Puzzle 1: Division Function

Problem: Write a function `safe_divide(a, b)` that divides `a` by `b`. Handle the `ZeroDivisionError` and return a friendly message instead.

Example:

python

Copy code

```
safe_divide(10, 2)   # Output: 5.0
safe_divide(10, 0)   # Output: "Error: Cannot divide by zero."
```

Puzzle 2: Convert to Integer

Problem: Create a function `convert_to_int(value)` that tries to convert `value` to an integer. If it fails, return a message indicating the failure.

Example:

python

Copy code

```
convert_to_int("123")   # Output: 123
convert_to_int("abc")   # Output: "Error: Cannot convert to integer."
```

Puzzle 3: File Read

Problem: Implement a function `read_file(filename)` that attempts to read a file. Handle `FileNotFoundError` and return an appropriate message if the file doesn't exist.

Example:

python

Copy code

```
read_file("existing_file.txt")   # Output: Contents of the file

read_file("non_existent_file.txt")   # Output: "Error: File not found."
```

Puzzle 4: List Index Access

Problem: Write a function `get_element(lst, index)` that returns the element at the specified `index` in the list. Handle `IndexError` and return a message if the index is out of range.

Example:

python

Copy code

```
get_element([1, 2, 3], 1)   # Output: 2

get_element([1, 2, 3], 5)   # Output: "Error: Index out of range."
```

Puzzle 5: JSON Parsing

Problem: Create a function `parse_json(json_string)` that tries to parse a JSON string. If it fails, return an error message.

Example:

python

Copy code

```
parse_json('{"name": "Alice"}')   # Output: {'name': 'Alice'}

parse_json('{"name": "Alice",}')   # Output: "Error: Invalid JSON."
```

Puzzle 6: Safe String Split

Problem: Implement a function `safe_split(s, delimiter)` that splits a string `s` by `delimiter`. If `s` is not a string, handle the exception and return a message.

Example:

python

Copy code

```
safe_split("hello,world", ",")   # Output: ['hello', 'world']

safe_split(123, ",")              # Output: "Error: Input is not
a string."
```

Puzzle 7: Negative Square Root

Problem: Write a function `safe_sqrt(n)` that calculates the square root of `n`. If `n` is negative, handle the `ValueError` and return a message.

Example:

python

Copy code

```
safe_sqrt(4)   # Output: 2.0

safe_sqrt(-4)  # Output: "Error: Cannot take the square root of
a negative number."
```

Puzzle 8: Validate Age

Problem: Create a function `validate_age(age)` that checks if the age is a positive integer. If it is not, raise a `ValueError` with a message indicating the issue.

Example:

python

Copy code

```
validate_age(25)   # No output (valid)

validate_age(-5)   # Raises ValueError: "Error: Age must be a positive integer."
```

Puzzle 9: Accessing Dictionary Keys

Problem: Implement a function `get_dict_value(d, key)` that attempts to get the value of a key from a dictionary. Handle `KeyError` and return a message if the key doesn't exist.

Example:

python

Copy code

```
get_dict_value({'a': 1, 'b': 2}, 'b')   # Output: 2

get_dict_value({'a': 1, 'b': 2}, 'c')   # Output: "Error: Key not found."
```

Puzzle 10: Calculate Average

Problem: Write a function `calculate_average(numbers)` that calculates the average of a list of numbers. If the list is empty, handle the exception and return a message.

Example:

python

Copy code

```
calculate_average([1, 2, 3])   # Output: 2.0
```

```
calculate_average([])           # Output: "Error: Cannot
calculate average of an empty list."
```