v pixloch. Platí, že  $1 \leq w, h \leq 1000$ ,  $0 \leq x_c \leq w$ ,  $0 \leq y_c \leq h$ ,  $0 \leq x_o, y_o \leq 1000$ ,  $0 \leq \alpha < 360$ ,  $0 < s \leq 10$ .

### Formát výstupu

Pre každé tlačidlo na vstupe vypíšte jeden riadok – CSS príkaz na zobrazenie tlačidla vo formáte: “.itemN {width: Wpx; height: Hpx; left: Lpx; top: Tpx; transform: rotate(Udeg); }”.

To znamená, že  $N$ -té tlačidlo s rozmermi  $(W, H)$  sa najprv umiestni tak, aby jeho ľavý horný roh bol na pozícii  $(L, T)$  a následne sa otočí okolo svojho stredu o  $U$  stupňov. Rozmery a pozíciu píšeme v pixloch.

Formát musíte dodržať presne<sup>1</sup>, všetky čísla musia mať presne jednu cifru za desatinnou čiarkou (výsledok zaokrúhlite) a musí platiť, že  $0 \leq U < 360$ .

### Príklad

vstup	výstup
<pre>2 200 100 50 75 400 200 45 0.5 200 100 50 75 400 200 90 2.0</pre>	<pre>.item1 { width: 100.0px; height: 50.0px; left: 376.5px; top: 183.8px; transform: rotate(45.0deg); } .item2 { width: 400.0px; height: 200.0px; left: 250.0px; top: 200.0px; transform: rotate(90.0deg); }</pre>

V ukázkovom výstupe je z technických príčin každý riadok rozdelený na tri. Váš program má vypísať jedno pravidlo len na jeden riadok.

## 3. Trhanie vriec

kat. T; 0 b za popis, 20 b za program

*Bôby... Predo mnou bôby, za mnou bôby, pred bôbom bôby, za bôbom bôby, pred Bobom bôby... Všade samé bôby. Ako keby sa s nimi vrece roztrhlo. A veru sa aj roztrhlo. A nie jedno.*

Farmár Bob, vlastní obrovské pole a na ňom pestuje bôby. Bôby sú zdravé a sadia sa veľmi jednoducho: stačí priniesť vrece semienok bôbov na pole a vrece roztrhnúť. Semienka z roztrhnutého vreca sa rozprsknú po okolí a pokryjú súvislú plochu poľa. Bob je v trhaní vriecí neskutočne zručný a dokáže roztrhnúť vrece tak, aby semienka pokryli ľubovoľnú súvislú plochu poľa.

Predvčerom si Bob zaumienil, že spraví zo svojho poľa umelecké dielo. Neznáša stereotypy a tak chce, aby každý kúsok jeho poľa vyzeral inak. Cez internet si kúpil niekoľko rôznych odrôd bôbov a semienka každého druhu si dal zabaliť do osobitného vreca. Teraz rozmýšľa, ako vysadiť semienka.

### Úloha

Bob má  $n$  vriec so semienkami bôbov a v každom vreci má inú odrodu. Bobove pole je mriežka políčok  $r \times s$ , pričom celkový počet políčok je  $r \cdot s = 2^n$ . Na každom políčku môžu rásť bôby ľubovoľných odrôd (kľudne viac odrôd na jednom políčku) a dve políčka vyzerajú inak práve vtedy, ak na nich rastie iná množina bôbov.

Bob môže každú odrodu bôbov vysadiť na nejakú súvislú plochu políčok – políčka sú považované za susedné, ak sa dotýkajú stranami. Chce to spraviť tak, aby každé políčko vyzeralo inak, teda na každom políčku poľa bola iná množina bôbov – bude teda existovať aj políčko, na ktorom nie sú žiadne bôby, aj políčko na ktorom sú všetky bôby.

Vašou úlohou je pre dané  $r$  a  $s$  navrhnúť, ako má Bob roztrhnúť jednotlivé vrecia. Pre každú odrodu určite, na ktorých políčkach sa bude nachádzať.

### Formát vstupu

Na vstupe sú dve čísla  $r$  a  $s$ , oddelené medzerou. ( $4 \leq r, s \leq 256$ ,  $rs = 2^n$ , pre nejaké celé  $n$ .)

### Formát výstupu

Vypíšte popisy  $n$  odrôd oddelené prázdnyimi riadkami.

Popis  $k$ -tej odrody pozostáva z  $r$  riadkov dlhých  $s$  znakov, pričom  $j$ -ty znak v  $i$ -tom riadku má byť veľké  $k$ -te písmeno anglickej abecedy, v prípade, že na políčku  $(i, j)$  má byť zasadená  $k$ -ta odroda, a bodka v opačnom prípade. Pre istotu si pozri ukázkový výstup.

Plocha pokrytá každým písmenom musí byť súvislá, hoci plocha pokrytá bodkami nemusí. Za posledným popisom nevypisujte prázdny riadok.

<sup>1</sup>chceme predsa, aby ten kód vyzeral slušne a jednotne

Pokiaľ je možných riešení viacero, vypíšte ľubovoľné z nich.

### Príklad

vstup

```
4 4
```

výstup

```
AAAA
AAAA
....
....

BBB.
B...
BBBB
....

.CCC
.C..
CC..
CC..

.D..
DDD.
D.D.
D.D.
```

## 4. Turista programuje

kat. T; 0 b za popis, 20 b za program

Určite ste o ňom už počuli. Neuveriteľný chlapec, ktorého prezývajú Turista, sa narodil v neďalekom Bielorusku. Už ako jedenástorčný sa zúčastnil IOI (Medzinárodná olympiáda v informatike) a získal striebornú medailu. Ďalších 6 rokov mal samozrejme zlato, z čoho trikrát bol úplne prvý. Vyhral už asi každú súťaž, ktorú mohol (napr. Google Code Jam, Yandex.Algorithm, Facebook Hacker Cup). V Topcoderovi sa mu podarilo vyriešiť úlohu za 43 sekúnd (prečítanie zadania, naprogramovanie riešenia a submitnutie).

Pýtate sa ako je to možné? Po prvé, je to gényus a po druhé, má doma veľa čokolády. A práve čokoláda bude pravdepodobne veľmi dôležitá pri jeho nasledujúcej súťaži.

Čas, ktorý Turista potrebuje na vyriešenie úlohy nezávisí od jej obtiažnosti ale od toho, ako veľmi je unavený. Prvú úlohu vyrieši za minútu, druhú za dve minúty, tretiu za tri minúty a tak ďalej. Avšak namiesto riešenia úloh sa môže kedykoľvek rozhodnúť zjesť čokoládu a obnoviť si energiu. Zjesť čokoládu mu trvá  $c$  minút. Po tom, ako zje čokoládu, mu bude vyriešenie ďalšej úlohy trvať opäť len jednu minútu, no začne sa rýchlejšie unavovať. Po prvej čokoláde mu budú úlohy trvať postupne 1, 3, 5... minút a po druhej čokoláde 1, 4, 7... Po zjedení  $k$  čokolád stúpa čas riešenia úloh o  $k + 1$ .

### Úloha

Na súťaži je  $n$  úloh, ktoré musí Turista vyriešiť.

Vyriešenie prvej úlohy trvá Turistovi 1 minútu. Vyriešenie úlohy tesne po tom ako zjedol čokoládu mu trvá 1 minútu. Ak Turista práve nezjedol čokoládu, ale už predtým ich zjedol  $k$  a predošlú úlohu vyriešil za  $x$  minút, ďalšia úloha mu zaberie  $x + k + 1$  minút. Zjedenie čokolády mu trvá  $c$  minút.

Pre dané  $n$  a  $c$  zistite, kedy má Turista zjesť čokolády, aby mu vyriešenie všetkých úloh dokopy trvalo čo najkratšie.

### Formát vstupu

Na vstupe je jediný riadok obsahujúci dve celé čísla  $n$  a  $c$ ,  $1 \leq n, c \leq 1\,000\,000$ .

### Formát výstupu

Na prvý riadok výstupu vypíšte najmenší možný čas, za ktorý môže Turista vyriešiť všetky úlohy. Na druhý riadok vypíšte počet čokolád  $k$ , ktoré by mal počas toho zjesť. Na tretí riadok vypíšte  $k$  čísel oddelených medzerami, udávajúce, po koľkých vyriešených príkladoch by mal Turista zjesť čokoládu.

Pokiaľ je možných riešení viacero, vypíšte ľubovoľné z nich.

## Príklady

vstup

10 4

výstup

37  
2  
5 8

vstup

10 47

výstup

55  
0

*V prvom príklade, keď po 5. a 8. úlohe zje čokoládu, riešenie mu bude trvať  $1 + 2 + 3 + 4 + 5 + 4 + 1 + 3 + 5 + 4 + 1 + 4 = 37$  minút. V druhom príklade sa mu neoplatí jesť čokoládu.*