UNIVERSITY OF AGDER

# Files

Lecture 4

# Agenda

- Fibonacci
- Functions - Newton’s method
- Reading Files
- Biggest product

# Fibonacci

Each new term in the Fibonacci sequence is generated by adding the previous two terms.  By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

# Functions

- Creating Newton's Method:
  - A method to find better approximates of the roots (zeros) of a function

$$x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}$$

# Functions

- What do we need?
  - Function for f(x) which returns f(x)
  - Function for f’(x) which returns f’(x)
  - Function which calculates the new x
  - Stop condition, when f(x) is smaller than the error we choose

$$x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}$$

# Reading files

- Every file is read as text.
  - So if a file of numbers is opened, we have to manually convert it if we want numbers. We will come back to this

- Syntax

```
filename = “a_file.txt”
with open(filename, ‘r’) as f:
        result = f.read()
```

# Writing files

- Syntax

```
filename = “a_file.txt”
with open(filename, ‘w’) as f:
        f.write(“Python is cool“)
```

# Writing files

- We can also use loops to write

```
numbers = [1,2,3,4,5,6,7,8]
filename = “a_file.txt”
with open(filename, ‘w’) as f:
        for element in numbers:
            f.write(element + “\n”)
```

# Biggest product

- Download the file numbers.txt from Canvas

  It contains the following numbers:

```
1239809812497690481204712039840128476123098572389729138190851293812938120935810923581230958123095881585858129012384120391098901723498172340981283497241273589123750891235701283571092837512384712304897182309480982350928359012345123490893724012873408912123770185057912301958109123090581095067109401291234679612374891276349871263498123764918723461928312738649182374698123764987698768976987698769837643123470192378409823749812734981658192347558592758971298787908978978977889709789078987234981723948712344
```

# Biggest product

- Find the largest product of three adjacent digits in the sequence

  1*2*3 = 6

  2*3*9 = 54

- First ten digits:
  - 1239809812
  - 1239809812
  - 1239809812