Informatik I WS 2005/2006

Übungsblatt 5

Abgabe: 5.12.05 - 7.12.05

Aufgabe 1

Punkte: 4

a) Der Mathematiker Leibniz fand heraus, dass sich die Kreiszahl π durch folgende unendliche Reihe berechnen lässt:

$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - ...$

Schreiben Sie ein Programm leibniz.py, welches nach diesem Verfahren π berechnet.
Das Ergebnis nähert sich π mit zunehmender Summenlänge immer weiter an. Beim Start soll dem Programm die Anzahl der Summanden per Kommandozeile übergeben werden, damit das Programm nicht endlos läuft.

b) Der Mathematiker John Wallis fand eine andere Methode zur Berechnung von π:

$\frac{\pi}{2} = \frac{2}{1} * \frac{2}{3} * \frac{4}{3} * \frac{4}{5} * \frac{6}{5} * \frac{6}{7} * \frac{8}{7} * \frac{8}{9} * ...$

Schreiben Sie ein Programm wallis.py, welches π nach diesem Verfahren berechnet.
Das Ergebnis nähert sich π wiederum mit zunehmender Produktlänge weiter an. Die Anzahl der Faktoren soll auch hier dem Programm beim Start per Kommandozeile übergeben werden.

Aufgabe 2

Punkte: 5

Die Caesar-Verschlüsselung ist eines der ältesten bekannten Verfahren zur Verschlüsselung. Bereits Caesar verschlüsselte seine geheimen Nachrichten an Cicero mittels dieses Verfahrens.

Die Verschlüsselung erfolgt ganz einfach durch Verschieben jedes Buchstaben des Alphabets um k Positionen. Die Tabelle zeigt ein Beispiel für die Caesar-Verschlüsselung mit k = 3:

Original:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Chiffre:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Beim Chiffrieren wird z.B. „VENI, VIDI, VICI“ zu „YHQL, YLGL, YLFL“ codiert.
Schreiben Sie ein Programm caesar.py welches einen Verschiebungsparameter k als Kommandozeilenargument übergeben bekommt, und einen vom stdin gelesenen Text mittels Caesar-Chiffre codiert. Gehen Sie davon aus, dass als Eingabe nur Grossbuchstaben verwendet werden.
Schreiben Sie auch ein Programm um eine mittles Caesar-Chiffre verschlüsselte Botschaft wieder zu decodieren.
(Tip: Ist dafür wirklich ein neues Programm nötig?)

Aufgabe 3

Punkte: 6

Im Volksmund heisst es, dass Freitag der 13 ein Unglückstag sei. Schreiben Sie ein Programm, welches berechnet, wie viele solcher Freitage es im Jahr mindestens gibt, und wie viele es maximal sein können.
Vor der Berechnung soll das Programm den Benutzer noch fragen, ob die Berechnung für ein normales oder ein Schaltjahr durchgeführt werden soll.

Tip: Man sollte sich zuerst darüber klar werden, wie viele verschiedene Möglichkeiten es überhaupt geben kann und diese dann alle durchprobieren.

Aufgabe 4

Punkte: 5

In der Vorlesung haben Sie bereits den „Chaos Game“-Algorithmus zur Berechnung des Sierpinski-Dreiecks kennen gelernt. In dieser Aufgabe soll nun ein ähnlicher Algorithmus farn.py implementiert werden.
Das Programm soll, analog zum „Chaos Game“, N Punkte $(x_0, y_0), (x_1, y_1), (x_2, y_2)...(x_{n-1}, y_{n-1})$ zeichnen. Die Zahl N soll dem Programm per Kommandozeile übergeben werden.
Die Positionen der Punkte (x_i, y_i) werden folgendermassen berechnet:
Anfangs sollen die Variablen x_0 und y_0 die Werte 0.5, bzw 0 haben. Die neuen Positionen für die nächste Iteration werden gemäss folgender Tabelle berechnet:

Wahrscheinlichkeit	x_{i+1}	y_{i+1}
2%	0.5	0.27y
15%	-0.139x_i + 0.263y_i + 0.57	0.246x_i + 0.224y_i - 0.036
13%	0.170x_i - 0.y_i + 0.408	0.222x_i + 0.176y_i + 0.0893
70%	0.781x_i + 0.034y_i + 0.1075	-0.032x_i + 0.739y_i + 0.27

Zum Zeichnen der Punkte kommt wie in der Vorlesung beim „Chaos Game“ die Python Imaging Library zum Einsatz. Deren genauer Funktionsumfang und auch der Gebrauch von anderen Modulen wird im Laufe der Vorlesung noch erklärt werden. Um diese Aufgabe zu lösen, benötigen Sie lediglich die Funktionen:

- import Image #Importiert die Library
- im = Image.new(„RGB“, (512, 512), (256, 256, 256)) #Erzeugt ein neues Bild
- im.putpixel ((int(x*512), int(y*512)), (0, 1, 0)) #Zeichnet einen grünen Pixel an die Position (x*512, y*512)
- im.show() #Zeigt das Bild an

Zum Erzeugen der Zufallszahlen verwenden Sie analog zum „Chaos-Game“ die Funktion random.random(), die eine zufällige Fliesskommazahl Zahl zwischen 0 und 1 zurück liefert.
Auf der Vorlesungshomepage steht eine vorgefertigte Datei namens farn.py zum Download bereit. Dort müssen Sie dann nur noch den Python-Code zur Berechnung der neuen Punktepositionen an den vorgegebenen Stellen eintragen.

Kleine Zusatzaufgabe (ohne Bewertung) für die Experimentierfreudigen: Testen Sie das obige Programm auch mal mit anderen Parametern für die Transformation, beispielsweise:

Wahrscheinlichkeit	x_{i+1}	y_{i+1}
10%	0.05	0.6y_i
10%	-0.05x_i	-0.5y_i + 1.0
20%	0.46x_i - 0.15y_i	0.39x_i + 0.38y_i + 0.6
20%	0.47x_i - 0.15y_i	0.17x_i + 0.42y_i + 1.1
20%	0.43x_i + 0.28y_i	-0.25x_i + 0.45y_i + 1.0
20%	0.42x_i + 0.26y_i	-0.35x_i + 0.31y_i + 0.7

Das zugehörige Python-File sollte passenderweise den Namen baum.py statt farn.py erhalten.