Denne kolonnen er forbeholdt sensor

This column is for external examiner

Oppgave 2

a) JULENISSEN ✓

Funksjonen returnerer en sammenslåing av x og y med vært odde element i x og vært like (partall) element i y.

b) 64 ✓

Funksjonen ganger x med 2 helt til dette produktet blir større enn eller lik 10, og returnerer produktet av alle produktene som var mindre enn 10.

c) [8, 7, 6, 5, 3, 2, 1] ✓

Den sorter listen fra høyest til lavest

Oppgave 3

a)

```
def readTime():
    valid = False
    while not valid:
        h = int(input("Enter hour: "))
        if (h >= 0) and (h < 24):
            valid = True
        else: vali
            print("Hour must be between 0 and 23!")

    valid = False
    while not valid:
        m = int(input("Enter minute: "))
        if (m >= 0) and (m < 60):
            valid = True
        else:
            print("Minute must be between 0 and 59!")

    valid = False
    while not valid:
        s = int(input("Enter second: "))
        if (s >= 0) and (s < 59):
            valid = True
        else:
            print("Second must be between 0 and 59!")
    return [h, m, s]
```

3b)

```
def convertTime(time, mode):
    if mode == "sec":
        return time[0]*3600 + time[1]*60 + time[2]
    elif mode == "time":
        h = int(time/3600)
        time -= h * 3600
        m = int(time/60)
        time -= m*60
        s = time
        return [h, m, s]
```

3c)

```
def travelTime():
    print("Give departure time in hour, minute and second:")
    dep = readTime()
    dep_sec = convertTime(dep, "sec")
    print("Give arrival time in hour, minute and second:")
    while True:
        arr = readTime()
        arr_sec = convertTime(arr, "sec")
        if arr_sec >= dep_sec:
            break
        else:
            print("ERROR: Arrival time must be later", \
                  "than Departure time")
    travel_time = convertTime(arr_sec - dep_sec, "time")
    h, m, s = travel_time[0], travel_time[1], travel_time[2]
    return "travel: {0} hours, {1} min, {2} sec".format(h, m, s)
```


Fagtrykk Trondheim AS 05.2015

3c)

```
def analyzeBusRoutes(BusTable):
    sTime, sBus = -float("inf"), None  # treyeste
    fTime, fBus = float("inf"), None  # raskeste

    for bus in BusTable:

        busDep = [bus[1], bus[2]]
        busArr = [bus[3], bus[4]]

        busDepSec = convertTime(busDep, "sec")
        busArrSec = convertTime(busArr, "sec")

        time = busArrSec - busDepSec

        if time > sTime:

            sTime = time
            sBus = bus[0]

        if time < fTime:

            fTime = time
            fBus = bus[0]

    sTime = convertTime(sTime, "time")
    fTime = convertTime(fTime, "time")

    print("The slowest bus route is bus nr.", sBus, \
          "and it takes", sTime[0], "hour", sTime[1], "min.")

    print("The fastest bus route is bus nr.", fBus, \
          "and it takes", fTime[0], "hour", fTime[1], "min.")
```

Denne kolonnen er forbeholdt sensor

This column is for external examiner

Oppgave 4

a)
```
NTNU_Scores = (89, 77, 65, 41, 0)
NTNU_letters = ('A', 'B', 'C', 'D', 'E', 'F')
TASKS = ('1', '2a', '2b', '2c', '3a', '3b', '3c', '3d',
         '4a', '4b', '4c', '4d', '4e', '4f', '4g', '4h')
WEIGHTS = (25, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
```

b)
```
def makeArray(Numbers, Texts):
    array = []
    for i in range(len(Numbers)):
        array.append([Numbers[i], Text=[i]])
    return array
```

c)
```
def computeScore(Points):
    score = 0
    for i in range(len(Points)):
        score += Points[i]*WEIGHTS[i]/10
    return score
```

d)
```
def score2Letter(scoreSum, limitLetters):
    for i in range(len(limitLetters)):
        if scoreSum >= limitLetters[i][0]:
            grade = limitLetters[i][1]
    return grade
```

4e)

```
def addCandidate(candidateNumber, Scores):
    S = candidateNumber
    total = 0
    for score in Scores:
        S += "\t"+str(score)
        total += score
    S += "%.2f"%(total/len(Scores), 1)) + "\n"

    with open("eksamen.txt", "a"):
        f = open("eksamen.txt", "a")
        f.write(S)
        f.close()
    except::
        f = open("eksamen.txt", "w")
        f.write(S)
        f.close()
```


Fagtrykk Trondheim AS 05.2015

4f)

```
def readResultFile(filename):
    table = []
    with open(filename, "r") as f:
        for line in f:
            table.append(line.strip())
    # har nå fyllt ut table med strings.
    for i in range(len(table)):
        for n in range(18):
            if n <= 17:
                table[i][n] = int(table[i][n])
            else:
                table[i][n] = float(table[i][n])
    return table
```

Denne kolonnen er forbeholdt sensor

This column is for external examiner

4g)

```
def checkResultOK(filename):
    # feilmeldinger
    E1 = "ERROR: candidate {0} scores are not between 0-10!"
    E2 = "ERROR: candidate {0} has wrong control score!"
    E3 = "ERROR: Candidate {0} appears more than once!"
    E_check = True
    table = readResultFile(filename)
    candList = []
    for cand in table:
        score = cand[1:17]
        for i in range(len(cand)):
            if i == 0:
                if cand[0] in candList:
                    print(E3.format(cand[0]))
                    E_check = False
                else:
                    candList.append(cand[0])
            if i == 17:
                if cand[17] != computScore(score):
                    print(E2.format(cand[0]))
                    E_check = False
            else: # sjekker deloppgavene
                for i in range(len(score)):
                    if (score[i] <= WEIGHTS[i]) and (score[i] >= 0):
                        pass
                    else:
                        print(E1.format(cand[0]))
                        E_check = False
    return E_check
```

4h)

```
def listAll(filename, limitLetters):
    table = readResultFile(filename)
    s_table, cand_table = [], []
    for cand in table:
        nr = cand[0]
        score = cand[-1]
        grade = score2Letter(score, limitLetters)
        s = str(nr).ljust(6) + str(score).rjust(5) + grade
            + grade
        s_table.append([nr, s])
        cand_table.append(nr)
    cand_table.sort()
    for nr in cand_table:
        for s in s_table:
            if nr == s[0]:
                print(s)
                break
```

20,625

Svarskjema flervalgsoppgave / Form for Multiple Choice Questions

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgavenr	*A*	*B*	*C*	*D*
1.1		1		
1.2		1		
1.3		1	1	
1.4				1
1.5		1		
1.6	1			
1.7			1	
1.8	1			
1.9		1		
1.10			1	
1.11		1		
1.12				1
1.13	1			
1.14			1	
1.15		1		
1.16	1			
1.17	1			
1.18				1
1.19	1			
1.20	1			

17, 2, 1

X

X

8,25

X