Prof. Dr.-Ing. Nicolas Stein

Programming assignment

Python

Excercises (partly typical coding interview questions):

1. **Quadratic equation**

   Write a programme that calculates the zeros of the quadratic equation $ax^2 + bx + c = 0$. You determine the zeros using the so-called midnight formula

   $$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

   (a) First create a programme with fixed parameters a, b and c as a test example, e.g. $a = 1, b = 0, c = -4$, so that $x_{1,2} = \pm 2$ and first consider configurations with a positive number under the root.

   (b) Extend your programme so that the user can enter the parameters a, b and c via the console.

   (c) Look at the output of your programme if you now set $a = 1, b = 0$ and $c = 4$. Then expand your programme so that the case (discriminant under the root ¡ 0) is intercepted and you output a message.

   (d) Expand your programme again to also allow complex numbers as solutions.

   Note: A complex number z can be understood as a combination of two real numbers a and b: $z = a + bj$. Here a is the so-called real part and b the so-called imaginary part. The symbol j is referred to as an imaginary unit in Python. The following applies: $j = \sqrt{-1}$.

2. **Prime numbers**

   Write a Python programme that outputs all prime numbers between 100 and 200.

   (a) Firstly, think about how you can identify a prime number.

   (b) Implement your program in Python.

   Note: Modulo calculates the remainder b of the division n divided by m. You can define a function that assigns a unique divisor remainder b to each pair of numbers (n, m). This is called modulo and is implemented in Python with the %, e.g. 4%2=0 and 5%2=1.

3. **The FizzBuzz task**

   Given a list of integers `[45, 22, 14, 65, 97, 72]`. You are to implement the following in python:

   - Replace all numbers divisible by 3 with 'fizz'
   - Replace all numbers divisible by 5 with 'buzz'
   - Replace all numbers divisible by 3 and 5 with 'fizzbuzz'

4. **Palindromes**

   Write a Python programme that checks whether a string is a palindrome or not.

   Note: A palindrome is a meaningful sequence of letters, words or verses that result in exactly the same text when read forwards or backwards. (e.g. relief pillars)

5. **Bar charts with matplotlib**

   Write a program that visualises a bar chart with the data from Table 1. Plot Age, Sales and Income over the respective entries as a stacked bar chart with legend.

EMPID	Gender	Age	Sales	BMI	Income
E001	M	34	123	Normal	350
E002	F	40	114	Overweight	450
E003	F	37	135	Obesity	169
E004	M	30	139	Underweight	189
E005	F	44	117	Underweight	183
E006	M	36	121	Normal	80
E007	M	32	133	Obesity	166
E008	F	26	140	Normal	120
E009	M	32	133	Normal	75
E010	M	36	133	Underweight	40

Tabelle 1: Data

6. **Numpy and Matplotlib**

   Define two arrays of the same type and size which contain any x-values and y-values. The following plots are to be created using the matplotlib module

   (a) A scatter plot and a line plot

   (b) A scatter plot in which the points are connected to each other

(c) Display a label for the x-axis, the y-axis and the title

(d) Define two additional arrays with x and y values and plot both data sets, (x_1, y_1) and (x_2, y_2) in a single diagram. Display the names of the axes, the title and the legend

(e) Create a bar chart for the x and y values.

7. **Sort by insertion**

Sorting algorithms are among the best-known algorithms in computer science. Implement the so-called 'insertion sort' algorithm below. This is based on the following basic principles:

- human-like, intuitive sorting like in a card game
- starting with an empty hand, we pick up one card per time step and insert it into the correct position in the hand
- we compare the card from right to left with all the cards already in our hand
- at any time the cards are sorted in our hand

The algorithm can also be described with the following pseudocode with only one array:

```
INSERTION-SORT(A)
1  for j = 2 to A.länge
2      schlüssel = A[j]
3      // Füge A[j] in die sortierte Sequenz A[1..j − 1] ein.
4      i = j − 1
5      while i > 0 und A[i] > schlüssel
6          A[i + 1] = A[i]
7          i = i − 1
8      A[i + 1] = schlüssel
```

Implement the algorithm in python.