The questions on page 1 are for the in-class exam and must be solved within 50 minutes without any computing device. Text books, copies of slides, and notes are allowed.

The decision to do the take-home version of this exam can be postponed till 10:49AM, if you do not hand in at 10:50AM the answers to the questions below.

In grading, all questions will receive the same amount of points.

1. A continued fraction expansion of $\sqrt{2}$ is $1 + \dfrac{1}{2 + \frac{1}{2+\cdots}}$.

   The list of convergents in the continued fraction expansion of $\sqrt{2}$ is $[1, 2, 2, 2, 2, \ldots, 2]$.

   Give the definition of a Python function which takes on input a list L of convergents and returns the floating point approximation (of type `float`) defined by the continued fraction expansion given by L.

2. Consider the expression $f(x) = \exp(\sin(2\pi x))$.

   Write `sympy` code to compute Taylor series of $f(x)$ about $x = 0.5$, series with 2, 3, 4, and 5 terms.

   Give code to evaluate the series at 0.6 and to compare with $f(0.6)$.

3. Give Python code for the following:

   (a) Generate a random 100-by-100 matrix with random elements, normally distributed with mean 0 and standard deviation 1 using the function `normal` in the module `random` of `numpy`.

   (b) Plot the eigenvalues with `pyplot` or `pylab`.

4. Given an n-by-n matrix A, the permanent of the matrix is

$$\text{permanent}(A) = \sum_{\sigma} \prod_{i=1}^{n} A_{i\sigma_i}$$

   where σ runs over all permutations of the tuple $(1, 2, \ldots, n)$. The expansion formula for the permanent above is very similar to the row expansion formula for the determinant, except for the sign changes, which are absent in the permanent expansion.

   Write a Python function `permanent` that takes on input a matrix and that returns the permanent of the matrix. An auxiliary recursive function uses the expansion formula above to compute the permanent.

The questions on page 2 are for the take-home exam and must be solved individually. Answers are due on Monday 10 October, at 10AM.

1. A continued fraction expansion of $\sqrt{2}$ is $1 + \frac{1}{2 + \frac{1}{2 + \cdots}}$.

   The list of convergents in the continued fraction expansion of $\sqrt{2}$ is $[1, 2, 2, 2, 2, \ldots, 2]$.

   Consult the documentation in Sage and find how to compute the continued fraction expansion of $\sqrt{2}$. Give the appropriate commands in Sage for this calculation.

   How many terms in the list of convergents should be given to give a continuation fraction expansion of $\sqrt{2}$ accurate to 20 decimal places?

2. Write a Python function that takes on input a `sympy` expression `f` in the symbol `x`, a value for `x0` and a tolerance $\epsilon > 0$.

   The function returns a polynomial in `x` which is the truncated Taylor series of the expression about `x0`, truncated so that the error between the expression and the truncated series at `x0 + 0.1` is less than ϵ.

   Illustrate your function on the expression $\exp(\sin(2\pi x))$ at $x = 0.5$ and with $\epsilon = 0.001$.

3. Make a GUI following an object-oriented design to display the eigenvalues of random matrices on canvas.

   The GUI has three widgets: an entry field for the user to fill in the dimension of the matrix, a canvas to display the eigenvalues in the complex plane, and a button. When pressed the button clears the canvas, generates a new matrix, computes its eigenvalues, and then displays the eigenvalues on canvas.

   The entries of the matrix are normally distributed with mean 0 and standard deviation 1. Use the function `normal` in the module `random` of `numpy`.

4. Write a Python script that does the following:

   (a) The user is prompted for a dimension n and a ratio $r \in [0, 1]$. Then the script generates a random n-by-n matrix A of boolean values. For each entry, the computer does a coin toss to decide whether to put False or True in the entry. If $r = 0.5$, then the coin is fair, for higher values of r, True is favored above False. For example, if $r = 0.73$, then on average for large enough matrices, 73% of the entries of A will be True.

   (b) Interpret the matrix A as an adjacency matrix of a graph with n nodes. If the (i, j)th value of A is True, then nodes i and j are connected. A path between two nodes may just be a list with one edge (i, j), but in general the path may visit many intermediate nodes, e.g.: $[(i, k), (k, l), (l, m), (m, j)]$. Write a recursive function that given A, i, and j returns a path between nodes i and j or an empty list if no path is found after an exhaustive search through the matrix.

   (c) The script prompts the user for values for i and j and then prints the outcome of your recursive function.