Interpreter - evaluating binary (and unary) expressions

```
def process_mul_exp(ast):
    a = process_ast_exp(ast.left)
    b = process_ast_exp(ast.right)
    return a * b
```

```
def process_add_exp(ast):
    a = process_ast_exp(ast.left)
    b = process_ast_exp(ast.right)
    return a + b
```

```
def process_neg_exp(ast):
    result = process_ast_exp(ast.data)
    return -result
```

Unary expression

Binary expressions

```
def process_sub_exp(ast):
    a = process_ast_exp(ast.left)
    b = process_ast_exp(ast.right)
    return a - b
```

```
def process_div_exp(ast):
    a = process_ast_exp(ast.left)
    b = process_ast_exp(ast.right)
    if b == 0:
        raise ZeroDivisionError(f"Zero division error: cannot divide {a} by 0")
    return int(a / b)
```

Part 4. Putting it together