

Example Assignments

These are example assignments designed to showcase PyAutoGrader's capabilities. They demonstrate single-file grading, multi-file grading, extra file distribution, solution comparison, code quality checks, and AI detection.

Assignment 1: Weather Data Analysis

Course: ENGR 101 - Introduction to Programming **Points:** 100

Instructions

You are provided with a CSV file (`weather_data.csv`) containing 30 days of summer weather data with columns: `date`, `high_temp`, `low_temp` (temperatures in Fahrenheit).

Write a Python script that:

1. **Reads the CSV file** using `numpy.genfromtxt()` to load the numeric columns. Do not use `pandas`.
2. **Uses a for loop** to iterate through the data rows.
3. **Computes and stores** the following values in variables with these exact names:
 - `avg_high` - the average of all high temperatures
 - `avg_low` - the average of all low temperatures
 - `max_temp` - the highest temperature recorded (from either high or low columns)
 - `min_temp` - the lowest temperature recorded (from either high or low columns)
 - `hot_days` - the count of days where the high temperature is strictly greater than 85 F
4. **Prints a summary** of your results. Your output must include the text "Average High" somewhere in the printed output.
5. **Includes a name comment** at the top of your file in the format: `# Name: FirstName LastName`

File Requirements

- **Filename:** Must contain the word "weather" (e.g., `weather_analysis.py`, `my_weather_report.py`)
- **Data file:** `weather_data.csv` will be provided - place it in the same directory as your script

Example Output

```
Weather Data Analysis Results
Average High Temperature: 84.7 F
Average Low Temperature: 61.8 F
Maximum Temperature: 102.0 F
Minimum Temperature: 50.0 F
Number of Hot Days (>85 F): 13
```

Grading Criteria

Category	Points
Correct filename	5
Name comment included	5
Code originality check	5
Uses a for loop	10
Correct avg_high	15
Correct hot_days count	10
Other variables match solution	20
Output contains "Average High"	10
No single-letter variable names	10
Lines under 100 characters	10
Total	100

Assignment 2: Simply Supported Beam Deflection

Course: ME 325 - Mechanical Engineering Analysis **Points:** 100

Background

A simply supported beam of length L is subjected to a single point load P applied at a distance a from the left support. You will compute the support reactions and the deflection curve using Euler-Bernoulli beam theory, then plot the results.

Beam Parameters

Use the following values in your analysis:

Parameter	Value	Description
L	10 m	Beam length
E	200e9 Pa	Young's modulus (steel)
I	8.33e-6 m ⁴	Second moment of area
P	5000 N	Point load
a	4 m	Load position from left support

File Structure

You must organize your code into **two files**:

myfunctions.py - Calculation Functions

Implement the following three functions with **exact parameter names as shown**:

```
beam_reactions(P, a, L)
```

- Computes the support reactions RA and RB
- $RA = P * (L - a) / L$
- $RB = P * a / L$
- **Returns:** a tuple (RA, RB)

```
beam_deflection(x, P, a, L, E, I)
```

- Computes the beam deflection at position x (can be a scalar or numpy array)
- Uses the Euler-Bernoulli piecewise deflection equation
- $b = L - a$
- For $0 \leq x \leq a$: $v(x) = (P b x) / (6 L E I) (L^2 - b^2 - x^2)$
- For $a < x \leq L$: $v(x) = \text{above} + P (x - a)^3 / (6 E * I)$
- **Returns:** deflection value(s), positive downward

```
max_deflection(P, a, L, E, I)
```

- Computes the maximum deflection magnitude
- **Returns:** a positive float

```
### beam_analysis.py - Main Analysis Script
```

- Import numpy and your functions from myfunctions
- Define the beam parameters listed above
- Call `beam_reactions()` to get support reactions
- Create a numpy array of 100 evenly spaced x positions from 0 to L
- Compute the deflection curve using `beam_deflection()`
- Compute the maximum deflection using `max_deflection()`
- **Plot** the deflection vs. position with:
 - x-axis label: "Position (m)"
 - y-axis label: "Deflection (m)"
 - Title: "Beam Deflection"

```
## Grading Criteria
```

Category	Points
numpy imported	5
beam_reactions exists	5
beam_deflection exists	5
max_deflection exists	5
beam_reactions has correct parameters (P, a, L)	5
beam_reactions returns correct values	15
beam_deflection returns correct values	15
beam_deflection is called in main script	5
Plot is created	10
Plot has correct labels and title	10
Plot data matches solution	10
beam_reactions returns a tuple	10
Total	100