Uppgift 1

Ett (svenskt) registreringsnummer består av tre bokstäver mellan A och Z (förutom bokstäverna I,Q och V) samt tre siffror. I denna uppgift ska du skapa ett program som ber användaren mata in två registreringsnummer och därefter skriver ut hur många nummer som maximalt finns mellan dem. Detta ska du lösa genom att dels på en separat modul med namn `regnr.py` skapa en abstrakt datatyp som åtminstone stödjer följande primitiver:

create_regnr(tecken, siffror) Skapar en representation av ett registreringsnummer från tecknen `tecken` och siffrorna `siffror`.

is_same(r1,r2) Ger `True` om `r1` och `r2` representerar samma registreringsnummer

next(r) Returnerar en representation av nästa registreringsnummer givet `r`. Nästa registreringsnummer anses vara en ökning av talet om det är mindre än 999, annars blir talet 0 och sista bokstaven ökas (A blir B, B blir C osv). Om sista bokstaven var Z blir den A och näst sista bokstaven ökas. Samma sak gäller så klart om de två sista bokstäverna var ZZ. Registreringsnumret `ZZZ 999` kan inte ökas.

Du ska därefter skapa ett huvudprogram som använder sig av din ADT för att lösa problemet enligt nedanstående körexempel:

Körexempel 1 (Användarinmatning i fet stil)

```
Mata in första registreringsnumret: AAA 001
Mata in andra registreringsnumret: AAA 003
Antal mellanliggande regisreringsnummer: 1
```

Körexempel 2

```
Mata in första registreringsnumret: AZZ 900
Mata in andra registreringsnumret: BAA 001
Antal mellanliggande regisreringsnummer: 100
```

Körexempel 3

```
Mata in första registreringsnumret: AAP 999
Mata in andra registreringsnumret: AAR 003
Antal mellanliggande regisreringsnummer: 3
```

OBS: Det är givet att det andra registreringsnumret alltid kommer efter det första (är ”större”).

Uppgift 2

En kung har gjort sig ovän med kungarna i alla angränsande länder och har därför anlitat de bästa arkitekterna i landet för att bygga en ointaglig borg. En efter en lägger arkitekterna fram sina ritningar över borgens layout, men de är nu så många och så komplicerade att kungen behöver hjälp att avgöra vilka ritningar som överhuvudtaget innebär slutna borgar. Skriv ett program `castle_check.py` som tar in ett filnamn på kommandoraden. Filnamnet hör till en fil som skall innehålla en ritning för en borg. Följande gäller för ritningar:

- Murar markeras med `-` (bindesstreck) eller `|` (lodstreck). Alla murtecken som står bredvid varandra sluter tätt, oavsett om de står lodrätt eller vågrätt.
- Borgtorn markeras med `+`. Två borgtorn som står bredvid varandra sluter tätt.
- Hörn mot hörn sluter också tätt.
- Kungens tronrum markeras med `K`.
- I övrigt består filen endast av blanksteg och nyrader.

Låt programmet läsa igenom filen och kontrollera om tronrummet står säkert. Med säkert menas alltså att `K`:et står inne i en sluten slinga av murar och borgtorn. Programmet skall också skriva ut hela filen i terminalen. Du kan testa dina program med de givna filerna `CASTLE1.TXT` t.o.m. `CASTLE8.TXT` som finns i katalogen `given_files/CASTLE_DESIGNS`. I körexemplen nedan ser man hur programmet har startats från kommandoraden.

Körexempel 1

```
$ castle_check.py CASTLE1.TXT
+-----+
|     |
|  K  |
|     |
+-----+
Kungen är säker.
```

Körexempel 2

```
$ castle_check CASTLE2.TXT
    +----+
    |    |
+---+    |
|        +--
+--+        +-+
   |   K      |
   |          |
   +----------+
Kungen är säker.
```

Körexempel 3

```
$ castle_check CASTLE4.TXT
+----+
|    |
|  K
|    +
+----+
Kungen är INTE säker!
```

Körexempel 4

```
$ castle_check CASTLE6.TXT
   +---+
K  |   |
   +---+
Kungen är INTE säker.
```

Uppgift 3

Efter jul är tomten trött och tar där med en relaxresa till Bahamas. För att ta sig från nordpolen till hotellet behöver tomten åka flyg, buss och sedan taxi. Eftersom taxibilarna har fullt upp just denna period (många som vill fly undan snö och rusk) måste dessa bokas i förtid. Din uppgift är att räkna ut vilken tidpunkt som taxin skall bokas för.
Tomten kommer att mata in sin utresetid från nordpolen i ditt program. Det tar 6,5 timmar att flyga från Nordpolen till Bahamas. Utöver detta tar det maximalt 20 minuter att ta sig till busstationen när man väl kommer fram. Bussarna går var 30:e minut (varje hel och halvtimme) från flygplatsen till resorten. Bussresan tar maximalt 1 timme. Det tar sedan ca 5 minuter att gå från fjärrbussterminalen till taxiparkeringen. Vi utgår ifrån att nordpolen och Bahamas har samma tidszon.

Körexempel 1

```
Mata in utresetid: 11:35
Taxin skall bokas för kl 19:35.
```

Körexempel 2

```
Mata in utresetid: 17:00
Taxin skall bokas för kl 01:05.
```

Körexempel 3

```
Mata in utresetid: 07:11
Taxin skall bokas för kl 15:35.
```

Uppgift 4

Skriv ett program som låter användaren mata in en person som är en förfader eller förmoder. Inmatningen sker med orden mamma/pappa. Personen matas in på en lång rad. Raden kan vara godtyckligt lång. Programmet skall sedan mata ut personen på ett kortare format som använder orden farfar/morfar/mormor/farmor/mor/far (se körexemplen).

Körexempel 1

```
Mata in din släkting: mammas mamma
Du matade in din mormor.
```

Körexempel 2

```
Mata in din släkting: pappas mamma
Du matade in din farmor.
```

Körexempel 3

```
Mata in din släkting: mamma
Du matade in din mor.
```

Körexempel 4

```
Mata in din släkting: mammas pappas pappas mamma
Du matade in din morfars farmor.
```

Körexempel 5

```
Mata in din släkting: pappas mammas pappa
Du matade in din farmors far.
```

Körexempel 6

```
Mata in din släkting: pappas pappas pappas pappas pappas
pappas pappas pappas pappas pappas pappas pappas pappas pappas
pappas pappas pappas pappas pappas pappas pappas pappas pappas
pappas pappa
Du matade in din farfars farfars farfars farfars farfars
farfars farfars farfars farfars farfars farfars farfars far.
```

Uppgift 5

Det är dags att skriva julrim på dina paket. Tyvärr är du inte så poetiskt begåvad så du behöver ett program som kontrollerar om ditt rim verkligen rimmar. I denna uppgift antar vi att: Två ord rimmar om en av dessa två krav stämmer (kanske inte helt grammatiskt korrekt, men vi antar att det duger):

- De slutar på samma sekvens av en vokal och en (eller godtyckligt många) konsonanter. T.ex. rimmar ”katt” och ”natt” och ”hit” och ”dit”. ”Katt” och ”pott” rimmar inte och inte heller ”katt” och ”gast”. Observera att ordet ”marskt” skulle rimma på ordet ”barskt” om det fanns.
- De slutar på samma sekvens av vokal-(godtyckligt många konsonanter)-vokal. T.ex. rimmar ”lucka” och ”rucka”, ”apa” och ”gapa”, ”mixtra” och ”blixtra”. Exempel på ordpar som inte rimmar är ”gapa” och ”ripa”, ”gapa” och ”rapé”, ”vila” och ”visa”.

Om bokstäverna är stora eller små spelar ingen roll. Eventuella efterföljande punkteringstecken (’.’, ’,’, ’?’ och ’!’) skall ignoreras.
Ditt program skall alltså låta användaren mata in sitt julrim (två meningar) och kontrollera om det rimmar. Vi håller oss till ”vanliga rim” (inga uddrim/allitterationer), alltså är det är bara sista ordet i respektive mening som är av intresse.

Körexempel 1

```
Mata in rimmet (på två rader):
Det här kommer få dig att gapa
som en liten apa

Ditt rim rimmar, grattis!
```

Körexempel 2

```
Mata in rimmet (på två rader):
När du ute i kylan är utfrusen
Denna gör nog susen!

Ditt rim rimmar, grattis!
```

Körexempel 3

```
Mata in rimmet (på två rader):
Hoho sa tomten onomatopoetiskt
Här får du något patetiskt

Ditt rim rimmar, grattis!
```

Körexempel 4

```
Mata in rimmet (på två rader):
Vad är väl en bal på slottet?
inte är det flott

Det rimmar INTE!
```

Körexempel 5

```
Mata in rimmet (på två rader):
En sådan här tyckte jag borde skaffas
Nu behöver du inte oroa dig för att polisen dig haffar

Det rimmar INTE!
```

TIPS: För er som i stressen glömt vilka bokstäver som är vokaler kommer dessa här: A, E, I, O, U, Y, Å, Ä, Ö