

Grundläggande programmering i Python, 7.5 hp

Provmoment: Tentamen, 5 hp

Ladokkod: A237TG, A263TG & A274TG

Datum och tid: 2024-12-02, kl 13:00 – 2024-12-03, kl 12:59

Plats: Hemtentamen

Poäng:

- ❖ Godkänt resultat på Del 1 ger tillgång till Del 2.
- ❖ Maximal poäng på Del 2 är 40p.
- ❖ För godkänd tentamen krävs godkänt på Del 1 och minst 24p på Del 2.

Allmänna anvisningar:

- All kod ska vara lättläst och kommenterad. Rättande lärare måste kunna förstå hur du har tänkt
 - Förstår du inte frågan eller är osäker, kommentera hur du har förstått frågan. Kommentera sedan vad du vill åstadkomma med din kod
- Variabler ska ha meningsfulla namn
- Du får inte lämna in kod som genererar felmeddelande eller varningar
- Du får inte lämna kvar sektioner med bortkommenterad kod
- Om du inte vet hur hela en uppgift ska lösas, försök gör så mycket som möjligt ändå för delpoäng.

Försök om möjligt lämna programmet körbart och som ger/skriver ut delresultat. Komplettera gärna med extra diskuterande kommenterar som visar förståelse om det som inte kunde lösas i kod

Tentamen ska genomföras enskilt, dvs. samarbete är inte tillåtet. Det är inte heller tillåtet att använda någon form av AI-hjälpmedel, typ ChatGPT eller liknande. Genom att lämna in tentamen intygar du på heder och samvete att du har löst uppgifterna självständigt utan hjälp av någon annan person. Vid misstanke om fusk så kan plagiatgranskning ske. I övrigt finns inga begränsningar vad gäller hjälpmedel.

Nästkommmande tentamenstillfällen: TBD

Övrig information

Rättningstiden är i normalfall 15 arbetsdagar, till detta tillkommer upp till 5 arbetsdagar för administration.

Ansvariga lärare:

Carl Tinnsten, carl.tinnsten@hb.se, 070-284 88 06

Maila gärna frågor om du vill visa något annars ring. Lärare tillgängliga via telefon, tider 2/12 kl.13.00-18:00 och 3/12 kl.09:00-13:00.

Skulle det visa sig att det behövs ett förtydligande etc. på någon uppgift, så kommunicerar vi ut det till alla via Anslag i Canvas. Titta ifall något har dykt upp där innan du skickar en fråga.

Lycka Till!

Uppgift 1 (5p)

Skapa två funktioner $V_kon(r, h)$ och $A_kon(r, h)$ som beräknar volym och mantelytan (yt-arean) av en cirkulär kon. Därefter skriv ett huvudprogram "main()" som läser in värdena för radien och skriver ut volymen och arean av klotet, avrundat till två decimaler.

$$V = \frac{\pi r^2 h}{3} \quad A = \pi r \sqrt{r^2 + h^2}$$

Påvisa att funktionerna fungerar genom att skriva ett huvudprogram som anropar funktionerna $V_kon(r, h)$ och $A_kon(r, h)$ och returnerar värdena enligt följande utskrift.

```
1. Beräknar volym av en kon
2. Beräknar mantelytan av en kon
3. Avsluta programmet.
Ange det menyalternativ du vill köra: 1
Ange värden på radien 2.5
Ange värden på höjden 10
Volym av konen är 65.45 cm^3
1. Beräknar volym av en kon
2. Beräknar mantelytan av en kon
3. Avsluta programmet.
Ange det menyalternativ du vill köra: 2
Ange värden på radien 2
Ange värden på höjden 3
Arean av konen är 22.65 cm^2
1. Beräknar volym av en kon
2. Beräknar mantelytan av en kon
3. Avsluta programmet.
Ange det menyalternativ du vill köra: 3

Programmet avslutas.
```



Uppgift 2 (5p)

Skriv en kod som skapar en lista med slumpmässig genererad längd (max 100 element). Därefter fylls listan med slumptal mellan talen 0-9. Programmet skall sedan beräkna antal förekomster av varje tal i listan. Du kan importera randint från modulen random.

En exempelpkörning av funktionen kan se ut som följande:

De genererade talen är:

71487653881584708053

16581750568751156815

33423139888356766218

57686820068908667501

53371239098460318413

Antal förekomster av respektive tal är:

9 av talet 0

13 av talet 1

4 av talet 2

12 av talet 3

5 av talet 4

13 av talet 5

13 av talet 6

9 av talet 7

18 av talet 8

4 av talet 9

Uppgift 3 (5p)

Skriv ett program som ska skapa två listor, som sedan fylls med slumpstal. Därefter skapar programmet summeringen av talen som sedan skrivs ut enligt nedan.

Programmet frågar först efter hur många element som skall ingå i listorna, och skapar de tomma listorna A och B. Därefter fylls listorna (A och B) med slumpmässiga heltal mellan 1 och 20. En tredje lista (C), skall innehålla beräkningen och summan av de två första listorna, enligt nedan. Skriv ut listorna A, B och C parallellt i lodrät riktning (dvs varje par av värden presenteras på samma rad).

Formatering av utskriften skall vara enligt nedan.

Hur lång ska listorna vara? 5				
Lista A		Lista B		Lista C

19		2		19 + 2 = 21
3		18		3 + 18 = 21
9		4		9 + 4 = 13
3		7		3 + 7 = 10
3		17		3 + 17 = 20

Uppgift 4 (5p)

Nedanstående program ska skriva ut ett mönster av stjärnor, där antalet stjärnor ökar i varje rad med två stjärnor. Den första raden har bara en stjärna. Användaren ska ange antal rader i mönstret.

```
Ange antal rader: 6
      *
    * * *
  * * * * *
* * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
```

Tyvärr finns det några fel i programmet som gör att den inte fungerar korrekt. Din uppgift blir att rätta felen. Skriv en kommentar till varje ändring du gör, okommenterade ändringar ger inga poäng.

OBS! Du får inte lägga till eller ta bort kod, utan du får enbart justera redan skriven kod för att hitta felen och korrigera dessa.

```
1 rader = int(input("Ange antal rader: "))
2
3 k = 0
4
5 for i in range(1, rader+1):
6     for j in range(1, (rader+i)+1):
7         print(end=" ")
8
9     while k!=(2**i+1):
10        print("* ", end="") |
11        k += 1
12
13    k = i
14    print()
```

Uppgift 5 (20p)

I denna inlämningsuppgift kommer du att analysera och vidareförädla data från kameraövervakningar av hastigheter i Västra Götalands län. Datan kommer från Trafikverket och innehåller ett begränsat urval av hastigheter som dessa kameror registrerat på respektive mätplats den 11 september 2021 mellan kl. 07:00 – 18:00. All nödvändig information för att lösa nedanstående uppgifter finns i csv-filerna kameraData.csv och platsData.csv. Den ursprungliga kameraData.csv innehöll över 700 000 poster men ni får en nedbantad, men ändå en representativ version som "bara" innehåller ca 1 300 st. att arbeta med i stället. Det gör att ni lättare kan "manuellt" överskåda datan för att se om era beräknande program som ni ska skriva fungerar som de ska, genom att öppna csv-filen i ett godtyckligt kalkylprogram (t.ex. Microsoft Excel eller OpenOffice Calc).

Nedan följer en beskrivning av innehållet i respektive csv-fil:

kameraData.csv (skärmbild på de första raderna)

	A	B	C	D	E
1	MätplatsID	Gällande Hastighet	Hastighet	Datum	Tid
2	14075010	40	55	2021-09-11	11:15:31
3	14075010	40	54	2021-09-11	08:09:17
4	14075010	40	53	2021-09-11	13:02:41
5	14075010	40	49	2021-09-11	13:02:55
6	14075010	40	47	2021-09-11	13:14:45
7	14075010	40	45	2021-09-11	07:57:27

Varje rad i filen *kameraData.csv* innehåller en kameraregistrering med följande information:

- MätplatsID: Mätplatsens identifikationsnummer
- Gällande hastighet: Högsta tillåtna hastighet (km/h)
- Hastighet: Av kameran registrerad hastighet (km/h)
- Datum: Datum för registrering
- Tid: Klockslag för registrering

platsData.csv (skärmbild på de första raderna)

	A	B	C	D
1	MätplatsID	Namn	Vägnummer	Kommun
2	14002010	Bhpl Gestadvägen	E45	Vänersborg
3	14002020	Bhpl Gestadvägen	E45	Vänersborg
4	14002030	Bhpl Flybo	E45	Vänersborg
5	14002040	Bhpl Flybo	E45	Vänersborg
6	14002050	Frändefors	E45	Vänersborg
7	14002060	Frändefors	E45	Vänersborg

Varje rad i filen *platsData.csv* innehåller följande information:

- MätplatsID: Mätplatsens identifikationsnummer
- Namn: Namn på platsen där kameran är installerad
- Vägnummer: Vägnumret för den väg där kameran är installerad
- Kommun: Kommunen där kameran är installerad

- a) I denna uppgift ska du skapa ett enkelt menysystem som anropar de egendefinierade funktionerna som du kommer skapa för uppgift 5. När programmet har exekverat ett menyalternativ ska menyn åter komma upp och användaren ska kunna välja ett nytt

alternativ osv. Användaren får därigenom ett sammanhållet program och man behöver inte starta om programmet för att köra respektive uppgifter. Användaren avslutar programmet genom att välja Avsluta (alternativ 5).

Menysystemet ska ha följande struktur:

Meny

=====

1. Hämta data från fil – 5 a
2. Analysera data – 5 b
3. Analysera data – 5 c
4. Analysera data – 5 d
5. Avsluta

Välj menyalternativ (1-5):

Skapa en egendefinerad funktion `read_file(file_name)` som öppnar en csv-fil med angivna namnet till parametern `file_name` och läser in dess innehåll och returnerar resultatet i en tvådimensionell lista. Tanken är att när du sen kallar på denna funktion två gånger med argumenten `kameraData.csv` respektive `platsData.csv` så kommer datan att finnas inläst och redo för att kunna utföra efterkommande deluppgifter. Döp dina listor till `kameradata` respektive `platsdata`. När du använder funktionen `open()` för att påbörja inläsningen av filerna, skicka med `encoding = 'UTF-8'` som argument för att säkerställa att inläsningen lyckas, och att teckentolkningen på bl.a. åäö blir korrekt (om inte så varierar det lite hur det betar sig, beroende på operativsystem).

Avsluta uppgiften med ett huvudprogram som anropar funktionen och sedan skriver ut de tre första raderna i var och en av de två listorna, för att verifiera att det funkar som det skall. Utskriften ska se ut som följande:

Kameradata-filen:

```
[['MätplatsID', 'Gällande Hastighet', 'Hastighet', 'Datum',
'Tid'], ['14075010', '40', '55', '2021-09-11', '11:15:31'],
['14075010', '40', '54', '2021-09-11', '08:09:17']]
```

Platsdata-filen:

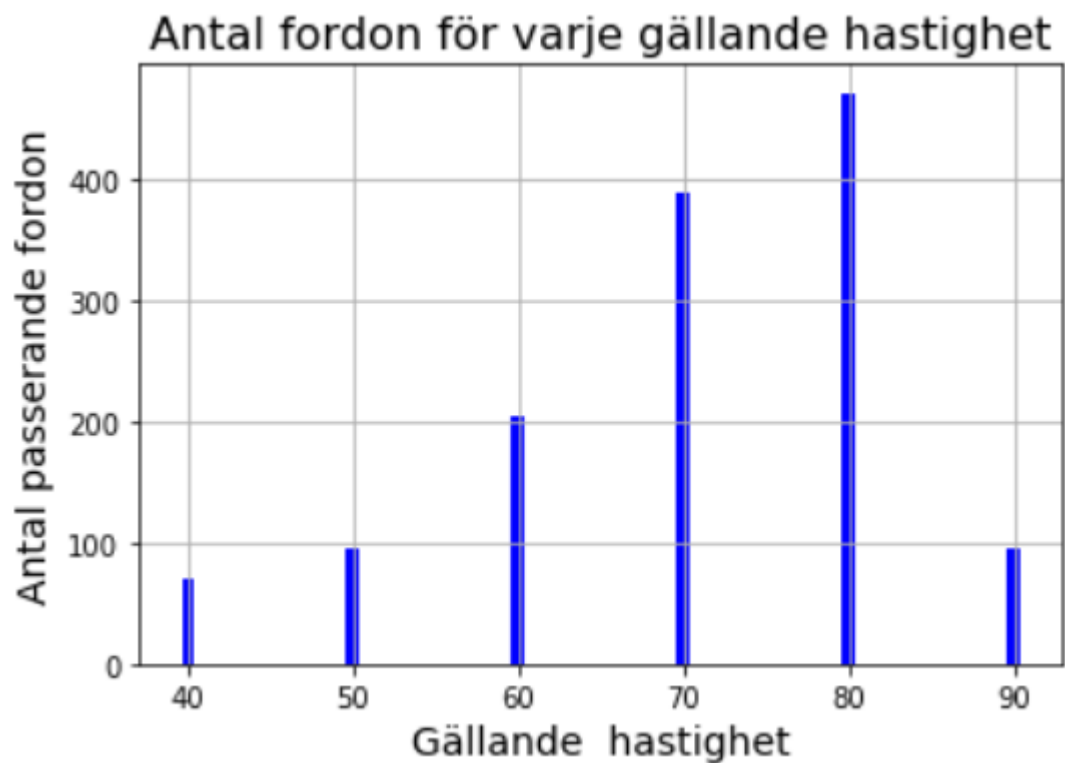
```
[['MätplatsID', 'Namn', 'Vägnummer', 'Kommun'], ['14002010',
'Bhpl Gestadvägen', 'E45', 'Vänersborg'], ['14002020', 'Bhpl
Gestadvägen', 'E45', 'Vänersborg']]
```



- b) Skapa en egendefinerad funktion `antal_bilar(kamera_data)` som beräknar totalt antal fordon som passerat vid var och en av de olika gällande hastigheterna. Funktionen ska skriva ut resultatet i textform, samt även rita ett stapeldiagram över resultatet. Avsluta uppgiften med ett huvudprogram som visar att din funktion fungerar korrekt och ger nedanstående utskrift och diagram.

Det finns 69 mätningar där gällande hastighet är 40 km/h
Det finns 96 mätningar där gällande hastighet är 50 km/h
Det finns 204 mätningar där gällande hastighet är 60 km/h
Det finns 389 mätningar där gällande hastighet är 70 km/h
Det finns 471 mätningar där gällande hastighet är 80 km/h
Det finns 94 mätningar där gällande hastighet är 90 km/h

Totalt passerade 1323 bilar.

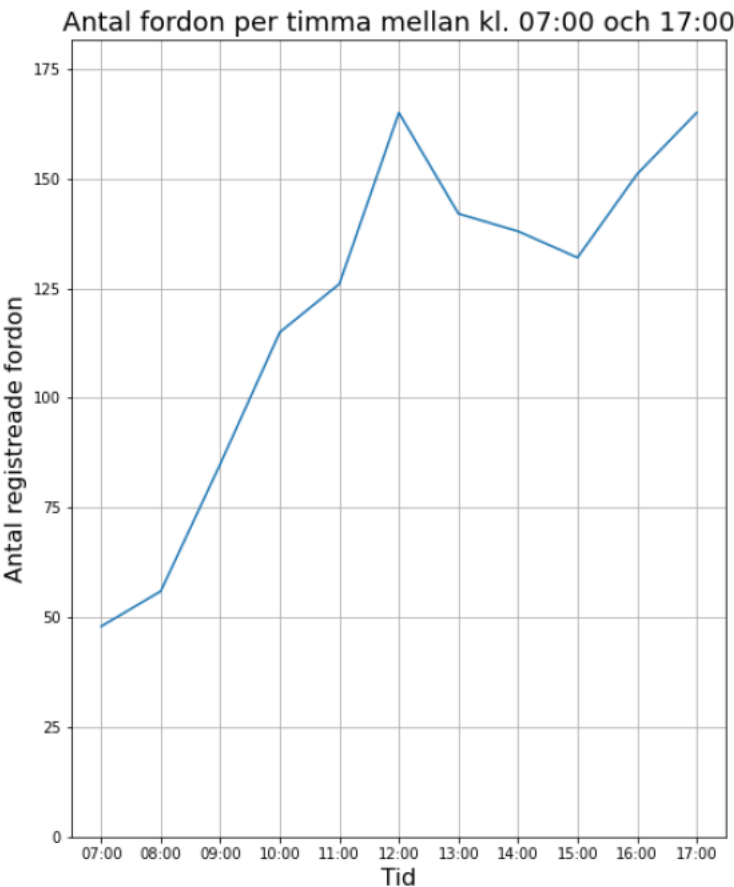


- c) Skapa en egendefinerad funktion `antal_kameror(plats_data)` som räknar antal kameror i varje kommun, sorterar resultatet i alfabetisk ordning efter kommun och presenterar resultatet som en tabell. Ni behöver inte ta hänsyn till om å, ä och ö sorteras fel, det kan bero på felaktig ordning i aktuell teckenkodning. I slutet ska även det totala antalet kameror summeras. Avsluta uppgiften med ett huvudprogram som visar att din funktion fungerar korrekt. Tabellen ska strukturellt se ut som nedan (obs värdena nedan är på denna deluppgift bara påhittade, så att du själv får träna på att kontrollera att dina resultat verkar stämma):

Kommun	Antal kameror
-----	-----
Alingsås	17
Bengtsfors	4
Essunga	10
Falköping	28
...	
Vara	7
Vänersborg	20
Vårgårda	13
-----	-----

Det finns totalt 297 kameror.

- d) Skapa en egendefinerad funktion `bilar_timme(kamera_data)` som ritar nedan diagram. Diagrammet visar summan av antal fordon som kamerorna har registrerat över alla mätplatser i alla kommuner uppdelat per timma mellan kl. 07 och 17. I diagrammet är x-axeln graderad i timmar mellan 07:00 och 17:00 och y-axeln i antal registrerade fordon, även rubrikerna skall vara med enligt nedan. Tips, använd den tillåtna modulen `datetime` i denna deluppgift.



Vi får lov att komma ihåg att detta diagram blir väldigt konceptuell i och med det nedbantade kameraData.csv-dataset:et. Verkligt antal bilar som åkte denna dag är snarare i storleksordningen ca 50000(!) per timma, jämfört med ca de 100 som vi får här