



P06 – Test Exam

Name, Vorname

Klasse

--	--

Hinweise

Lesen Sie die Aufgabenstellungen aufmerksam durch, *bevor* Sie Ihre Lösungen erarbeiten und die Antworten formulieren. Es ist empfohlen, sich zuerst einmal einen Überblick zu verschaffen und dann die Aufgabe anzupacken, von der Sie sich ein erstes Erfolgserlebnis versprechen. Danach läuft's erfahrungsgemäss lockerer. Lesen Sie dann die Aufgaben jeweils mehrmals durch, so dass Sie sicher sind, alles verstanden zu haben!

Die Zahlen in eckigen Klammern bedeuten den jeweiligen Punktwert der Aufgabe.

Total sind **80 Punkte** erreichbar. Zeit: **90 Minuten**

Für Ihre Antworten und Lösungen steht der Platz in diesem Dokument zur Verfügung.

Allfällige Zusatzblätter sind klar und deutlich mit Ihrem Namen und Ihrer Klasseneinteilung zu beschriften und diesem Dokument beizulegen.

Diese Unterlagen sind zusammengeheftet. Bitte reissen Sie keine Blätter heraus. Merci.

Auf dem **letzten** Blatt befinden sich Standard Python-Funktionen und andere Hinweise, die zum Lösen der Aufgaben nützlich sein könnten.

Inhalt

Nr.	Titel	Punktwert	Punkte
1	Mixed Pickles	8	
2	Badewetter	15	
3	Datentypen, Variablen, Ausdrücke	15	
4	Eindimensionale Liste	20	
5	Ablaufstrukturen	14	
6	Spielen mit Primzahlen	8	
	Anhang		
		Total	

1. Mixed Pickles

[8*1]

Kreuzen Sie bei den unten stehenden Aussagen unmissverständlich an, ob eine Aussage als ganzes zutrifft (**Richtig**) oder nicht (**Falsch**). Richtige Antworten geben je 1 Punkt Gutschrift, falsche hingegen je 1 Punkt Abzug. Keine Antwort entspricht 0 Punkten. Ein negativer Punktesaldo wird auf 0 Punkte aufgerundet.

R	F	Aussage
		Variablen in Python sind nur Namen, die an ein Objekt gebunden werden. Ein Objekt kann zu einem Zeitpunkt nur einen Namen haben.
		Zur Iteration über alle Elemente einer Liste sollte eine while -Schleife verwendet werden.
		Das folgende Codestück führt zu einem Fehler („Exception“): <pre>temperatur = 22.5 wert = int(temperatur) print "" + str(wert) + " (" + str(temperatur) + ")"</pre>
		Die folgenden beiden Codestücke sind bezüglich Ihrer Semantik identisch. <div style="display: flex; justify-content: space-between;"> <div> <p>Codestück 1:</p> <pre>liste = range(0, 10) for i in range(len(liste)): print liste[i]</pre> </div> <div> <p>Codestück 2:</p> <pre>for element in range(10): print element</pre> </div> </div>
		Die folgende Schleife iteriert über alle Elemente der Liste namens matrix: <pre>for rows in matrix[len(matrix) : 0 : -1]: #...</pre>
		Der folgende Funktionskopf enthält keine Fehler und entspricht den Konventionen von Python (PEP8) zur Benennung von Funktionen und Variablen: <pre>def Berechne_Geschwindigkeit(weg, zeit)</pre>
		Ein Stückchen Python Code lautet wie folgt: <pre>def verarbeite_zahl(zahl): zahl = zahl + 50 //weiterer Code, für die Aufgabe nicht relevant zahl = 10 verarbeite_zahl(zahl) print "Zahl=", zahl</pre> <p>Das Script erzeugt die Ausgabe „Zahl= 60“ auf der Python Konsole.</p>
		Der Wert einer globalen Variable x mit Typ int kann von einer Funktion im selben Script geändert werden.



2. Badewetter

[15]

In einer Liste sind die Temperaturen aller 7 Wochentage (Montag bis Sonntag) der kommenden Woche abgespeichert. Joe möchte wissen, an welchem Tag er wohl ins Schwimmbad gehen soll.

Schreiben sie dazu eine vollständige Funktion (Kopf und Körper; es ist kein Testprogramm nötig!), die eine Liste mit den Temperaturen der nächsten Woche (in der Reihenfolge der Wochentage von Montag ab) entgegen nimmt und den optimalen Badetag als Text (Mo., Di., Mi. etc.) zurückgibt. Sie können annehmen, dass am Tag mit der höchsten Temperatur optimales Badewetter herrscht.

Falls an mehreren Tagen die gleiche höchste Temperatur herrschen sollte, so sollte die Funktion den ersten Tag liefern.

3. Datentypen, Variablen, Ausdrücke

[5*3]

Die folgende Tabelle enthält in der linken Spalte Python-Codeschnipsel, in der rechten Spalte ist nach einem Ergebnis gefragt. Bitte tragen Sie das Ergebnis entsprechend der Frage(n) direkt in die Tabelle ein. Seien Sie in Ihrer Antwort möglichst präzise, was z.B. den Datentyp des Ergebnisses angeht (machen sie etwa deutlich, ob es sich um einen `int`, `float` oder `string` Wert handelt).

Code	Ergebnis
<pre>x = 7 y = 3 e = x - y / y - x</pre>	Wert von <code>e</code> ?
<pre>a = [1,2,3,4,5] e = not len(a)%2 >0</pre>	<p>Wert von <code>e</code>?</p> <p>Mit dem Teilausdruck „<code>len(a)%2>0</code>“ fragt man anschaulich, ob ...?</p>
<pre>d = 2.3456789 s = 2 e = int(d * 10**s + 0.5) / 10.0**s</pre>	<p>Wert von <code>e</code>?</p> <p>Welche Bedeutung hat die Variable „<code>s</code>“ anschaulich?</p>
<p>Gegeben sind zwei positive ganze Zahlen <code>n</code> und <code>k</code> vom Typ <code>int</code>.</p> <p>Schreiben Sie einen Python-Ausdruck, der die grösste Zahl kleiner oder gleich <code>k</code> berechnet, die ganzzahlig durch <code>n</code> teilbar ist. (<code>k >=n</code>)</p> <p>(Hinweis: die Modulodivision <code>%</code> ist auch hier ganz nützlich.)</p>	<code>resultat =</code>
<pre>x = 9 y = x x = x + 2</pre>	Wert von <code>y</code> ?

4. Eindimensionale Liste

[18]

Gegeben sei eine Liste zufälliger Länge >1 , gefüllt mit zufälligen `float`-Zahlen im Bereich `0.0 - 99.99`.

Schreiben Sie eine Funktion `get_durchschnitt()`, welche den Durchschnitt aller Listen-Elemente berechnet.

Schreiben Sie eine zweite Funktion `get_nearest_index()`, welche unter Verwendung von der obigen Funktion `get_durchschnitt()` das Listen-Element sucht, welches dem Durchschnitt aller Elemente am nächsten liegt, und dessen Index zurückgibt.

Beide Funktionen nehmen eine `list` von `float` Werten als Parameter entgegen.

Tipp: Verwenden Sie die in Python bereits verfügbare Funktion `abs(<some_float>)`: Sie liefert den absoluten (positiven) Betrag eines `float` Wertes.



5. Ablaufstrukturen

[6+8]

5.1 String-Erzeugung

[6]

Schreiben Sie ein Script, welches eine Zeichenkette mit den Zahlen 1 bis 99, jeweils getrennt durch Komma, erzeugt und in folgender Form auf die Konsole schreibt (mit den entsprechenden Zahlen anstatt der drei Punkte):

1,2,3,...,97,98,99

Definieren Sie entsprechende Variablen und nutzen Sie eine Schleife, um die Zeichenkette mit Inhalt zu füllen. Beachten Sie, dass vor der ersten Zahl und nach der letzten Zahl kein Komma stehen soll.

5.2 Codeverständnis

[8]

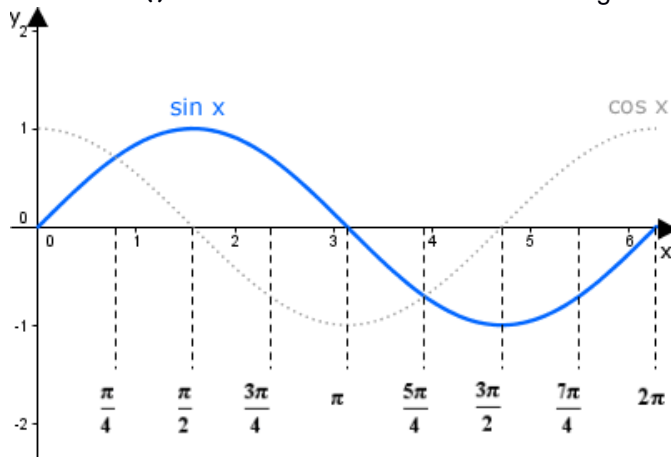
Der folgende Programmcode stammt aus einer Anwendung, die einen Tongenerator implementiert.

Analysieren Sie die Funktion `calc_wave_value(input, mode)`, indem sie die Resultate der Funktions-Rückgabewerte als Diagramm des `input`-Wertes zwischen 0.0 und 5.0 aufzeichnen.

Zeichnen sie den Kurvenverlauf für jeden der angegebenen Modi in die vorbereiteten Diagramme auf der nächsten Seite.

Hinweise:

- Der %-Operator (Modulo-Operator) berechnet den Divisions-Rest
- `math.sin()` stellt die Sinus-Funktion aus der Trigonometrie zur Verfügung:

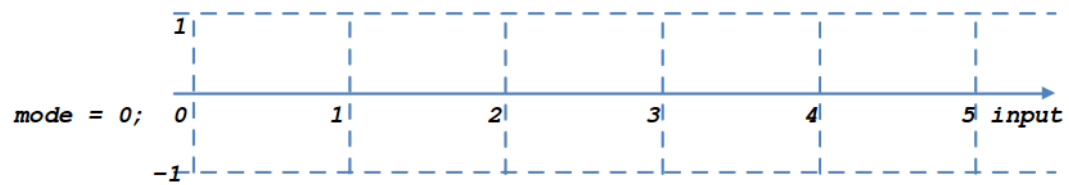
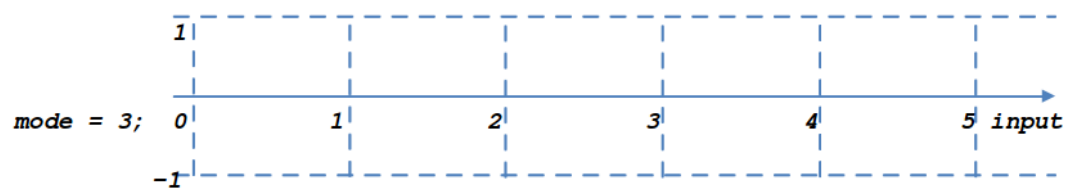
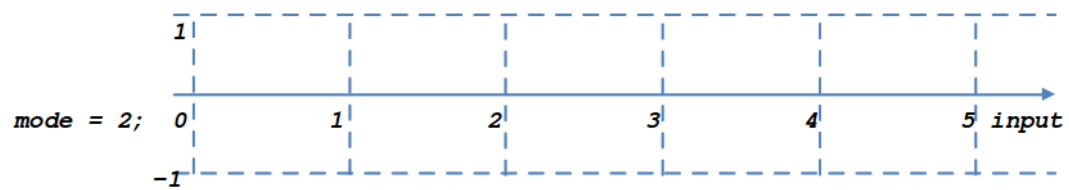
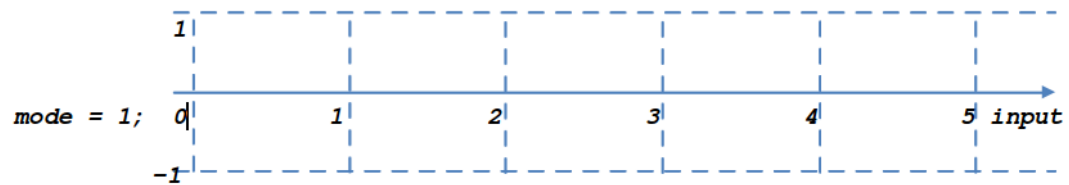


Wie man obiger Grafik entnehmen kann, ist beispielsweise `math.sin(math.pi/2) == 1.0`.
(Quelle: <http://images.onlinemathe.de/images/content/images/393c1b567e7dace5d443bf005e7a1c94.gif>)

- `math.pi` ist die Zahl $\pi = 3.141592653589793$

```
import math

def calc_wave_value(input, mode):
    if mode == 1:
        if input%2.0 < 1.0:
            return 1.0
        else:
            return -1.0
    elif mode == 2:
        return input%2.0 - 1.0
    elif mode == 3:
        return math.sin(input)
    else:
        return 0.0
```



6. Spielen mit Primzahlen

[8]

Gegeben ist eine Funktion `ist_primzahl(p)`, welche einen booleschen Wert zurückgibt.

Diese stellt fest, ob eine bestimmte `int` Zahl `p` eine Primzahl ist oder nicht. (Diese Funktion müssen Sie nicht entwickeln, sondern können Sie benutzen!)

Entwickeln Sie eine zusätzliche Funktion `naechst_groessere_primzahl(m)`, die ausgehend von einer Zahl `m` die nächst grösserer Primzahl `p` ermittelt. Sollte `m` selbst eine Primzahl sein, so liefert die Funktion `naechst_groessere_primzahl(m)` den Wert von `m` selbst als Ergebnis zurück.

Innerhalb der Funktion `naechst_groessere_primzahl()` soll die oben gegebene Funktion `ist_primzahl()` benutzt werden.

Anhang

Willkürliche Auswahl von Referenzmaterial aus der Python Hilfe.

Hilfe
range(...) range(stop) -> list of integers range(start, stop[, step]) -> list of integers Return a list containing an arithmetic progression of integers. range(i, j) returns [i, i+1, i+2, ..., j-1]; start (!) defaults to 0. When step is given, it specifies the increment (or decrement). For example, range(4) returns [0, 1, 2, 3]. The end point is omitted! These are exactly the valid indices for a list of 4 elements.
int(...) [...] Convert a number or string to an integer, or return 0 if no arguments are given. If x is floating point, the conversion truncates towards zero. If x is outside the integer range, the function returns a long instead.
float(...) [...] Convert a number or string to an integer, or return 0 if no arguments are given. If x is floating point, the conversion truncates towards zero. If x is outside the integer range, the function returns a long instead.
str(...) [...] Return a nice string representation of the object. If the argument is a string, the return value is the same object.
abs(...) abs(number) -> number Return the absolute value of the argument.
len(...) len(object) -> integer Return the number of items of a sequence or mapping.