

KIK Language Test Cases

Test Case 1: Conditional Logic (if-else if-else)

This test demonstrates the use of an if-else if-else block to determine a grade based on a score.

Expected Output:
Grade: B

```
import "io.kik";

int kik() {
    int score = 85;

    if score >= 90: {
        cout << "Grade: A" << endl;
    } else if score >= 80: {
        cout << "Grade: B" << endl;
    } else: {
        cout << "Grade: C or lower" << endl;
    }

    return 0;
}
```

Test Case 2: Basic Arithmetic

This test shows basic variable declaration, addition, and printing the result to the console.

Expected Output:
Final output after adding: 98

```
import "io.kik";

int kik() {
    int a = 8;
    int b = 90;
    int result = a + b;

    cout << "Final output after adding: " << result << endl;

    return 0;
}
```

Test Case 3: Variable & Constant Declaration

This test verifies the syntax for declaring and initializing multiple data types, including the special rule for multi-variable lines and constants.

Expected Output:
Age: 25
Z value: 100
Hello!

```
import "io.kik";

int kik() {
    int age = 25;
    float pi = 3.14;
    bool isKik = true;

    int x, y, z = 100;

    constant str GREETING = "Hello!";

    cout << "Age: " << age << endl;
    cout << "Z value: " << z << endl;
    cout << GREETING << endl;

    return 0;
}
```

Test Case 4: Operator Expressions

This test checks the evaluation of an arithmetic expression and a comparison operator within an if statement.

Expected Output:
Expression evaluated correctly!

```
import "io.kik";

int kik() {
    int a = 10;
    int b = 5;

    int result = (a * 2) + b;

    if result == 25: {
        cout << "Expression evaluated correctly!" << endl;
    } else: {
        cout << "Expression evaluation failed." << endl;
    }

    return 0;
}
```

Test Case 5: Standard Input/Output (cin, cout)

This test demonstrates how to prompt a user for input and read it using cin, then display a formatted message using cout.

Expected Output (Interactive):
The program will prompt for a name and age, then print a personalized greeting.

```
import "io.kik";

int kik() {
    int userAge;
    str userName;

    output("Enter your name: ");
    cin >> userName;

    output("Enter your age: ");
    cin >> userAge;

    cout << "Hello, " << userName << "! You are " << userAge << " years old." << endl;

    return 0;
}
```

Test Case 6: do-while Loop

This test verifies the unique syntax of the KIK do-while loop, which requires an empty block {} after the condition.

Expected Output:
Starting do-while loop...
Counter is: 0
Counter is: 1
Counter is: 2
Loop finished.

```
import "io.kik";

int kik() {
    int counter = 0;

    output("Starting do-while loop...");

    do {
        cout << "Counter is: " << counter << endl;
        counter++;
    } while counter < 3: { }

    output("Loop finished.");

    return 0;
}
```

Test Case 7: Basic Output Function

This test demonstrates the use of the output() function to print variables of different types.

Expected Output:
The score is: 100
Welcome to KIK!

```
import "io.kik";

int kik() {
    int score = 100;
    str message = "Welcome to KIK!";

    output("The score is: ");
    output(score);
    output("\n");
    output(message);

    return 0;
}
```

Test Case 8: Simple if-else Statement

This test checks a basic if-else condition without parentheses around the expression.

Expected Output:
The number is positive.

```
import "io.kik";

int kik() {
    int number = 10;

    if number > 0: {
        output("The number is positive.");
    } else: {
        output("The number is not positive.");
    }

    return 0;
}
```

Test Case 9: for Loop

This test verifies the specific colon-based syntax of the KIK for loop.

Expected Output:
Counting: 0 1 2 3 4

```
import "io.kik";

int kik() {
    output("Counting: ");
    for int i = 0; i < 5;: i = i + 1: {
        output(i);
        output(" ");
    }

    return 0;
}
```

Test Case 10: while Loop

This test demonstrates a basic while loop, showing a countdown.

Expected Output:
3 2 1 Liftoff!

```
import "io.kik";

int kik() {
    int countdown = 3;

    while countdown > 0: {
        output(countdown);
        output(" ");
        countdown = countdown - 1;
    }

    output("Liftoff!");

    return 0;
}
```

Test Case 11: Function Definition and Call

This test shows how to define a separate function (multiply) and call it from the main kik function.

Expected Output:
The product is: 12

```
import "io.kik";

int multiply(int x, int y) {
    return x * y;
}

int kik() {
    output("The product is: ");
    output(multiply(3, 4));

    return 0;
}
```