# Technical University of Denmark

| | |
|---|---|
| Course name | Introduction to programming and data processing |
| Course number | 02633 |
| Aids allowed | All Aids |
| Exam duration | 2 hours |
| Weighting | All exercises have equal weight |

## Contents



## Submission details

You must hand in your solution electronically:

1. You can upload your solutions individually on CodeJudge (`dtu.codejudge.net/prog-jan17/assignment`) under *Afleveringer/Exam*. When you hand in a solution on CodeJudge, the test example given in the assignment description will be run on your solution. If your solution passes this single test, it will appear as *Submitted*. This means that your solution passes on this single test example. You can upload to CodeJudge as many times as you like during the exam.

2. You must upload your solutions on CampusNet. Each assignment must be uploaded as one separate .py file, given the same name as the function in the assignment:

   (a) `trianglearea.py`
   (b) `polygonarea.py`
   (c) `fitpolynomial.py`
   (d) `predictability.py`
   (e) `syllables.py`

   The files must be handed in separately (*not* as a zip-file) and must have these exact filenames.

After the exam, your solutions will be automatically evaluated on CodeJudge on a range of different tests, to check that they work correctly in general. The assessment of you solution is based only on how many of the automated tests it passes.

- Make sure that your code follows the specifications exactly.
- Each solution shall not contain any additional code beyond the specified function.
- Remember, you can check if your solutions follow the specifications by uploading them to CodeJudge.
- Note that all vectors and matrices used as input or output must be numpy arrays.

## Assignment A Heron's formula

Given the three side lengths ($a$, $b$, and $c$) in a triangle, the area can be computed using Heron's formula,

$$A = \sqrt{p(p-a)(p-b)(p-c)},$$

where

$$p = \frac{1}{2}(a+b+c).$$

The formula is only valid when the side lengths constitute a possible triangle, i.e. when the three following inequalities are satisified:

$$a+b>c, \quad a+c>b, \quad b+c>a.$$

### Problem definition

Create a function named `trianglearea` that takes as input the three side lengths of a triangle and computes the area using Heron's formula. If the given side lengths do not constitute a possible triangle, the function must return zero.

### Solution template

```
def trianglearea(a, b, c):
  #insert your code
  return A
```

| Input | |
|---|---|
| `a`, `b`, `c` | Side lengths (positive decimal numbers). |
| **Output** | |
| `A` | Area of triangle. |

### Example

Consider the following input, $a = 4.5$, $b = 6$, and $c = 7.5$. These side lengths satisfy all three inequality, so they constitute a possible triangle. The area is then computed as

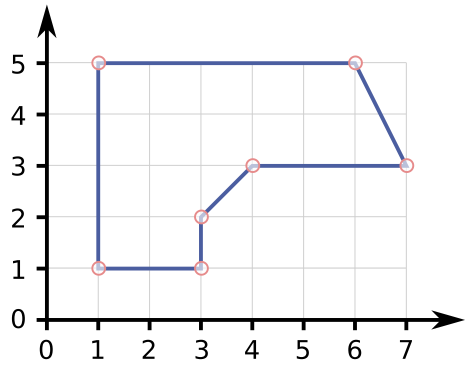
$$p = \frac{1}{2}(4.5+6+7.5) = 9$$

$$A = \sqrt{9(9-4.5)(9-6)(9-7.5)} = 13.5$$

A

## Assignment B Area of a polygon

A polygon can be specified by a list of vertex coordinates $\{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n)\}$. For example, the polygon shown in the figure below can be specified by the coordinates $\{(1,1),(3,1),(3,2),(4,3),(7,3),(6,5),(1,5)\}$.




Based on the coordinates, the area of the polygon can be computed using the following formula

$$A = \frac{1}{2}\left( \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_4 \\ y_3 & y_4 \end{vmatrix} + \cdots + \begin{vmatrix} x_n & x_1 \\ y_n & y_1 \end{vmatrix} \right),$$

where $\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = (x_1 y_2 - y_1 x_2)$ denotes the determinant of the given $2 \times 2$ matrix. Note that the formula for the area contains one determinant term for each edge in the polygon.

Remark: The formula is valid when the polygon does not intersect itself and the coordinates are specified in counter-clock-wise order. In this exercise, we assume these conditions are satisfied.

### Problem definition

Create a function named `polygonarea` that takes as input two vectors containing the x- and y-coordinates respectively, and returns the area of the polygon.

### Solution template

```
def polygonarea(x, y):
  #insert your code
  return A
```

| Input | |
|---|---|
| `x, y` | X- and y-coordinates specifying a polygon (vectors of decimal numbers). |

| Output | |
|---|---|
| `A` | Area of the polygon. |

### Example

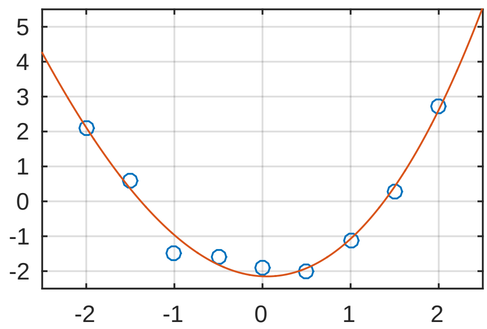
Consider the following input coordinates, which coorespond to the polygon in the figure.

$$x = [1,\ 3,\ 3,\ 4,\ 7,\ 6,\ 1], \quad y = [1,\ 1,\ 2,\ 3,\ 3,\ 5,\ 5]$$

The area can the be computed as

$$\begin{aligned} A &= \frac{1}{2}\left( \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_4 \\ y_3 & y_4 \end{vmatrix} + \begin{vmatrix} x_4 & x_5 \\ y_4 & y_5 \end{vmatrix} + \begin{vmatrix} x_5 & x_6 \\ y_5 & y_6 \end{vmatrix} + \begin{vmatrix} x_6 & x_7 \\ y_6 & y_7 \end{vmatrix} + \begin{vmatrix} x_7 & x_1 \\ y_7 & y_1 \end{vmatrix} \right) \\ &= \frac{1}{2}\left( \begin{vmatrix} 1 & 3 \\ 1 & 1 \end{vmatrix} + \begin{vmatrix} 3 & 3 \\ 1 & 2 \end{vmatrix} + \begin{vmatrix} 3 & 4 \\ 2 & 3 \end{vmatrix} + \begin{vmatrix} 4 & 7 \\ 3 & 3 \end{vmatrix} + \begin{vmatrix} 7 & 6 \\ 3 & 5 \end{vmatrix} + \begin{vmatrix} 6 & 1 \\ 5 & 5 \end{vmatrix} + \begin{vmatrix} 1 & 1 \\ 5 & 1 \end{vmatrix} \right) \\ &= \frac{1}{2}\Big((-2) + 3 + 1 + (-9) + 17 + 25 + (-4)\Big) = 15.5 \end{aligned}$$

## Assignment C  Fitting a polynomial




You are given a set of x and y points $(x_1, y_1),\ (x_2, y_2), \ldots,\ (x_k, y_k)$ and you wish to find a polynomial of order $n$ that fits the data points as well as possible. You must determine the coefficients $a_0, \ldots, a_n$ in the polynomial $f_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ using the least squares approach: First, construct the $k \times (n+1)$ matrix $W$,

$$W = \begin{bmatrix} 1 & x_1^1 & \ldots & x_1^n \\ 1 & x_2^1 & \ldots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_k^1 & \ldots & x_k^n \end{bmatrix}.$$

Then, solve for the coefficients using the following formula, where the vector $y = [y_1,\ y_2, \ldots, y_k]^\top$,

$$a = (W^\top W)^{-1} W^\top y.$$

The result is the vector $a$ which has $n+1$ elements.

### Problem definition

Create a function named `fitpolynomial` that takes as input two vectors of length $k$ containing the x and y points, as well as the order $n$ of the polynomial. The function must return the vector of coefficients $a$.

### Solution template

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

| | |
|---|---|
| **Input** | |
| `x, y` | Input and output points (vectors of decimal numbers, length $k$). |
| `n` | Order of the polynomial (positive whole number). |
| **Output** | |
| `a` | Polynomial coefficients (vector of decimal numbers). |

### Example

Consider the following input (also illustrated in the figure), which yields the following matrix $W$.

$$x = [-2,\ -1.5,\ -1,\ -0.5,\ 0,\ 0.5,\ 1,\ 1.5,\ 2]^\top,$$
$$y = [2.1,\ 0.6,\ -1.5,\ -1.6,\ -1.9,\ -2,\ -1.1,\ 0.3,\ 2.7]^\top,$$
$$n = 3.$$

$$W = \begin{bmatrix} 1 & -2 & 4 & -8 \\ 1 & -1.5 & 2.25 & -3.375 \\ 1 & -1 & 1 & -1 \\ 1 & -0.5 & 0.25 & -0.125 \\ 1 & 0 & 0 & 0 \\ 1 & 0.5 & 0.25 & 0.125 \\ 1 & 1 & 1 & 1 \\ 1 & 1.5 & 2.25 & 3.375 \\ 1 & 2 & 4 & 8 \end{bmatrix}$$

The coefficients computed using the formula yields the result $a \approx [-2.1476,\ -0.1135,\ 1.1286,\ 0.0599]^\top$.

## Assignment D Predictability of a number sequence

A sequence of numbers can be more or less predictable, in the sense that it can be easy or difficult to guess the next number in the sequence given the previous numbers. Here, we consider a sequence of length $N$ that contains only integers between 1 and 5 (both included), and we define the predictability of the sequence according to the following algorithm.

- You are given a sequence of numbers, $[x_1,\ x_2, \dots,\ x_N]$
- Compute a vector $q$ (with 5 elements) where element $q[i]$ is the number of times the integer $i$ occurs in the sequence.
- Compute a matrix $R$ (of dimension $5 \times 5$) where element $R[i, j]$ is the number of times the integer $j$ immediately follows after $i$ in the sequence.
- Compute the predictability $P$ according to the following formula

$$P = \frac{1}{N}\left(\frac{q[x_1]}{N} + \sum_{n=2}^{N} \frac{R[x_{n-1}, x_n]}{q[x_{n-1}]}\right).$$

### Problem definition

Create a function named `predictability` that takes a sequence `x` of numbers as input and returns the predictability $P$ computed using the algorithm above.

### Solution template

```python
def predictability(x):
  #insert your code
  return P
```

| Input | |
|---|---|
| `x` | Number sequence (vector of whole numbers between 1 and 5). |
| **Output** | |
| `P` | Predictability (decimal number). |

### Example

Consider the following sequence of numbers of length $N = 15$

$$x = [1,\ 5,\ 5,\ \underset{\sim}{3,\ 5},\ 1,\ \underset{\sim}{3,\ 5},\ \underline{\underline{4}},\ 5,\ \underline{\underline{4}},\ 1,\ 5,\ 5,\ \underline{\underline{4}}].$$

The vector $q$ should then be $q = [3,\ 0,\ 2,\ \underline{\underline{3}},\ 7]$. For example (shown with double underline) we have $q[4] = 3$ because the number 4 occurs 3 times, The matrix $R$ can be computed as

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \underset{\sim}{2} \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 3 & 2 \end{bmatrix}$$

For example (shown with squiggly underline) we have $R[3, 5] = 2$ because 5 follows immediately after 3 two times in the sequence. Finally, $P$ can be computed as

$$P = \frac{1}{15}\left(\frac{3}{15} + \frac{2}{3} + \frac{2}{7} + \frac{1}{7} + \frac{2}{2} + \frac{1}{7} + \frac{1}{3} + \frac{2}{2} + \frac{3}{7} + \frac{1}{3} + \frac{3}{7} + \frac{1}{3} + \frac{2}{3} + \frac{2}{7} + \frac{3}{7}\right) \approx 0.4451$$

## Assignment E Syllable counter

Speed-reading software presents a text to a reader one word at at time at a fast rate. According to one theory, the duration of presentation of a word should roughly be proportional to the number of syllables $N$ in the word. In this exercise, you must create a function that (approximately) estimates the number of syllables in a word, according to the following rule:

*The number of syllables is equal to the number of vowels in the word, except that two or more consecutive vowels count only as one, and one or more vowels at the end of the word are not counted.*

You can assume that the input contains only lower case characters `a`–`z`, of which the characters `a`, `e`, `i`, `o`, `u`, and `y` are vowels.

### Problem definition

Create a function named `syllables` that takes as input a string containing a single word, and returns the number of syllables estimated based on the rule above.

### Solution template

```
def syllables(word):
  #insert your code
  return N
```

| Input | |
|---|---|
| `word` | Word (string). |
| **Output** | |
| `N` | Number of syllables (whole number). |

### Example

Consider the input word `cheesecake`. The word contains one double-vowel and three single-vowels of which one is at the end of the word.

`cheesecake`

Thus, the number of syllables should be estimated as $N = 3$. The contributing vowels are underlined.

E