# EE2703 - Assignment3 - EE24B114

## CODE Summary:

### 1. Keyboard Grid and Layout

- QWERTY Grid: Functions assign physical (x, y) coordinates to each letter according to a simple QWERTY-like staggered grid.
- Initial Layout: Uses QWERTY mapping as the baseline; unknown/non-letter characters default to the spacebar position

### 2. Text Preprocessing

- The code converts the input text to lowercase and filters it so only permitted characters are kept; others are mapped to space.

### 3. Cost Function

- path_length_cost: Computes the sum of Euclidean distances between the coordinates of each pair of consecutive characters in the text, according to the current layout.
- This models the total finger "travel distance" needed to type the text.

### 4. Simulated Annealing Optimizer

- Algorithm:
    - Starts from the initial (QWERTY) layout.
    - Proposes new layouts by randomly swapping pairs of key positions.

      ```
      k1, k2 = rng.sample(keys, 2)

      new_layout[k1],
      new_layout[k2]=new_layout[k2],new_layout[k1]
      ```

    - Accepts new layouts if they improve the cost; or, sometimes, accepts worse layouts based on a probability that decreases as temperature cools (the essence of simulated annealing).
    - Tracks both the best cost seen so far and the cost of the current candidate at each iteration for plotting.
- Cooling Schedule: Temperature (`T`) decays by multiplying with `alpha` after every iteration.

`args = parser.parse_args()` is required to actually read and store user inputs

Defines an optional  argument `--x` with a specific data type and default value of `<program-assigned-data>`

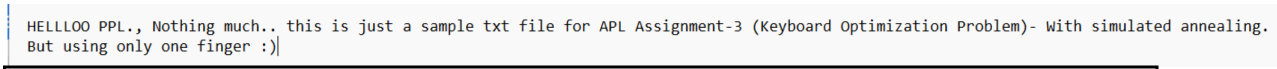
## 5. Plotting

- Two plots are generated:
    - A cost/iteration trace showing progress of optimization.
    - A visual scatter plot of final optimized key positions.
- Images are saved in the current working directory.

## 6. Main Routine

- Loads text, maps the initial layout, evaluates baseline cost, runs simulated annealing, and outputs results.

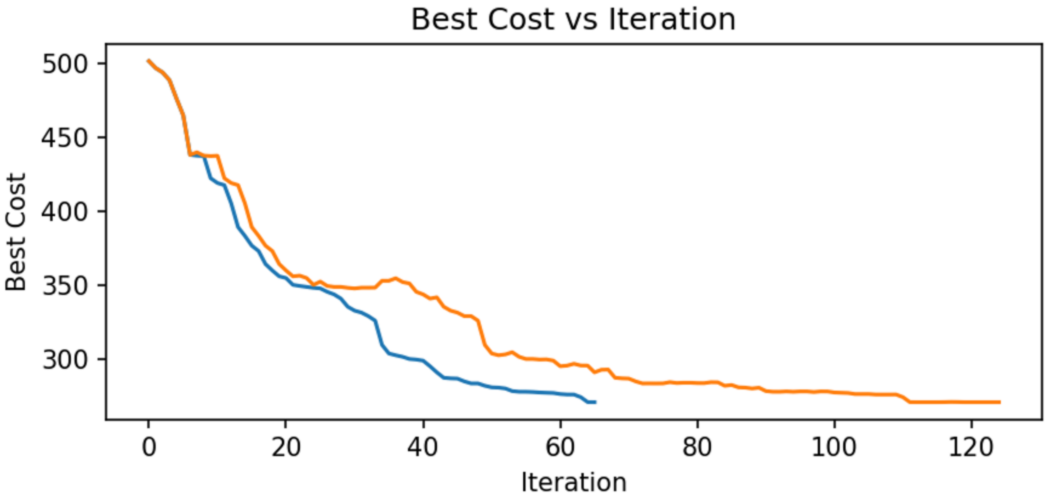
Using this *sample text*:

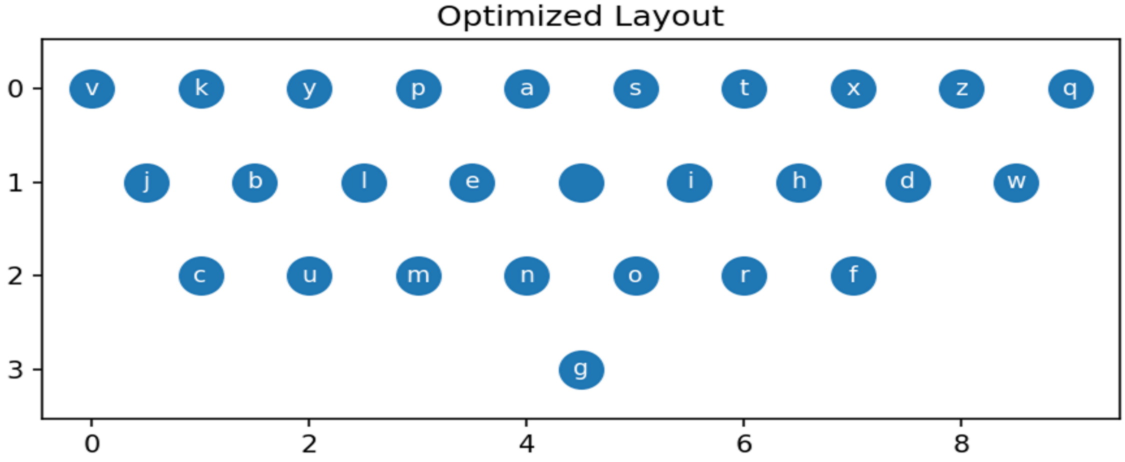


```
HELLLOO PPL., Nothing much.. this is just a sample txt file for APL Assignment-3 (Keyboard Optimization Problem)- With simulated annealing.
But using only one finger :)
```

- Baseline (QWERTY assignment) cost: 501.2296
- Optimized cost: 271.0465 (improvement 230.1831)
- Runtime: 0.33s over 5000 iterations

-





Run the code: `python3 kbd_optim.py --input "sample.txt" --iters 5000 --t0 1.0 --alpha 0.999`