Introduktion till programmering i Python

DAT455 (Chalmers), DIT001 GU)
Tentamen 2020-10-10 kl 8:30-10:30 på distans via Canvas
Examinator Aarne Ranta, email aarne@chalmers.se

Tentan består av tre frågor. **Du måste svara på alla frågor**. Skalan är G/U. För att få ett G krävs ett av följande alternativ:

- minst 50% rätt för varje fråga för sig
- minst 80% rätt för två av frågorna och 20% för den återstående frågan

Svaret ges i form av en fil som ska heta `python_exam.py` och kunna importeras i Python3. Filen ska innehålla funktioner som specificeras i frågorna. Filen kan också innehålla kommentarer som Python3 ignorerar men lärarna kan läsa. Filen ska inte importera biblioteket graphics (samma som i Lab 2, tillgänglig på https://mcsp.wartburg.edu/zelle/python/graphics.py), men inga andra bibliotek.

Alla hjälpmedel är tillåtna, men man får inte kommunicera med andra människor under tentans gång, med undantag av email till aarne@chalmers.se ifall någonting är oklart.

Vårt bästa tips är att du faktiskt testar dina svar i en Python-tolk och ser till att den fungerar innan du lämnar in den. Lycka till!

Innan du börjar med frågorna: skriv följande kommentarer överst i din fil

```
# Jag bekräftar härmed att jag inte kommunicerar med andra personer än kursens lärare under tentans gång.

# Jag är medveten om att fusk i tentan kan leda till disciplinåtgärder.
```

Fråga 1. Interaktiva program: inreseregler

Landet L har satt upp regler för inresande till landet under en pågående pandemi. Reglerna är följande:

- **smittotalet** i ett land beräknas som antalet fall (i två veckor) per 100 000 invånare i landet
- om resenärens land har smittotal 50 eller högre, är landet **rött** och inresa förbjuden
- om resenärens land har smittotal 25 eller högre, men under 50, är landet **gult** och inresa tillåten med tio dagars karantän
- om resenärens land har smittotal lägre än 25, är landet **grönt** och inresa tillåten
- undantag: om resenärens land har smittotal som är lägre än smittotalet i L är landet grönt oavsett talet

Skriv en funktion `pandemic_rules()`, som tar som argument det aktuella smittotalet i L samt antalet fall och folkmängd i resenärens land och returnerar färgen. Skriv även en interaktiv funktion `apply_pandemic_rules()`, som tar smittotalet i L, ställer två frågor till resenären, beräknar färgen med hjälp av `pandemic_rules()`, och skriver ut beskedet enligt följande skärmdump. (Poängen med att dela upp detta i två funktioner är att själva beräkningen av färgen kan återanvändas, och att den enkelt kan testas för sig själv.)

```
>>> pandemic_rules(23,10000000,2000)
'grönt'

>>> apply_pandemic_rules(23)
Hur stor är ditt lands befolkning? 10000000
Hur många fall har ditt land haft? 2550
Ditt land är gult.
```

Fråga 2. Algoritmer med strängar och tal: delbarhet med 9

Ett snabbt sätt att avgöra om ett tal är delbart med 9 är att beräkna summan av dess siffror. Om man gör detta upprepade gånger får man till slut en enda siffra. Med undantag för talet 0 är den här siffran 9 om och endast om det ursprungliga talet är delbart med 9. Till exempel:

1998 → 1+9+9+8 → 27 → 2+7 → 9

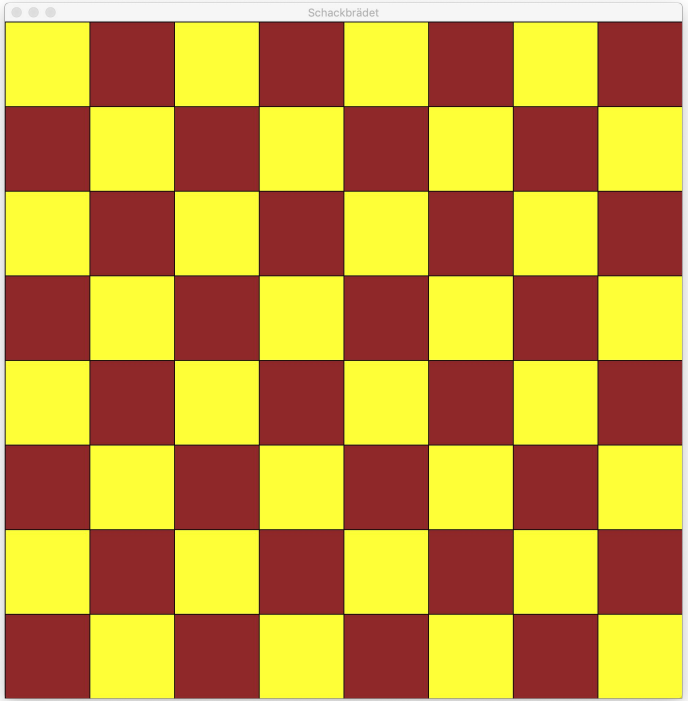
Skriv en Python-funktion som avgör delbarhet med 9 på det här sättet och även skriver ut de mellanliggande summorna vid varje steg. Följande skärmdump visar en körning av funktionen på talet 666666888888888 (6 stycken 6:or och 9 stycken 8:or).

```
>>> div9(666666888888888)
108
9
True
```

Fråga 3. Objekt, bibliotek, grafik

Ett schackbräde består av ett rutnät om 8x8=64 kvadratiska rutor. Varannan ruta är mörk, varannan ljus.

```
>>> Chessboard('brown','yellow',100).draw()
```



Bilden bredvid har skapats av en Python-klass `Chessboard`, som producerar ett fönster med bilden när man anropar dess metod `draw()`enligt skärmdumpen. Koden ska använda `graphics`-biblioteket (https://mcsp.wartburg.edu/zelle/python/graphics.py, samma som i kursboken och i Lab 2) och ta två färger som argument - den mörka och den ljusa. Fönstrets storlek ska vara 800x800 pixlar, förutom titelbalken, där titeln "Schackbrädet" ska visas. 800 beräknas som 8 gånger rutans storlek, som är det tredje argumentet i klassens konstruerare.

I den här frågan är det viktigt att välja och använda rätt metoder från biblioteket, initialisera klassen på ett rätt sätt, samt att skapa ett kompakt program som räknar ut rutornas positioner och färger utan alltför mycket upprepning. Ett litet tips är att klassen kan skrivas med ca 15 rader, så om du har många fler har du kanske gjort det alltför krångligt, vilket kan leda till färre poäng i tentan. Men för 50% räcker det att när du (och den som rättar tentan) kör anropet blir resultatet det samma som bredvid, samt förstås att färgerna och rutans storlek kan varieras.