

## Errata för lösningar

### Lösning 9.1b (korrigerade rader i fetstil):

I denna lösning gjordes ett klassiskt misstag genom att utelämna två stycken "-1" i koden vilket kommer att ge upphov till felmeddelanden om man inte har med dem.

```
import random

avg = 0
count = 0
i = 1
total = 0
temp = 0
while i < len(lista) :
    count = count + 1
    lista[i] = random.randint(1, 100)
    total = total + lista[i]
    i = i + 1
avg = total / count
i = 0
for _ in range(len(lista)-1) :
    while i < len(lista)-1 :
        if lista[i] > lista[i+1] :
            temp = lista[i+1]
            lista[i+1] = lista[i]
            lista[i] = temp
        i = i + 1
i = 0
```

### Lösning 9.8 (korrigerade rader i fetstil):

I denna lösning är det indenteringen som blivit felaktig vid typsättningen. De markerade raderna skall flyttas lite till höger jämfört med i boken.

```
def sortera(L) :
    length = len(L)
    for i in range(length-1) :
        for j in range(i+1, length) :
            if L[j] < L[i] :
                temp = L[i]
                L[i] = L[j]
                L[j] = temp
    return L
```

### Lösning 9.9 (korrigerade rader i fetstil)

I denna lösning, som bygger vidare på ett exempel från tidigare i boken, hängde det gamla namnet på listan ('tal') med. Rätt version ser ut som nedan.

```
def median(L) :
    L = sortera(L)
    length = len(L)
    if length % 2 == 1 :
        return L[length // 2] # Udda antal element
    else :
        return (L[length // 2] + L[length // 2 - 1]) / 2
```

### Lösning 9.10 (korrigerade rader i fetstil)

Även i denna lösning var det ett "-1" som saknades.

```
def finns(L, x) :
    lower = 0
    upper = len(L)
    while (upper > lower+1) :
        element = lower + ((upper - lower) // 2)
        if L[element] == x :
            return True
        elif L[element] > x :
            upper = element
        else :
            lower = element
    return L[upper-1] == x or L[lower] == x
```

### Lösning 9.12 (korrigerade rader i fetstil)

En rad skall flyttas lite åt höger.

```
def isPrime(n) :
    if n > 1 :
        for i in range(2, int(n**0.5)+1) :
            if (n % i) == 0 :
                return False
        return True
    else :
        return False
```

## Lösning 9.14 (korrigerade rader i fetstil)

En rad skall flyttas lite åt höger.

```
import random

# ord.txt är en textfil med ord som kan väljas
# läs in denna till listan orden
with open('ord.txt') as f:
    orden = [l.rstrip() for l in f.readlines()]

# Välj ett slumpvis ord ur listan orden
ordet = orden[random.randint(0, len(orden))]

# Skapa en sträng av ?, så många som det finns bokstäver i ordet
visa = '?' * len(ordet)

# Skapa en lista som lagrar de gissade bokstäverna, tom vid start
gissat = [] # Gissade bokstäver

print('Hänga gubbe!')
while visa != ordet : # Fortsätt tills hela ordet har gissats
    gissade = ''.join(gissat) # Skapa sträng med gissade bokstäver
    print(f'{visa} Gissat: {gissade}')

    b = input('Gissa en bokstav :')
    if b in ordet : # Om den gissade bokstaven finns i ordet...
        visa = list(visa) # ...ändra typ till lista, enklast så

        # Gå igenom ordet, en bokstav i taget, och jämför med gissningen
        for i in range(len(ordet)) :
            if ordet[i] == b : # Om gissningen finns i ordet på plats i...
                visa[i] = b # ...ändra ? till den gissade bokstaven

        # Då alla bokstäver jämförts, ändra tillbaka visa till sträng
        visa = ''.join(visa)

    gissat = gissat + [b] # Kom ihåg gissningen

# Programmet avslutas när man har lyckats hitta ordet
print('Du gissade rätt:', ordet)
```