**[Python, C, JavaScript]** Given two strings, compute the minimum number of edits needed to transform the first string into the second string. A single edit is an insertion, deletion, or substitution of a single character.

**[Python, C, JavaScript]** Given an input string and a pattern, implement regular expression matching with support for '.' and '*'.

- '.' Matches any single character
- '*' Matches zero or more of the preceding element

**[JavaScript]** You are tasked to implement a real-time audio processing module within a web browser. You need to record audio from the microphone and visualize the loudness of the audio signal within a webpage. The implementation needs to support the latest releases of Chrome, Safari, and Firefox.

**[*]** You are tasked to collect 1000 hours of "labeled English speech data" for training purposes. The data is pairs of audio files and their corresponding transcripts. The transcripts should be 99%+ accurate. How do you go about this? How fast can you gather 100 hours? How about 10000 hours? Provide as much detail as possible.
HINT: TED Talks are freely available for download and are also hand transcripted.

**[C]** We need to instantiate a large number of objects (100,000s) per second. All objects have the same size in bytes. Compute and memory resources are very limited. What are the drawbacks of using "malloc"? Design and implement a memory pool to overcome these shortcomings. Elaborate on how the newly-designed pool overcomes "malloc"'s drawbacks.

**[C]** Implement a matrix-vector multiplication function in C. All elements are stored as fixed-point numbers. The elements of the matrix are stored as Q1.7. Elements of the vector are stored as Q5.10. The result should be stored as Q5.10. Implement the function in pure C. Now assume you are running on an Arm Cortex-A with NEON SIMD (e.g. Raspberry Pi 3). Re-write the C implementation using intrinsics to optimize for speed. Assume the cache is large enough to store all inputs. What would you change if the cache was not large enough?

void matrix_vector_multiplication(const int8_t *M, const int16_t *V, int16_t *R);

**[Python-Numpy]** Implement (in Numpy) a unidirectional multi-layer LSTM classifier with input and forget gates coupled. You can find information about this variant of LSTM here (look for **CIFG**). The model should accept a feature vector as input and emit the corresponding posterior. Then train a character-based language model to generate text resembling Shakespear (use any online dataset you see fit).

[**Python**] Grapheme to Phoneme (G2P) is a model that accepts a word as input and outputs its pronunciation. It is useful for expanding the vocabulary of a speech recognition system at runtime (e.g. contact list on your smartphone). Train a G2P model for English language. You can use any technique and any modern deep learning framework.
HINT: CMU has released an open-source dictionary that can be used as training data here.