PYTHON PROGRAMMING PROJECTS

CS102

1. Guidelines.

Your CS102 project involves writing two programs. All programs must be written in the Pythonlanguage. Use comments liberally to indicate what you are doing and to make the code more readable. Present the finished programs to your lecturer, Dr. Götz Pfeiffer, room C219 in Arus de Brun. Deadline for submission is Friday, January 12, 2007. A 10% penalty will be applied to late submissions. No more submissions will be accepted after January 31, 2007.

1. **Loan Repayments.** Write a program to calculate the repayments on a given loan over a period. The formula is:
$$x = P\,\frac{r\,(1+r)^n}{(1+r)^n - 1}$$
where €P is to be repaid in equal installments of €x over n periods at an interest rate r *per period*. (Be careful to note that, for example, an interest rate of 3% gives $r = 3/100$ and not 3.)

   Use the program to find the monthly repayments on a loan of €250000 over 15 years (180 months) at a rate of 6% *per annum*.
2. **Count Letters.** Write a program that counts the number of times the different letters of the alphabet (`a`, `b`, `c`, ..., `z` and `A`, `B`, `C`, ..., `Z`) occur in a file. Do not distinguish between lowercase and uppercase letters.

   Can you extend the program in such a way that it also reports on the number of:
   (a) vowels,
   (b) consonants,
   (c) uppercase letters,
   (d) lowercase letters

   that were read?
3. **Random Numbers.** Use the `random.randint(1, 100)` (from the `random` module) to generate 500 random numbers between 1 and 100. Count the number of occurrences of each number and print out the frequency of occurrence.

   Can you extend the program in such a way that, using the `graphics` module, it displays a bar chart showing for the ranges 1–10, 11–20, ... 91–100 how many numbers fall into the range?
4. **Time Management.** Write a function that takes the time as three integer arguments (for hours, minutes and seconds), and returns the number of seconds since midnight. Use this

Date: 22 November 2006.

function to write a program that asks for two time values in the form $hh : mm : ss$ and calculates the difference in seconds.

5. **Newton's Method.** Use Newton's method to find a root of $f(x) = x - \sin(x)$. The formula is:
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$
Get it to print out the answer when the difference between successive iterations is less than `1e-8`.
6. **Square and Cube Roots.** Let a be a positive real number and let the sequence of real numbers numbers x_i be given by $x_0 = 1$, and
$$x_{i+1} = \frac{1}{2}\left(x_i + \frac{a}{x_i}\right)$$
for $i = 0, 1, 2, \ldots$. Then x_i converges to the square root of a.

   Write a program that reads in the value of a interactively and uses this algorithm to compute the square root of a.

   Can you extend the program so that it also finds the cube root of a? The formula is: $x_0 = 1$, and
$$x_{i+1} = \frac{1}{3}\left(2x_i + \frac{a}{x_i^2}\right)$$
for $i = 0, 1, 2, \ldots$.
7. **Quadratic Polynomials.** A polynomial of degree 2 in x is given by
$$a\,x^2 + b\,x + c.$$
Write the code for the function
`f(a, b, c, x)`
that will compute the value of an arbitrary polynomial of degree 2. Use the identity
$$(ax + b)x + c = ax^2 + bx + c$$
to minimise multiplications.

   Use your function and the `graphics` module to plot the polynomial $x^2 - 3x + 2$ on the interval $[0, 3]$. The variable x should go from 0 to 3 in steps of 0.1. By examining the plot, can you tell where the roots of the polynomial are?
8. **Perfect Numbers.** An integer number is said to be a *perfect number* if its factors (including 1 but excluding the number itself) add up to the number. For example, 6 is a perfect number because its factors are $1, 2, 3$ and $6 = 1+2+3$. Write a function `isPerfect()` that determines if a parameter `number` is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and 10000. Print the factors of each perfect number to confirm that the number is indeed perfect.
9. **Guess My Number.** Write a program that plays the game "Guess My Number" as follows. Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 1000. The program then prints

```
I have a number between 1 and 1000.
Can you guess my number?
Please type your first guess.
```

The player then types a first guess. The program responds with one of the following.

1. `Excellent!  You guessed the number!`
   `Would you like to play again (y or n)?`
2. `Too low.  Try again.`
3. `Too high.  Try again.`

If the player's guess is incorrect, your program should loop until the player finally gets the number right.

10. **Roman Numerals.** Write a function that converts a positive integer (in the range 1 to 4999) into the corresponding *Roman numeral*. Use this function to write a program that prints a table of all the Roman numeral equivalents of the decimal numbers in the range 1 to 4999.
11. **Doomsday.** The *Doomsday rule* provides a way of calculating the day of the wee of a given date. It is based on the fact that within any calendar year, the dates 4/4, 6/6, 8/8, 10/10 and 12/12 always occur on the same day of the week. Also the *last* day of February falls on the same day of the week. This day is called the year's *Doomsday*. For a year $2000 + y$ in the range 2000 – 2099 it is determined by the formula

$$\text{Tuesday} + \left(\left\lfloor \frac{y}{12} \right\rfloor + y \bmod 12 + \left\lfloor \frac{y \bmod 12}{12} \right\rfloor \right) \bmod 7$$

(where $\lfloor \frac{a}{b} \rfloor$ denotes the result of the integer division `a / b`). For a year $1900+y$ in the range 1900 – 1999, replace 'Tuesday' by 'Wednesday' in the formula. Write a program that inputs a date in the range 1900 – 2099 and uses the Doomsday rule to determine the day of the week.

12. **Regression Line.** Write a program that graphically plots a regression line, that is, the line with the best fit through a collection of points. First ask the user to specify the data points by clicking on them in a graphics window. To find the end of input, place a small rectangle labeled "Done" in the lower left corner of the window; the program will stop gathering points when the user clicks inside that rectangle.

    The regression line is the line with equation

$$y = \overline{y} + m(x - \overline{x}),$$

where

$$m = \frac{\sum x_i y_i - n\overline{x}\overline{y}}{\sum x_i^2 - n\overline{x}^2},$$

$\overline{x}$ is the mean of the x-values, $\overline{y}$ is the mean of the y-values, and n is the number of points.

As the user clicks on the points, the program should draw them in the graphics window and keep track of the count of input values and the running sum of x, y, x^2 and xy values. When the user clicks the "Done" rectangle, the program then computes the value of y (using the equations above) corresponding to the x values at the left and right edges of the window

to compute the endpoints of the regression line spanning the window. After the line is drawn, the program will pause for another mouseclick before closing the window and quitting.

13. **Five-click House.** Write a program that allows the user to draw a simple house using five mouse clicks. The first two clicks will be the opposite coreners of the rectangular frame of the house. The third click will indicate the center of the top edge of a rectangular door. The door should have a total width that is $\frac{1}{5}$ of the width of the house frame. The sides of the door should extend from the the corners of the top down to the bottom of the frame. the fourth click will indicate the center of a square window. the window is half as wide as the door. The last click will indicate th epeak of the roof. The edges of the roof will extend from the point at the peak to the corners of the top edge of the house frame.