

Anvisningar

- Tentan skickas in senast klockan 12:00 på blackboard under fliken ”tenta”. Inskickningsmöjligheten stänger klockan 12:00. Ni kan skicka in hur många gånger som helst och sista försöket är det som räknas. Inga undantag görs för sen inlämning, så skicka in i tid!
- Svaren sparas i helst i en .py-fil där svaren på frågorna ges som kommentarer och svar i form av kod är i körbart skick. En vanlig text-fil (dvs .txt) går också bra. En pdf-fil är ok, men se i sådana fall till att det inte uppstår formateringsproblem vid konverteringen till pdf. Examinator/rättande lärare bedömer endast det som har skickats in.
- Det är tillåtet att använda en Pythoninstallation, men det är inte tillåtet att kopiera lösningar från andra studenter eller internet. Det är heller inte tillåtet att på något sätt samarbeta med andra studenter (eller individer). Svaren kommer att undersökas för plagiering.
- Vid varje uppgift anges hur många poäng som maximalt utdelas för uppgiften.
- Denna tentamen består av två delar: **Del 1** och **Del 2**. På Del1 kan man få maximalt 30 poäng. På Del 2 kan man få maximalt 20 poäng.
- För betyg 3 gäller det att få minst 15 poäng på Del 1 (Del 2 behöver alltså inte göras för att få detta betyg).
- För betyg 4 gäller det att
 - få minst 15 poäng på Del1 och minst 8 poäng på Del 2,
 - samt muntligen examineras senare idag (Tenta 2) på frågorna i Del 2 samt relaterade frågor, varefter en samlad bedömning avgör betyget.

- För betyg 5 gäller det att
 - få minst 15 poäng på Del1 och minst 13 poäng på Del 2,
 - samt muntligen examineras senare idag (Tenta 2) på frågorna i Del 2 samt relaterade frågor varefter en samlad bedömning avgör betyget.
- För att få göra den muntliga tentamen för högre betyg så måste man vara anmäld till den. Om man ändrar sig och inte längre är intresserad av ett högre betyg så avstår man från att ansluta till den muntliga redovisningen. Anslut i tid om i vill examineras muntligt. Om ni ansluter till mötet sent, finns mindre tid för att göra en bedömning.

Del I (Denna del består av sex uppgifter)

Uppgift I (6p)

- a) Betrakta programmet nedan. Syftet med programmet är att användaren anger en summa pengar i for av ett positiv heltal. Sedan räknar programmet ut hur många hundralappar, femtiolappar, tjugolappar, och enkronor denna summa motsvarar och skriver ut detta på skärmen. Programmet fungerar emellertid inte som det ska. Er uppgift är att lägga till kod på raderna 5, 7, och 9 samt ändra rad 10 så att programmet fungerar som det är tänkt. Endast dessa rader får ändras. Svara på följande sätt:

rad 5: ...

rad 7: ...

rad 9: ...

rad 10: ...

```

1pengar = int(input("Ange hur mycket pengar:"))
2
3rest = pengar
4antal_hundralappar = rest//100
5
6antal_femtiolappar = rest//50
7
8antal_tjugolappar = rest//20;
9
10antal_enkronor = pengar
11
12print(pengar, "kronor motsvarar", antal_hundralappar,
13      "hundralappar,", antal_femtiolappar,
14      "femtiolappar,", antal_tjugolappar,
15      "tjugolappar och", antal_enkronor, "enkronor.")

```

- b) Betrakta följande uttryck: `not((2 A 4) and (1.0 B 1.0))` där **A** och **B** ska bytas ut mot relationsoperatorer (t.ex. `==`, `<`, `>`) sådana att uttrycket blir `False`. Uppgiften är att ange alla val av sådana relationsoperatorer **A** och **B** så att uttrycket blir `False`.

Uppgift 2 (5p)

a) Betrakta följande program:

```
x = int(input("Ange x:"))
value = 0
for i in range(0,x,2):
    value += i*(i-1)
print(value)
```

Skriv ett nytt program som fungerar på samma sätt som detta, men inte använder sig av for-loop(ar). Utskriften på skärmen ska bli densamma för det nya programmet som för det gamla programmet för samma val av input.

b) Betrakta följande program:

```
import random

x = int(input("Ange x:"))
y = int(input("Ange y:"))

i = 0
j = 0

max_value = 0

while i < x:
    val1 = random.randint(0,x)
    while j < y:
        val2 = random.randint(0,y)
        if max_value < val1*val2:
            max_value = val1*val2
        j += 1
    j = 0
    i += 1

print("Det största värdet var:", max_value)
```

Skriv ett nytt program som fungerar på samma sätt som detta, men inte använder sig av while-loop(ar). Utskriften på skärmen ska bli densamma för det nya programmet som den för det gamla programmet för samma val av input.

Uppgift 3 (7p)

- a) Betrakta funktionen `foo_bar` nedan där vissa delar har dolts. Ange vad som ska stå bakom de dolda delarna A, B, C och D, så att funktionen returnerar `False` vid följande anrop: `foo_bar("1235321")`, `foo_bar("anna")`, `foo_bar("lol")`, `foo_bar("gg")`, men returnerar `True` vid följande anrop: `foo_bar("1235322")`, `foo_bar("annars")`, `foo_bar("loled")`, `foo_bar("stonks")`.

Obs! det finns såklart flera lösningar på detta problem. Alla lösningar som fungerar godtages.

```
def foo_bar(string):
    string2 = A
    for i in range(len(B)):
        string2 += C
    return D
```

- b) Betrakta funktionerna `bar` och `foo` nedan.

```
def bar(value):
    return value*(value-1)*(value-2)

def foo(value):
    # ...
    # ...
    # ...
    while True:
        if value > 10:
            break
        else:
            value += bar(value)
    return(value)
```

Antag att vi anropar `foo` med heltalsvariabeln `var` som argument, dvs. `foo(var)`. Svara nu på följande delfrågor. För varje delfråga måste ni kort motivera svaret.

- Vad blir $x = \text{foo}(var)$ om $var > 10$?
- För vilka positiva heltal var gäller det att `foo(var)` returnerar ett värde och inte fastnar i en oändlig loop?
- Vad händer om var är ett negativt heltal vid anropet `foo(var)`?
- Ersätt raderna med "# ..." med kod så att `foo` returnerar ett värde för de fall där den nuvarande implementationen av `foo` inte returnerar ett värde.

Uppgift 4 (5p)

Betrakta följande funktioner `count_function`, `foo` och `bar`.

```
import random

def count_function(arg):
    count = 0
    for i in range(arg):
        for j in range(i):
            count += j
    return count

def foo(arg1, arg2):
    f = open(arg1, "w")
    for i in range(arg2):
        for j in range(i):
            f.write(str(j))
            f.write(random.choice("abc"))
        f.write("\n\n")
    f.close()

def bar(arg1, arg2):
    # 1
    foo(arg1, arg2)
    count = 0
    f = open(arg1, "r")
    for line in f:
        # 2
        pass
        # 3

    f.close()
    return count
```

- a) Beskriv kortfattat vad funktionen `foo` gör.

För frågor b) och c) nedan så utgör era svar endast den delen kod som efterfrågas. Ni ska alltså inte svara med den kompletta koden för alla funktioner.

- b) Lägg till kod vid ”#1” så att ett felmeddelande skrivs ut på skärmen när `arg1` inte är en sträng och `arg2` inte är ett positivt heltal. Dessutom returneras värdet 0 i dessa fall.
- c) Ersätt koden mellan ”#2” och ”#3” (dvs. ”pass”) med kod så att om `x` är ett positivt heltal så returnerar `count_function(x)` samma värde som `bar("fil.txt", x)`.

Uppgift 5 (4p)

Betrakta följande program:

```
def foo(string1, string2):
    var1 = set(string1)
    var2 = set(string2)
    var3 = var1 | var2
    var4 = {}
    for i in var3:
        var4[i] = 0
        for j in string1:
            if i == j:
                var4[i] += 1
        for j in string2:
            if i == j:
                var4[i] += 1
    return var4

print(foo("hej du", "hej hej hejsan"))
```

- Ange (inte rad för rad utan övergripande) vad funktionen `foo` gör.
- Vad skrivs ut på skärmen när programmet körs?

Uppgift 6 (3p)

Betrakta exemplet nedan där en klass återfinns som representerar bollar (det är valfritt om ni vill att dessa ska vara tvådimensionella eller tredimensionella). Vissa delar har dolts. Ersätt de dolda delarna A, B och C med kod så att när programmet körs så skrivs volymen ut för en boll med radie 1.

```
import math

class Ball:
    def __init__(self, radius):
        A
    def compute_volume(self):
        return B
    def __str__(self):
        return "Bollen har volymen: " + C

boll = Ball(1)
print(boll)
```

Del 2 (denna del består av två uppgifter)

Denna del göres endast om ni har för avsikt att delta i den muntliga tentamen. Om ni inte hinner klart allt, lämna in det ni har gjort.

Uppgift 7 (12p)

I denna uppgift ska du skapa ett krypteringssystem. Uppgiften består av två deluppgifter där den första handlar om att skapa nycklar som sedan används i den andra deluppgiften för att kryptera eller dekryptera meddelanden.

- a) (6p) I denna uppgift ska du skapa en funktion `generate_keys` som genererar krypteringsnycklar. Funktionen tar två argument: en text-sträng `text` och ett heltal `antal`. Argumentet `antal` anger hur många nycklar som ska genereras. Nycklarna representeras med lämplig datatyp. Den viktiga egenskapen för en nyckel är att varje unikt tecken i strängen `text` representeras av ett annat unikt tecken i strängen `text`. Detta används sedan i krypteringen/dekrypteringen i uppgift b) för att byta ut tecken mot andra tecken. För att illustrera detta, om strängen `text` är "annars" så kan en nyckel vara att: a byts mot n, n byts mot s, r byts mot a, s byts mot r. Notera att "annars" bara innehåller fyra unika tecken.

Funktionen `generate_keys` skapar `antal` många slumpmässiga nycklar enligt ovan och returnerar dessa.

- b) (6p) Skapa en funktion `encrypt_decrypt` som tar tre argument. Det första är en sträng `text` som ska krypteras eller dekrypteras, det andra är en lista med nycklar (som skapas genom att anropa `generate_keys` innan), det tredje är av typen `bool` och anger om kryptering eller dekryptering ska ske.

Kryptering sker genom att man byter ut tecknen i `text` ett efter ett (med start från det första tecknet) mot tecken i den första nyckeln i listan av nycklar. Så fort man har kommit upp i det antal tecken som är lika med längden på nyckeln, så byter man nyckel till den andra nyckeln i listan osv. Efter den sista nyckeln så byter man till den första nyckeln igen och börjar om. Dekryptering sker på motsvarande sätt, med skillnaden att man byter tillbaka tecknen igen. Om ett tecken som ska krypteras eller dekrypteras inte återfinns i nycklarna så lämnas tecknet okrypterat.

Uppgift 8 (8p)

I denna uppgift ska ni skapa en klass Kortlek som representerar en kortlek. Det är den [vanligt förekommande kortleken](#) med 52 spelkort (4 färger och 13 olika valörer). Kraven på denna klass är att följande funktionalitet ska ingå:

- Det ska finnas en metod som drar ett slumpmässigt kort från kortleken. Om det inte finns några kort i kortleken så ska detta hanteras på lämpligt sätt.
- Det ska finnas en metod för att lägga till ett kort till kortleken. Om kortet redan finns så ska detta hanteras. Om det man försöker lägga till inte är ett kort så ska även detta hanteras.
- Det ska finnas en metod som samlar in alla kort i kortleken (så att kortleken innehåller alla 52 kort).
- Metoden `__str__` ska implementeras.
- Det ska dessutom finnas en metod i klassen som jämför två kort med två andra kort och avgör vilka två kort som är bäst, eller om de är lika bra.
 - Om de första två korten är ett par (t.ex. par i sjuor), medan de andra två korten inte är ett par (t.ex. en knekt och en nia) så är det första två korten bäst.
 - Om de första två korten är ett par och de andra två korten är ett par och de första två korten har en annan valör än de andra två korten så är de två kort bäst som har högst valör. T.ex. i fallet (knekt, knekt) och (dam, dam), så är (dam, dam) bäst.
 - Om de första två korten inte utgör ett par och de andra två korten inte heller utgör ett par och det finns endast ett kort av alla fyra kort som har högst valör, så är de två kort bäst som innehåller kortet med högst valör. T.ex. om de först korten är (knekt, sju) och de andra två korten är (dam, två) så är de sista två korten bäst eftersom ”dam” är det högsta totalt sätt.
 - Antag att det kortet med högst valör bland de första två korten har samma valör som det kort med högst valör bland de andra två korten och det kortet med lägst valör bland de första två korten inte har samma valör som det kort med lägst valör bland de andra två korten. I detta fall jämför man det kort med näst högst valör för de första två korten med det kort med näst högst valör för de andra två korten. Det kort med högst valör av dessa två är bäst och således är de två korten bäst där detta kort ingår.
 -
 - Om inget av ovanstående fall gäller så är de första två korten lika bra som de andra två korten.