CSC, KTH **Betygsgradering:** Grunduppgift medel, betyg E-A
DD1310/DD1314 våren 2014 (Python)

121 Master Mind

P-uppgiften ska göras individuellt. Läs CSC:s hederskodex innan du börjar!

Varudeklaration: Datastrukturer, klasser.

Du ska skriva ett program som spelar Master Mind. Användaren ska tänka på ett tal och datorn försöker gissa det. Så här kan det se ut:

```
Tänk på ett fyrsiffrigt tal med fyra olika siffror och tryck retur!
Jag försöker gissa talet och du anger rätt och fel med t ex RFF, där
   R innebär att en siffra är rätt och står på rätt plats
   F innebär att en riktig siffra står på fel plats
   (retur) innebär att inga av siffrorna finns med i talet

   Jag gissar 1023 och du svarar: RF
   Jag gissar 1245 och du svarar: RR
   Jag gissar 1267 och du svarar: RF
   Jag gissar 1346 och du svarar: RF
   Jag gissar 1605 och du svarar:
   Jag gissar 7243 och du svarar: RRRR
   Det tog alltså 6 gissningar, bättre kan jag!
```

Programmet läser ett svar och kodar det till ett heltal genom att låta varje R vara värt 10 och varje F värt 1, t ex RFF $\Rightarrow$ 12.
Så här kan programmet gå tillväga för att få fram sina gissningar:
Alla gamla gissningar och motsvarande svarskoder sparas. Fundera över hur du ska lagra dessa på ett bra sätt.

För att en gissning ska vara en rimlig lösning måste följande gälla:

1. Talet skall bestå av fyra olika siffror
2. Talet ska ge samma svarskod som varje gammal gissning fick som svar.

Programmet börjar med att gissa på 1023 (det första fyrsiffriga talet med fyra olika siffror). Nästa gång måste programmet välja en gissning som inte motsäger någon av de tidigare gissningarna (t ex genom att stega igenom talen från 1023 till 9876 och ta det första som uppfyller kraven ovan).
Vi tittar lite närmare på exemplet ovan. 1245 var programmets senaste gissning.

- Programmet ska nu söka efter ett tal som är större än 1245 och består av fyra olika siffror.
- När programmet hittat ett sådant tal (t.ex. 1246), jämför det detta med de gamla gissningarna.
  - 1023 (första gissningen) i jämförelse med 1246 ger koden 11 (RF), OK så långt.
  - 1245 (andra gissningen) i jämförelse med 1246 ger koden 30 (RRR), men den skulle givit 20 (RR).

- 1246 förkastas alltså, eftersom den inte gav rätt svarskod på andra gissningen.
- 1247, 1248, 1249 och 1250 förkastas av samma skäl. 1251 har inte fyra olika siffror. Programmet måste räkna upp till 1267 innan det hittar en bra gissning.
  - 1023 (första gissningen) i jämförelse med 1267 ger koden 11 (RF).
  - 1245 (andra gissningen) i jämförelse med 1267 ger koden 20 (RR).
- Alltså är 1267 en rimlig gissning och programmet skriver ut denna.

När ingen rimlig gissning existerar, ska programmet tala om detta.

Tips: Skriv en funktion compare(guess1,guess2) som returnerar svarskoden som erhålls då vi jämför två gissningar.

Extrauppgift, betyg C: Inför felkontroll av användarens inmatning.

Extrauppgift, betyg B: Programmet ska kunna spela spelet med ombytta roller, dvs datorn slumpar fram ett tal, människan gissar och programmet svarar. Ett exempel:

```
Hmmm... OK, Jag tänker på ett tal!
Din gissning: 1234 Då säger jag FF Din gissning: 1122 Då säger jag
   F Din gissning: 2345 Då säger jag RF Din gissning: 2673 Då säger
   jag RRR Din gissning: 2683 JAAA, det var rätt på 5 försök!
```

Observera att om användaren anger två lika siffror och en sådan finns med i det påtänkta talet, ska datorn bara ge rätt för den ena. Om någon av dem står på rätt plats, ska man få ett R, annars ett F.

Extrauppgift, betyg A: Gör ett grafiskt användargränssnitt där man kan köra bägge spelvarianterna.