**Easy Tasks (60-74 points)**

**Task 1: Function Evaluator**

Objective: Write a program to evaluate a simple linear function f(x) = mx + b.

* Input:
  + m (slope): 2
  + b (y-intercept): 3
  + x (value): 4
* Output:
  + f(x): 11

Instructions: Implement a function that accepts three parameters **m**, **b**, and **x** and returns the value of **f(x)**.

**Task 2: Domain and Range Identifier**

Objective: Implement a program that identifies the domain and range of a given set of ordered pairs.

* Input:
  + Set of ordered pairs: {(1, 2), (3, 6), (4, 8)}
* Output:
  + Domain: {1, 3, 4}
  + Range: {2, 6, 8}

Instructions: Develop a function that takes a list of ordered pairs and returns two sets: one for the domain and one for the range.

**Task 3: Even/Odd Function Identifier**

Objective: Create a program that determines if a given function is even, odd, or neither based on its graph.

* Input:
  + Graphical function data (as a list of points): [(-1,1), (0,0), (1,1)]
* Output:
  + Function Type: Even

Instructions: Implement a function that processes a list of points representing a function's graph and determines if the function is even or odd.

**Medium Tasks (75-89 points)**

**Task 4: Injective Function Validator**

Objective: Create a program that checks if a function is injective based on a set of input-output pairs.

* Input:
  + Set of input-output pairs: {(2, 4), (3, 6), (4, 8)}
* Output:
  + Injective Status: True

Instructions: Develop a function that takes a list of input-output pairs and determines if the function is injective.

**Task 5: Surjective Function Checker**

Objective: Implement a program that checks if a function is surjective from its domain to a given codomain.

* Input:
  + Function pairs: {(1, 2), (2, 3), (3, 4)}
  + Codomain: {2, 3, 4}
* Output:
  + Surjective Status: True

Instructions: Implement a function that validates if a given function is surjective.

**Hard Tasks (90-100 points)**

**Task 6: Function Combination Tool**

Objective: Implement a program that combines two functions f(x) and g(x) by addition, subtraction, multiplication, or division.

* Input:
  + f(x) = x^2
  + g(x) = 2x + 1
  + Operation: Addition
  + Value: x = 3
* Output:
  + Result: 16

Instructions: Develop a program that takes two functions, an operation, and a value, then returns the result of the combination.

**Task 7: Graph Information Extractor**

Objective: Write a program to extract and report key features from a function’s graph such as intercepts, maxima, and minima.

* Input:
  + Graphical function data (as a list of points): [(-2, -4), (-1, -1), (0, 0), (1, 1), (2, 4)]
* Output:
  + x-intercepts: {0}
  + y-intercepts: {0}
  + Maxima: None
  + Minima: None

Instructions: Implement a function that analyzes a list of points to determine the graph's intercepts, maxima, and minima. Discuss the method for finding these features in a discrete set of points.