



1. feladat – Véletlenszám

Adott a következő 32 bites egész számokat előállító véletlenszám generátor C/Pascal nyelvben implementálva:

```
unsigned int m_w;
unsigned int m_z;

void resetMyRandom() {
    m_w = 11;
    m_z = 173;
}

unsigned int myrandom(unsigned int M) {
    m_z = 36969 * (m_z & 65535) + (m_z >> 16);
    m_w = 18000 * (m_w & 65535) + (m_w >> 16);
    return ((m_z << 16) + m_w) % M;
}
```

```
var m_w,m_z:longword;

procedure resetMyRandom;
begin
    m_w:= 11;
    m_z:= 173;
end;

function myrandom(M:longword):longword;
begin
    m_z:= 36969 * (m_z and 65535) + (m_z shr 16);
    m_w:= 18000 * (m_w and 65535) + (m_w shr 16);
    myrandom:= ((m_z shl 16) + m_w) mod M;
end;
```

- A & B - bitenkénti ÉS művelet
- A >> n - bitenkénti eltolás jobbra n bittel
- A << n - bitenkénti eltolás balra n bittel
- A % M - modulo, maradékos osztás – maradék
- A = B – értékadás

- A and B - bitenkénti ÉS művelet
- A shr n - bitenkénti eltolás jobbra n bittel
- A shl n - bitenkénti eltolás balra n bittel
- A mod M - modulo, maradékos osztás – maradék
- A := B - értékadás

A.) A fenti véletlenszám generátor segítségével állítsunk elő N ($0 < N \leq 10000$) darab **különböző** síkbeli pontot ($0 < M \leq 4095$). A pontok előállítását addig végezzük, amíg meg nem kapjuk az N darab különböző pontot. A generálás során bizonyos pontokat ismételten előállíthatunk, ezeket ismétléseknek nevezzük. Írassuk ki az ismétlések számát.

A pontok előállítása előtt hívjuk meg a resetMyRandom függvényt.

Egy síkbeli pont előállítása két myrandom függvényhívással történik:

```
unsigned int x = myrandom(M);
unsigned int y = myrandom(M);
```

```
x:= myrandom(M);
y:= myrandom(M);
```

B.) Ugyancsak a megadott véletlenszám generátor segítségével állítsunk elő N darab ($100000 < N \leq 1000000$) **különböző** síkbeli pontot ($0 < M \leq 4095$). **A pontok előállítása előtt hívjuk meg a resetMyRandom függvényt.** Írassuk ki az ismétlések számát.

C.) Ugyancsak a megadott véletlenszám generátor segítségével állítsunk elő N darab ($100000 < N \leq 1000000$) **különböző** síkbeli pontot ($0 < M \leq 65535$). **A pontok előállítása előtt hívjuk meg a resetMyRandom függvényt.** Írassuk ki az ismétlések számát.

Példa:

M = 4
N = 10

1.	2.	3.		4.	5.	6.	7.		8.		9.	10.	
(0,3)	(2,0)	(3,2)	(3,2)	(3,1)	(0,2)	(2,1)	(1,1)	(2,0)	(2,2)	(2,1)	(0,3)	(2,3)	(1,0)

Eredmény: 4

Az adatokat a VELETLEN.IN fájlból olvassuk be, amelynek három sora van. Minden sor két egész számot tartalmaz, az M és N számokat az A.), B.), illetve a C.) alpontokhoz. Az adatok helyesek, az M értéke úgy van megadva, hogy lehetséges legyen az N darab különböző szám előállítása.

Bemenet: VELETLEN.IN	Kimenet: VELETLEN.OUT
<M _A > <N _A >	<E _A >
<M _B > <N _B >	<E _B >
<M _C > <N _C >	<E _C >



2. feladat – SCRABBLE

Adott egy $n \times m$ (n és m páratlan értékek; $0 < n, m < 100$) méretű karakter-mátrix, amely elemei az angol-ABC kisbetűi vagy csillag-karakterek lehetnek. E mátrix egy egyszerűsített, sajátos scrabble-játék eredményeként jött létre. E játék arról szól, hogy a játékosok felváltva, az adott nyelven helyes szavakat tesznek le egy $n \times m$ méretű táblára. További szabályok, hogy:

- Az első letett szó valamely betűje a tábla középső cellájára kell kerüljön.
- A szavak vízszintesen (balról-jobbra) vagy függőlegesen (fentről-lefele) helyezhetők a táblára. (tehát egy szó iránya lehet vízszintes vagy függőleges)
- Minden következő szónak “metszenie” kell legalább egy már letett szót. Csak “metszési-kapcsolódás” megengedett, amely azt jelenti, hogy a letett szó *csak* ellentétes irányú előzőleg letett szavakkal kerül érintkezésbe (vízszintes szó csak függőlegessel, és fordítva)

Fontos észrevétel, hogy egy scrabble-táblán minden folytonos (maximális) vízszintes / függőleges betű-szekvencia helyes szó az adott nyelven.

Adva lévén a scrabble-táblát ábrázoló karakter-mátrix, rekonstruáljunk egy lehetséges játékot: milyen sorrendben, mely szavak lettek letéve a táblára. Több helyes válasz esetén a lexikográfikus sorrendbeli elsőt adjuk meg.

PÉLDA SCRABBLE.IN:

```
13 21
*****F*****
*****O*****
*****G*****
**SZARVASBOGAR*****
**A**E**S**KERTI
KEK**R**F**B**A**
**KAKASMANDIKOVAL**
*****R**N*L**K**
*****K**F*Y**U**
SAPIENTIA**O*A**L**
*****A*S**I**U**
*****N*****S**
INFORMATIKA*****
```

PÉLDA SCRABBLE.OUT:

```
KAKASMANDIKOVAL
BOLYAI
FARKAS
INFO
KALKULUS
KERTI
SAKK
KEK
SAPIENTIA
SZARVASBOGAR
FOGAS
TANA
INFORMATIKA
VERS
```