



Alle teorispørsmål skal besvares og begrunnes. Alle oppgavene skal demonstreres til en studentassistent på sal. I oppgaver der du skriver programkode skal også denne vises fram. Lykke til!

1 Sparing

Kodesnutt 1

```
def calc_v():  
a)   v = 1000 * (1 + (0.05 / 12.0))**(12 * 20)  
      return v
```

Kodesnutt 2

```
def calc_v_arg(A, r, k):  
b)   v = A * (1 + (r / 12.0))**(12 * k)  
      return v
```

Kodesnutt 3

```
def calc_v_values(A, r, k):  
    vlist = []  
    for i in xrange(k):  
c)     vlist.append(A * (1 + (r / 12.0))**(12 * (i+1)))  
  
    return vlist
```

Kodesnutt 4

```
print "Angi investeringssum: ",  
A = input()  
  
print "Angi rente: ",  
r = input()  
d)   print "Angi antall sparingsaar: ",  
      k = input()  
  
print calc_v_values(A, r, k)
```

e) Renten er større enn null så vi trenger ikke forsøke med en høyere verdi enn v. Negative summer gir heller ingen mening så vi kan starte med 0.

Kodesnutt 5

```
def find_a_for_v(v):  
    for i in range(v):  
        if calc_v_arg(i) >= v:  
            return i
```

2 Ωsteviruset

Kodesnutt 6

```
cheeses = {  
    'cheddar':  
        ('A235-4', 'A236-1', 'A236-2', 'A236-3', 'A236-5', 'C31-1', 'C31-2'),  
    'mozzarella':  
        ('Q456-9', 'Q456-8', 'A234-5', 'Q457-1', 'Q457-2'),  
    'gombost':  
        ('ZLAFS55-4', 'ZLAFS55-9', 'GOMBOS-7', 'A236-4'),  
    'geitost':  
        ('SPAZ-1', 'SPAZ-3', 'EMACS45-0'),  
    'port salut':  
        ('B15-1', 'B15-2', 'B15-3', 'B15-4', 'B16-1', 'B16-2', 'B16-4'),  
    'camembert':  
        ('A243-1', 'A234-2', 'A234-3', 'A234-4', 'A235-1', 'A235-2', 'A235-3'),  
    'ridder':  
        ('GOMBOS-4', 'B16-3'),  
}  
  
print(cheeses['port salut'])  
print()  
  
for cheesename, cheese in cheeses.items():  
    for row in cheese:  
        prefix = row.split("-")[0]  
        if prefix in ('A234', 'A235', 'B13', 'B14', 'B15', 'C31'):  
            print(row, cheesename, "VIRUS DETECTED")  
  
print()  
  
for cheesename, cheese in cheeses.items():  
    for row in cheese:  
        prefix = row.split("-")[0]  
        if prefix not in ('A234', 'A235', 'B13', 'B14', 'B15', 'C31'):  
            print(row, cheesename)
```

3 Bursdagsdatabasen**Kodesnutt 7**

```
# encoding: utf-8
birthdays = {
    "22 nov": ["Lars", "Mathias"],
    "10 des": "Elle",
    "30 okt": ["Veronica", "Rune"],
    "12 jan": "Silje",
    "31 okt": "Willy",
    "8 jul": ["Brage", "Øystein"],
    "1 mar": "Nina"
}

def add_birthday_to_date(date, name):
    try:
        birthdays[date].append(name)
    except KeyError:
        birthdays[date] = name
    except AttributeError:
        birthdays[date] = [birthdays[date][0], name]
```

4 Tallforekomster i fil

a) Dette kan gjøres på flere måter:

Kodesnutt 8: Alternativ 1

```
def number_of_lines(filename):
    fp = open(filename, "r")
    i = 0
    for c in fp.read():
        if c == "\n": i += 1

    return i
```

Kodesnutt 9: Alternativ 2

```
def number_of_lines2(filename):
    fp = open(filename)
    i = 0
    for line in fp.readlines():
        i+=1
    return i
```

Kodesnutt 10: Alternativ 3

```
def number_of_lines3(filename):
    return len(open(filename).readlines())
```

Kodesnutt 11

```
def number_frequency(filename):  
    freq = {}  
  
    fp = open(filename)  
b)  for line in fp.readlines():  
        number = int(line.strip())  
        freq[number] = freq.get(number, 0) + 1  
  
    return freq
```

Kodesnutt 12

```
c) for num, value in number_frequency("nummer.txt").items():  
    print("%d: %d" % (num, value))
```

5 (frivillig) Opptaksgrenser**Kodesnutt 13**

```
poeng_2011 = open("../poenggrenser_2011.csv").readlines()

def antall_studier_alle(grenser):
    i = 0
    for linje in grenser:
        grense = linje.strip().split(",")[1]

        if grense == '"Alle"':
            i += 1

    return i

def gjennomsnitt_opptak(grenser):
    n = 0
    avg = 0

    for linje in grenser:
        grense = linje.strip().split(",")[1]

        if grense == '"Alle"':
            continue

        n += 1
        avg += float(grense)

    avg /= n
    return avg

def laveste_opptaksgrense(grenser):
    laveste_studie = None
    laveste_grense = float('inf')

    for linje in grenser:
        studie, grense = linje.strip().split(",")
        studie = studie.strip('\"')

        if grense == '"Alle"':
            continue

        grense = float(grense)
        if grense < laveste_grense:
            laveste_grense = grense
            laveste_studie = studie

    return laveste_studie

print(antall_studier_alle(poeng_2011))
print(gjennomsnitt_opptak(poeng_2011))
print(laveste_opptaksgrense(poeng_2011))
```
