

Uppgift 1

Kirk skall anordna en schackturnering ombord. Men för att det ska gå måste du skriva en modul i Python som kan hantera schackbräden. Modulen skall användas som en ADT av ett större program och behöver fungera som den givna filen (spel.py) beskriver. Du bestämmer själv hur data lagras inuti ADTn men programmet skall fungera enligt körexemplet. Du får inte ändra på den givna koden i huvudprogrammet.

Det behöver finnas 4 funktioner som hanterar schackbrädet. En som skapar och returnerar ett tomt schackbräde. En som lägger till en pjäs på schackbrädet. En som flyttar en pjäs från en ruta till en annan på brädet. En som skriver ut alla rutor som har en pjäs på sig, utskriften skall se ut som körexemplet nedan när huvudprogrammet körs.

En pjäs är i huvudprogrammet endast en bokstav och du behöver inte anpassa din adt för att kunna spara någon annan typ av data.

Notera att en koordinat på spelbrädet är en bokstav och en siffra i en tuple.

Koordinaterna för insättning och förflyttning av pjäserna behöver du inte kontrollera, vi räknar med att de alltid är inom brädet och att drag är lagliga.

Körexempel:

```
( 'a', 1) : #
( 'a', 2) : B
( 'b', 1) : S
( 'b', 2) : B

( 'a', 1) : #
( 'a', 4) : B
( 'b', 1) : S
( 'b', 3) : B
```

	a	b	c	d	e	f	g	h	
8									8
7									7
6									6
5									5
4									4
3									3
2									2
1									1
	a	b	c	d	e	f	g	h	

Uppgift 2

Sisko har en stor databas av text ombord. Han försöker också skriva ett program som ska hjälpa personalen stava bättre. Han tänker använda databasen med text för att identifiera vilka ord som är rätt eller inte. Vetenskapsofficerarna har massor av konstiga idéer om häftiga algoritmer man kan använda för rättstavning men Sisko struntar i sådant schabbel och vill bygga något enklare.

För att Siskos program ska fungera behöver han förbereda data i databasen så att det bara blir ord kvar. Just nu finns det skräptecken i början och slutet av orden. Nedan finns en algoritm som du ska följa för att rensa alla skräptecken.

1. Läs in ett ord från filen, orden skiljs med ny rad.
2. Ta bort alla tecken som inte är bokstäver fram till första bokstaven i ordet.
3. Ta bort alla tecken som inte är bokstäver efter den sista bokstaven i ordet.
4. Gör om eventuella gemener (små bokstäver) till versaler (stora bokstäver), tecken som inte är bokstäver skall inte ändras.
5. Lägg till ordet i en lista.
6. Upprepa 1-6 tills alla ord i i filen har behandlats.
7. Skriv ut listan.

Observera att ditt program ska fungera för alla fall inte bara de som finns i **ordlista.txt**.

ordlista.txt innehåller:

```
hej
!hEj#$
!he#j#$
&h#e$j%$#@#
!@#$$%^&*() )__+=HeJ{ }|?>|
```

Körexempel:

```
HEJ
HEJ
HE#J
H#E$J
HEJ
```

Uppgift 3

Archer har varit ute på en resa och tittat på berg. Tyvärr gick filmrullen (man hade sådana innan digitalkameror) i kameran sönder när han sköt pilbåge (Archer... Get it?). Därför har han gett dig i uppgift att skriva ett program som ritar berg.

Du ska skapa en funktion som heter `draw_mountain`. Funktionen tar en parameter, ett heltal som beskriver hur högt berget skall vara och skriver sedan ut ett berg av den storleken. Ditt program ska sedan ta inmatning från användaren som inte behöver felkontrolleras och använda den inmatningen för att anropa funktionen.

Körexempel:

```
Mata in storleken på berget: 1
/\
```

Körexempel:

```
Mata in storleken på berget: 2
 /\
/\  \
```

Körexempel:

```
Mata in storleken på berget: 4

 /\
 /\  \
 /\    \
 /\      \
 /\        \
```

Körexempel:

```
Mata in storleken på berget: 9

 /\
 /\  \
 /\    \
 /\      \
 /\        \
 /\          \
 /\            \
 /\              \
 /\                \
```

Uppgift 4

Janeway ska spela det nya Star Trek rollspelet ombord. I rollspel slår deltagarna ofta tärningar. Janeway upplever att hennes tärning ofta slår väldigt låga siffror och är missnöjd med detta. För att slå sina tärningar använder Janeway modulen `random` i Python och vill därför att du testat den.

Hon vill att du skriver ett program som slår en 20 sidig tärning 20 000 gånger och sedan skriver ut antalet förekomster av varje sida i en tabell, se körexemplet. Det är viktigt att tabellen formateras på samma sätt som exemplet. Observera dock att resultaten inte kommer bli samma eftersom vi använder slumpgenerering, dock bör ditt resultat vara liknande.

Körexempel:

sida		antal förekomster
1		1007
2		990
3		1042
4		976
5		1033
6		1025
7		972
8		950
9		1037
10		948
11		1005
12		973
13		997
14		957
15		985
16		960
17		991
18		1044
19		1032
20		1076

Uppgift 5

Picard ska ta ett kort (pick a card.... Get it?) men innan han kan ta ett kort från kortleken behöver den sorteras. Normalt sett sorteras kortleken efter värdet på kortet och färgen i andra hand. Picard vill istället att kortleken bara sorteras efter färgen på korten i lexikografisk ordning (som en ordbok).

Din uppgift är att ändra anropet till funktionen `sort_deck` i den givna koden så att kortleken istället sorteras på färgen. Du får inte ändra på koden i funktionerna `create_deck` eller `sort_deck`.

Körexemplet:

```
(1, 'clubs')
(2, 'clubs')
(3, 'clubs')
(4, 'clubs')
(5, 'clubs')
(6, 'clubs')
(7, 'clubs')
(8, 'clubs')
(9, 'clubs')
(10, 'clubs')
(11, 'clubs')
(12, 'clubs')
(13, 'clubs')
(1, 'diamonds')
(2, 'diamonds')
(3, 'diamonds')
(4, 'diamonds')
(5, 'diamonds')
(6, 'diamonds')
(7, 'diamonds')
(8, 'diamonds')
(9, 'diamonds')
(10, 'diamonds')
(11, 'diamonds')
(12, 'diamonds')
(13, 'diamonds')
(1, 'hearts')
.
.
.
```

Resterande del av tabellen utelämnad av platsskäl