

# Eksamensoppgaver



Går gjennom noen eksamensoppgaver, både kodeforståelse og koding.

# Eksamensoppgaver



Går gjennom noen eksamensoppgaver, både kodeforståelse og koding.

Fokus på forståelse, visualisering og "tankeprosessen".

# Kodeforståelse



Viktig å lese oppgaveteksten.

Ofte er det spørsmål om både hva som returneres for en bestemt input-verdi, i tillegg til å beskrive funksjonen med en setning.

# Kodeforståelse



Viktig å lese oppgaveteksten.

Ofte er det spørsmål om både hva som returneres for en bestemt input-verdi, i tillegg til å beskrive funksjonen med en setning.

Det vil *ikke* si at vi skriver hva som skjer, men heller hva som er poenget med funksjonen.

# Kodeforståelse



**Feil svar:** Funksjonen setter  $r = 1$ , og så lenge  $x$  er større eller lik 1 ganges  $r$  med  $x$ , og  $x$  minker med 1.

```
def f(x):  
    r = 1  
    while x >= 1:  
        r *= x  
        x -= 1  
    return r
```

# Kodeforståelse



**Riktig(ere) svar:** Funksjonen beregner verdien  $x!$ ,  $x$  fakultet.

```
def f(x):  
    r = 1  
    while x >= 1:  
        r *= x  
        x -= 1  
    return r
```

## Kodeforståelse 2016 2d



Hva blir skrevet ut på skjermen hvis koden vist under blir kjørt?  
Forklar med en setning hva funksjonen  $f$  gjør

```
def f(x):  
    y = 0  
    while x > 0:  
        y = y + x % 10  
        x = int( x / 10 )  
    if y >= 10:  
        y = f( y )  
    return y  
  
print( f(32145) )
```

## Kodeforståelse 2016 kont 4b



6

$f$  regner ut den rekursive tverrsummen tallet som den får inn som parameter, dvs. fortsetter å ta tverrsum helt til vi får et ensifret tall. Tverrsummen av 32145 blir 15, tverrsummen av 15 blir deretter 6.



## Kodeforståelse 2016 kont 4b



Hva returneres hvis man kaller  $f([[3, 5], [2, 4], [1, 3]])$  med koden nedenfor?

Forklar med en setning hva funksjonen  $f()$  gjør?

```
def f(b):  
    c=len(b[0])  
    d=len(b)  
    g = [[0 for row in range(d)]  
          for col in range(c)]  
    for e in range(0, c):  
        for f in range(0, d):  
            g[e][f] = b[f][e]  
    return g
```

## Kodeforståelse 2016 kont 4b



[[3, 2, 1], [5, 4, 3]]

Evt. ingenting ettersom det er feil med innrykk i oppgaveteksten.

Begge svar gir full pott!

Funksjonen transponerer en matrise.

## Kodeforståelse 2017 2a

Funksjonen *bin\_search* er ment til å skulle utføre binærsøk, men resulterer i feilmeldingen *"IndexError: list index out of range"*.

I hvilken linje er feilen?

Hva skulle egentlig stått på den linjen for at funksjonen skal virke etter sin hensikt?

```
def bin_search(liste, verdi, imin, imax):  
    if (imax < imin):  
        return False  
    else:  
        imid = (imin + imax)  
        if verdi < liste[imid]:  
            return bin_search(liste, verdi, imin, imid-1)  
        elif verdi > liste[imid]:  
            return bin_search(liste, verdi, imid+1, imax)  
        else:  
            return imid
```

## Kodeforståelse 2017 2a



Linje 5

$imid = (imid + imax)/2$

## Kodeforståelse 2018 sett 1 2a



Hva returneres ved funksjonskallet under?

*myst(((True and False) or (False and True)), ((False or True) and (not(not True))))*

```
def myst(val1, val2):  
    if (val1 and val2):  
        return 1  
    elif (val1 and not val2):  
        return 2  
    elif (not val1 and val2):  
        return 3  
    else:  
        return 4
```



## Kodeforståelse 2018 kont 2e og f



Hva blir skrevet ut til skjerm når du kjører programmet vist under?

$z = ((2, 0, 11, 8, 5), (14, 17, 13, 8, 0))$  *print(myst3(z))*

Beskriv med en setning hva koden i oppgave 2e gjør.

```
def myst3(a):  
    s = ''  
    for r in a:  
        for c in r:  
            s += chr(ord('A') + c)  
    return s
```

## Kodeforståelse 2018 kont 2e og f



**CALIFORNIA** Store bokstaver (pga. `ord('A')`) og ingen fnutter (pga. `printing`)

Henter ut ett og ett tall fra et tuppel av tupler, og lager en sammenhengende tekst av store bokstaver hvor `c` angir hvor langt unna hver bokstav er fra `A` i alfabetet (ASCII- tabellen).



## Programmering 2016 kont 2



Du kan anta at alle funksjonene mottar gyldige argumenter (inn-verdier). Du kan benytte deg av funksjoner fra deloppgaver selv om du ikke har løst deloppgaven.

I denne oppgaven skal man lage funksjoner for å lese en tekstfil med binærkode og gjøre om dette til tegn kodet i eget kodesett for tegn og bokstaver og lagre det til en tekstfil.

## Programmering 2016 kont 2a



Lag funksjonen *load<sub>b</sub>in* som har en inn-parameter *filename*, som er navnet på fila som skal lastes inn. Funksjonen skal lese inn alt innholdet i fila og returnere innholdet som en tekststreng uten linjeskift eller mellomrom. Fila som det leses fra er en tekstfil bestående av binære tall (0 og 1). Hvis fila ikke eksisterer eller ikke kan åpnes, skal funksjonen returnere en tom streng samt skrive ut følgende feilmelding til skjerm: "Error: Could not open file <filename>", der <filename> er navnet på fila.

# Programming 2016 kont 2a



```
def load_bin(filename):  
    binstring = ""  
    try:  
        f = open(filename, "r")  
  
        for line in f:  
            binstring += line.strip()  
  
        f.close()  
    except:  
        print("Error: Could not open file", filename)  
    return binstring
```

## Programmering 2016 kont 2b



Lag funksjonen *bin<sub>t</sub>o<sub>d</sub>ec* som har en inn-parameter *binary*, som er en tekststreng av ukjent størrelse bestående av binære tall (tekststreng med nuller og enere). Funksjonen skal returnere et heltall (dvs. i titalssystemet) som tilsvarer det binære tallet angitt med tekststrengen *binary*. Oppgaven **skal ikke løses** ved hjelp av innebygde funksjoner for å oversette binærtall til heltall.

## Programmering 2016 kont 2b



```
def bin_to_dec(binary):  
    decimal = 0  
    reversed_binary = binary[::-1]  
    for i in range(len(reversed_binary)):  
        decimal += int(reversed_binary[i]) * 2**i  
    return decimal
```

## Programmering 2016 kont 2c

Lag funksjonen *dec\_toc* som har en inn-parameter *dec*, som er et heltall med verdi mellom 0 og 31. Funksjonen skal returnere et tegn eller en bokstav avhengig av verdien av *dec*.

- Hvis *dec* har verdien 0 skal tegnet for " " (mellomrom) returneres
- Hvis *dec* har verdien 1 skal tegnet for "," (komma) returneres
- Hvis *dec* har verdien 2 skal tegnet for "." (punktum) returneres
- Hvis *dec* har en verdi mellom 3 og 31 skal en stor bokstav i det norske alfabetet returneres, der *dec* = 3 gir bokstaven "A", *dec* = 4 gir bokstaven "B", helt opp til *dec* = 31 som gir bokstaven "Å".
- For alle andre verdier av *dec* skal funksjonen returnere en tom streng.



Antar denne konstanten eksisterer.

*DTC = ",.ABCDEFGHIJKLMNOPQRSTUVWXYZ"*

```
def dec_to_char(dec):  
    if dec < len(DTC):  
        return DTC[dec]  
    else:  
        return ""
```



Lag funksjonen *bin\_to\_text* som har en inn-parameter *binstring*, som er en tekststreng av ukjent lengde bestående av binære tall (tekststreng med nuller og enere). Funksjonen skal returnere en tekststreng bestående av bokstaver og tegn som er kodet i henhold til oppgave 2c der hvert tegn er representert med 5 bit. Inn-parameteren *binstring* vil alltid være et multiplum av fem siffer.



## Programmering 2016 kont 2d



```
def bin_to_txt(binstring):  
    txt = ""  
    for i in range(0, len(binstring), 5):  
        decimal = bin_to_dec(binstring[i:i+5])  
        txt += dec_to_char(decimal)  
    return txt
```

## Programmering 2016 kont 2e

Lag funksjonen *main*, uten parametere. Funksjonen skal gjøre følgende:

1. Skrive ut teksten "Binary-to-text converter" til skjerm
2. Spørre brukeren om navn på fil det skal lastes fra (tekstfil som inneholder binære tall) og ta vare på filnavnet i variabelen *b\_file*.
3. Oversette tekststrengen av binære tall til tekst med bokstaver og tegn og lagre innholdet i variabelen *txt*.
4. Spørre brukeren om navnet på fil som resultatet skal lagres til og ta vare på filnavnet i variabelen *t\_file*.
5. Skrive innholdet av variabelen *txt* til fila med filnavn angitt i variabelen *t\_file*.
6. Skrive ut til brukeren at: "*b\_file* has been converted and saved to *t\_file*".

Hvis funksjonen får problemer med å skrive til fila, skal følgende feilmelding skrives: "Error: Could not write to file *t\_file*".

## Programming 2016 kont 2e

```
def main():
    print("Binary-to-text converter")
    b_file      = input("Name of binary file to load from: ")
    b_string    = load_bin(b_file)
    txt         = bin_to_txt(b_string)
    t_file      = input("Name of text file to save to: ")
    try:
        f = open(t_file, "w")
        f.write(txt)
        f.close()
        print(b_file, "has been converted and saved to",
              t_file)
    except:
        print("Error: Could not write to file", t_file)
```



Lag funksjonen *yatzy*. Den skal ha 5 innparametere, kalt *t1*, *t2*, *t3*, *t4* og *t5*. Innparametrene representerer 5 tall mellom 1 og 6 (5 terninger). Funksjonen skal returnere ei liste som inneholder de 5 tallene i sortert rekkefølge, eller en feilmelding hvis en av tallene er større enn 6 eller mindre enn 1.



```
def yatzy(t1, t2, t3, t4, t5):  
    l = [t1, t2, t3, t4, t5]  
    for i in range(len(l)):  
        for j in range(len(l) - i - 1):  
            if l[j] > 6 or l[j + 1] > 6:  
                return "Ikke bruk input stoerre enn 6!"  
            if l[j] < 1 or l[j + 1] < 1:  
                return "Ikke bruk input mindre enn 1!"  
  
            if l[j] > l[j + 1]:  
                l[j], l[j + 1] = l[j + 1], l[j]  
  
    return l
```



Lag funksjonen *maxi<sub>y</sub>atzy*. Den skal ta inn en liste med 5 eller 6 tall, og den skal returnere en skriftlig melding til brukeren som sier hvor mange terninger som ble kastet, hvilken verdi det var flest av, og hvor mange like det var av den verdien. Hvis det blir "uavgjort mellom to tall" brukes det høyeste tallet.

## Programmering 2013 2b

```
def maxi_yatzy(lst):  
    n = len(lst)  
    most_val = None  
    most_num = 0  
  
    for base_dice in range(1, 6 + 1):  
        num = 0  
        for dice in lst:  
            if dice == base_dice:  
                num += 1  
        if num >= most_num:  
            most_val = base_dice  
            most_num = num  
    return "Du kastet " + str(n) + " terninger og fikk  
        flest " + str(most_val)  
        + " (" + str(most_num) +  
        " like)."
```