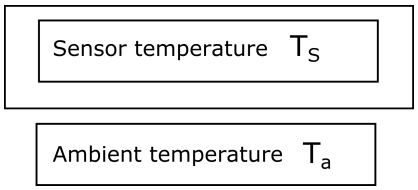
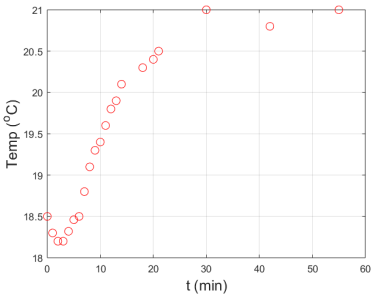
Assignment B Temperature Sensor Model

Imagine you enter a warm room at time $t = 0$, carrying a temperature sensor shown in Figure 2a. You record the output of the sensor starting at time $t = 0$. After some time, you plot your recorded data as in Figure 2b. The data show an initial decrease in the temperature even after you have entered the warm room at time $t = 0$.



(a) Temperature Sensor



(b) Warm room data

At first you are confused by the decrease in the sensor reading after entering the room. But then, your physicist friend explains that the measured ambient temperature T_a includes not only the present temperature, but also all temperatures measured a time t_{lag} before the present moment.

She then suggests to use the following equations to calculate the warm room temperature at all times:

$$T(t) = \begin{cases} T_1(t) = T_{a_1} - (T_{a_1} - T_0) \cdot e^{-r \cdot t} & \text{if } t < t_{lag}, \\ T_2(t) = T_{a_2} - (T_{a_2} - T_1(t_{lag})) \cdot e^{-r \cdot (t - t_{lag})} & \text{if } t \geq t_{lag}. \end{cases}$$

Remark: Note that $T_1(t_{lag})$ appearing in the case $t \geq t_{lag}$ is a function of t_{lag}, just like $T_1(t)$ is a function of t in the case $t < t_{lag}$.

Problemdefinition

Create a function named `Sensor` that takes as input T_{a_1}, T_{a_2}, T_0, r, t_{lag}, t and returns values of the temperature $T(t)$ following the above equations. The values to be used are given in the example below. (*Hint: Input time t and output temperature T are vectors. All other inputs are scalars.*)

Løsningsskabelon

```
function T = Sensor(Ta1,Ta2,T0,r,tlag,t)
% Insert your code
```

```
def Sensor(Ta1,Ta2,T0,r,tlag,t):
  #insert your code
  return T
```

```
Sensor <- function(Ta1,Ta2,T0,r,tlag,t) {
  #insert your code
  return(T)
}
```

Input	
T_{a_1}, T_{a_2}, T_0, r, t_{lag}, t	Ambient and initial temperatures, room property, present and history times.

Output	
`T`	Measured temperature at the present time t.

Eksempel

Consider the following input values:

$$T_{a_1} = 16.7,\ T_{a_2} = 21,\ T_0 = 18.5,\ r = 0.09,\ t_{lag} = 2.5,\ t = \text{a vector of values from 0 to 9}$$

The temperature can then be computed as

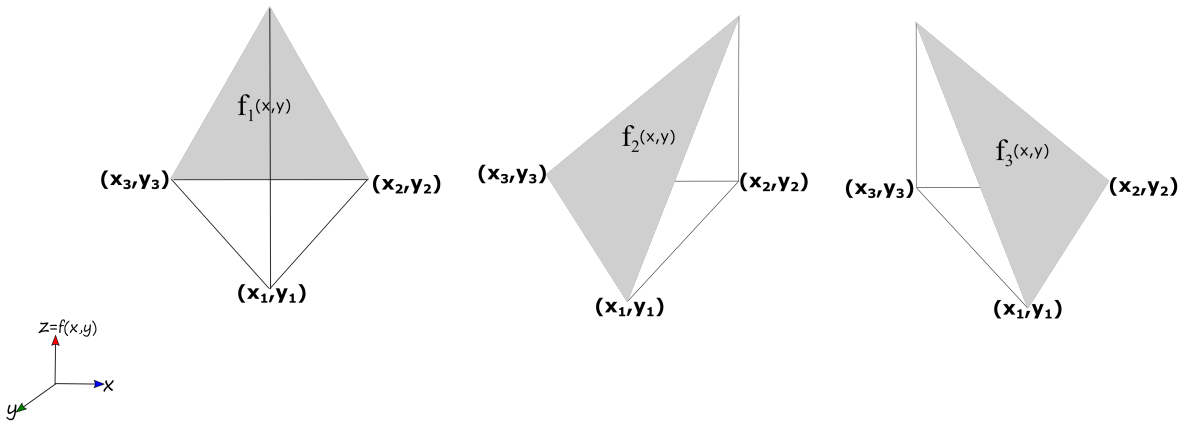
$$T(t) = [18.5000, 18.3451, 18.2035, 18.2633, 18.4988, 18.7141, 18.9109, 19.0907, 19.2550, 19.4052]$$

```
function T = Sensor(Ta1,Ta2,To,r,tlag,t)
    T(t < tlag) = T1(Ta1,Ta2,To,r,t(t < tlag));
    T(t >= tlag) = Ta2 - (Ta2 - T1(Ta1,Ta2,To,r,tlag))*E(r, t(t >= tlag) - tlag);
end

function e = E(r, t)
    e = exp(-r*t);
end

function T = T1(Ta1,Ta2,To,r,t)
    % T = Ta1 - (Ta1 - To)*exp(-r*t);
    T = Ta1 - (Ta1 - To)*E(r,t);
end
```

Assignment C Basis Function



Suppose you want to compute functions $f_1(x, y), f_2(x, y)$ and $f_3(x, y)$ from the above figure, given a set of $\mathbf{x}$ and $\mathbf{y}$ points (x_1, y_1), (x_2, y_2) and (x_3, y_3). The functions are defined as follows:

$$f_i(x, y) = a_i x + b_i y + c_i, \text{ and satisfy } f_i(x_j, y_j) = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases} \tag{1}$$

We would need to determine coefficients a_i, b_i and c_i. For this exercise we ONLY determine the a_i, which are given by

$$a_i = \frac{\widetilde{a}_i}{2\triangle}, \qquad \text{where} \tag{2}$$

$$\widetilde{a}_i = x_j y_k - x_k y_j, \tag{3}$$

for index ordering $(i, j, k) = (1, 2, 3),\ (2, 3, 1),\ (3, 1, 2)$ and

$$\triangle = \frac{1}{2}(x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3). \tag{4}$$

Here, $\triangle$ is the area of the gray triangles in the above figures.

Remark: In this exercise, you only need to calculate a_i from equation (2) using $\widetilde{a}_i$ and $\triangle$ from equations (3) and (4), respectively.

Problemdefinition

Create a function named `BasisFunction` that takes as input the vectors $\mathbf{x} = [x_1, x_2, x_3]$ and $\mathbf{y} = [y_1, y_2, y_3]$. The function must return the vector of coefficients $a = [a_1, a_2, a_3]$.

Løsningsskabelon

```
function a = BasisFunction(x, y)
% Insert your code
```

```
def BasisFunction(x, y):
  #insert your code
  return a
```

```
BasisFunction <- function(x, y){
  #insert your code
  return(a)
}
```

Input	
`x`, `y`	Input coordinates (vectors).

Output	
`a`	First coefficients of basis functions f_1, f_2, f_3.

Eksempel

$$\text{If } x = [2, 5, 7],\ y = [3, 3.5, 2] \text{ then} \tag{5}$$
$$a = [2.6364, -3.0909, 1.4545]$$

C

```
% It would not work on codejudge, but there's probably an error there!

function a = BasisFunction(x, y)
    D2 = x(1)*y(2) - x(2)*y(1) + x(2)*y(3) - x(3)*y(2) + x(3)*y(1) - x(1)*y(3);
    i = 1:3; % i = [1 2 3]
%     j = circshift(i,[0 2]); % j = [2 3 1]
%     k = circshift(i,[0 1]); % k = [3 1 2]
%     a(i) = (x(j).*y(k) - x(k).*y(j) / D2;
    a(i) = (x(circshift(i,[0 2])).*y(circshift(i,[0 1])) - x(circshift(i,[0 1])).*y
(circshift(i,[0 2]))) / D2;
%-------------------------------------------------
% Solution for dummies:
    a(1) = (x(2)*y(3) - x(3)*y(2)) / D2;
    a(2) = (x(3)*y(1) - x(1)*y(3)) / D2;
    a(3) = (x(1)*y(2) - x(2)*y(1)) / D2;
end
```

Assignment D Date Format

You have just received the following message from your friend who needs your urgent assistance to deliver some software in 15 minutes time:

Hej!

Could you please write and send me a program which inputs a date in the format ***dd/mm/yy*** *and outputs it in the format* ***month dd, year****. For example,* ***14/08/92*** *becomes:* ***August 14, 1992****. An elderly client is in desperate need for it and he's willing to pay handsomely!!!*

Problemdefinition

Create a function named `DateFormat` that takes as input a string containing a date of the form dd/mm/yy, and returns the new date format as month dd, year. (*Hint: The date format is in English. For clarity, we focus on the years 1900–1999.*)

Løsningsskabelon

```
function F = DateFormat(dd/mm/yy)
% Insert your code
```

```
def DateFormat(dd/mm/yy):
  #insert your code
  return F
```

```
DateFormat <- function(dd/mm/yy) {
  #insert your code
  return(F)
}
```

Input

`dd/mm/yy` (dd/mm/yy) (string).

Output

`F` The new date format.

Eksempel

Consider the input date `14/08/92`. Then the new date format should read

August 14, 1992.

D

```
function F = DateFormat(date)
    % date = dd/mm/yy
    dd = date(1:2);
    mm = str2num(date(4:5));
    yy = date(7:8);
    M = {'January ', 'February ', 'March ', 'April ', 'May ', 'June ',...
         'July ', 'August ', 'September ', 'October ', 'November ',...
         'December '};
    F = [M{mm} dd ', 19' yy];
end
```

Assignment E Seat Allocation

DTU Airlines (a short-range air flight operator) have just purchased a new aircraft. They have a problem with their software for seat allocation and have sublet your software company with a contract to design a new program. When flyers book their seats, the program must display which seats are available and which are booked.

A booked seat is marked with an X. For example, if seats 1B, 3D and 5A are taken the seat arrangement looks like

1	A	X	C	D
2	A	B	C	D
3	A	B	C	X
4	A	B	C	D
5	X	B	C	D

DTU Airlines have clearly outlined what their software should output. If they input a seat that is already taken, for example seat 1B, the program returns

"Seat 1B is already taken"

The program must also PRINT the seat arrangement, as shown above.
If a seat is not yet taken the program must display

"You are lucky, seat 1B is not yet taken"

and then also PRINT the new seat arrangement with X in place of the input seat.

Remark: Use the given seat pattern above as your initial seat arrangement.

Problemdefinition

Create a function named `SeatAllocation` that takes a string `x` of a seat number as input and returns the seat arrangement as *seats* while you also print on the screen whether a seat is taken or not. (*Hint: The initial seat arrangement can be defined within the function. It is enough if the function processes properly only the first requested seat allocation.*)

Løsningsskabelon

```
function seats = SeatAllocation(x)
% Insert your code
```

```
def SeatAllocation(x):
  #insert your code
  return seats
```

```
SeatAllocation <- function(x) {
  #insert your code
  return(seats)
}
```

Input

`x`	seat number (string).

Output

`seats`	The new seat arrangement.

Eksempel

Consider the input seat `3C`. Then the output should read

"You are lucky, seat 3C is not yet taken"

1	A	X	C	D
2	A	B	C	D
3	A	B	X	X
4	A	B	C	D
5	X	B	C	D

```
function seats = SeatAllocation(x)
    S(:,1) = ['1' '2' '3' '4' '5']';
    S(:,2) = 'A';
    S(:,3) = 'B';
    S(:,4) = 'C';
    S(:,5) = 'D';
    S([10 11 23]) = 'X';
    i = str2num(x(1));
    j = int8(x(2) - 64);
    if (S(i,j) == 'X')
        fprintf('Seat %s is already taken.\n', [x(1) x(2)]);
        for k = 1:5
            fprintf('%s\n', S(k,:));
        end
    else
        fprintf('You are lucky, seat %s is not yet taken\n', [x(1) x(2)]);
        S(i,j) = 'X';
        for k = 1:5
            fprintf('%s\n', S(k,:));
        end
    end
end
```