

- Tentan har flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt antal rätta alternativ får man noll poäng på frågan.
 - Man måste bli godkänd på del A (4 rätt på 8 frågor) för att del B ska rättas.
 - Del B består utav 8 frågor med varierande poäng (totalt 12p).
 - Inga externa bibliotek får användas om de inte nämns eller finns med i uppgiften, man får dock använda inbyggda funktioner som `len` och `range`.
 - **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
 - **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.
-

Del A: flervalsfrågor

1. Vad är resultatet av satsen `print([x + 2 for x in range(2)])`?

- A. [2, 3, 4]
- B. [3, 4]
- C. [0, 1, 2]
- D. [2, 3]
- E. Ett särfall

2. Vad skrivs ut om vi kör koden till höger?

- A. 1
- B. 2
- C. 3
- D. 4

```
d = { 'hej' : 'du', (1, 2, 3) : 'a' }  
print(len(d[(1, 2, 3)]))
```

E. Inget av alternativen A.-D. då ett särfall lyfts.

3. Vilka utav följande påståenden är korrekta baserat på information från kodraden till höger?

- A. A, B och C är klasser.
- B. A är en klass, B och C är metoder.
- C. B och C är klasser och A är en metod.
- D. A är en basklass(/superklass) till B och C.
- E. A är en subclass till B och C.

```
class A(B,C):
```

4. I vilken ordning skrivs tuplerna ut av koden till höger?

- A. (1,a), (1,b), (2,a), (2,b)
- B. (1,b), (1,a), (2,b), (2,a)
- C. (2,a), (2,b), (1,a), (1,b)
- D. (2,b), (2,a), (1,b), (1,a)

```
for x in [2, 1]:  
    for y in ['b', 'a']:  
        print((x,y))
```

5. Hur ska man skriva för att komma åt strängen `secret` i `d`?

- A. `d[1]['pw'][2,0]`
- B. `d[2]['pw'][3,1]`
- C. `d['pw'][2][0]`
- D. `d['pw'][2,0]`
- E. `d['pw'][3][1]`
- F. Går ej.

```
d = {"hej": 2, "pw": [8,93,["secret",1]]}
```

6. Vad blir det för utskrift av koden till höger?

- A. 6, 1, -15
- B. 15, 10, -15
- C. 5, 10, -15
- D. 15, 1, -15
- E. Ingenting, det blir ett särfall.

```
x=1
def f(x, y):
    y = 5
    return x + y
x=10
y=-15
print(f(0,0), x, y)
```

7. Vilka tre värden skrivs ut av koden till höger? (ignorera radbyte med print)

- A. False; True; True
- B. False; True; False
- C. True; False; True
- D. True; True; False

```
x, y = True, False
print((x and y) or (not x and not y))
print((x or y) and (not x or not y))
print(not(x and not (x and y)))
```

8. Vad är värdet på `zs` efter att vi kört koden till höger?

- A. `[-1, 4, 3]`
- B. `[1, 4, 5]`
- C. `[-1, -4, -5]`
- D. `[4, 4, 4]`
- E. Inget av alternativen A.-D. då ett särfall lyfts.

```
xs = [1, 4, 3]
ys = [-1, 4, 5]
zs = []

for i in range(3):
    x = xs[i]
    y = ys[i]
    if x <= y:
        zs.append(x)
    else:
        zs.append(y)
```

Del B: kodfrågor

9. (1p) Skriv en ekvivalent version av funktionen nedan som använder `for` istället för `while`.

```
def f(s):
    i = 0
    s_list = []
    while i < len(s):
        s_list.append(s[i])
        i = i + 1
    return s_list
```

10. (2p) Skriv en funktion `my_div_calc` som tar en lista `l` med element och ett heltal `x`, och returnerar hur många tal i `l` som är jämnt delbara med `x`. Ditt program ska kunna hantera listor som innehåller element som är av godtycklig datatyp (inte bara heltal) och ska då inte krascha utan bara gå vidare till nästa element. Exempel

```
l = [15, [3], 'hej', 9, 13, (6, 18)] # endast element 15 och 9 ar delbara med 3
x = 3
print(my_div_calc(l, x))
2
```

11. (1p) Nedan kod är ett försök att skriva en rekursiv funktion för att summera alla tal i listan. Den innehåller buggar på två rader. Fixa dessa

```
def add_up(lis):
    if len(lis)==0:
        return lis.pop()
    else:
        elem = lis.pop()
        return add_up(lis)
```

12. (2p) Betrakta funktionen nedan. Vad gör funktionen och vad händer om `z1` och `z2` har olika längd? (0.5p). Skriv en ekvivalent version av funktionen som ej använder `map`, `sum` och `lambda`. (1.5p)

```
def f(z1, z2):
    return sum(map(lambda x, y: x*x + y, z1, z2))
```

13. (1p)

Betrakta klassen `Circle` till höger för att representera cirklar i planet. Skriv kod som: (1) skapar två cirkelobject `c1` och `c2` och (2) beräknar om `c1` har mindre area än `c2` genom att använda metoden `__lt__`.

```
class Circle:
    def __init__(self, x, y, r):
        self.x = x
        self.y = y
        self.r = r
    def __lt__(self, other):
        """
        Compares area by
        comparing radius.
        """
        return self.r < other.r
```

14. (1p) Lägg till metoden `__eq__` till klassen `Circle` som tittar om två cirklar är identiska (samma koordinater och radie).
15. (2p) Utöka klassen i föregående uppgift genom att implementera en metod i `Circle` som testar om cirkeln omges av en annan cirkel. Visa hur man kallar på metoden för att göra jämförelsen.
16. (2p) Skriv en funktion som tar två strängar `w` och `T`, och returnerar `True` om strängen `w` är representerad i strängen `T`, annars `False`. Att strängen `w` är representerad i strängen `T` definieras i denna uppgift som att alla bokstäver i `w` finns och förekommer i ordning i `T`. Skilj på stora och små bokstäver. Exempel på nästa sida →

```
print(check_presence('hej', 'hej pa dig'))
True
print(check_presence('hej', 'Hej pa dig')) # stor bokstav ej lika
False
print(check_presence('heeeeeej', 'hej pa dig')) # Bokstaver i T kan ateranvandas.
True
print(check_presence('titta', 'tag inte Tottes leksak'))
True
```