```
/*Mochire Boaz Momanyi
C++ code on functions
BSE-05-0005/2024
25 Sunday 2025
Version 2*/
#include <iostream>
#include <cmath>

using namespace std;

// function prototype
int interest(int principle, int time, int rate);
float division(float a, float b);
int addition(int x, int y);
int product(int x, int y);

// New functions for log and tan
double naturalLog(double val);
double tangent(double angle_radians);

int main() {
```

```
    int amt_interest, principle, rate, time;
    float div_result;

    cout << "functions in C++" << endl;
    // Calling functions-arguments
    cout << "Enter the principle: " << endl;
    cin >> principle;

    cout << "Enter the time: " << endl;
    cin >> time;

    cout << "Enter the rate: " << endl;
    cin >> rate;

    amt_interest = interest(principle, rate,
time);
    cout << "The interest is :" << amt_interest
<< endl;

    // Example calls for existing functions
    cout << "Division of 10 and 2 is: " <<
```

```
division(10.0, 2.0) << endl;
    cout << "Addition of 5 and 7 is: " <<
addition(5, 7) << endl;
    cout << "Product of 3 and 4 is: " <<
product(3, 4) << endl;

    // Recalling the new log and tan
functions
    double test_value_log = 10.0;
    cout << "Natural log of " <<
test_value_log << " is: " <<
naturalLog(test_value_log) << endl;

    //tan function expects radians. M_PI is
from cmath for pi.
    double test_angle_degrees = 45.0;
    double test_angle_radians =
test_angle_degrees * M_PI / 180.0; //
Convert degrees to radians
    cout << "Tangent of " <<
test_angle_degrees << " degrees (" <<
```

```
test_angle_radians << " radians) is: " <<
tangent(test_angle_radians) << endl;

    return 0;
}

// Function definitions
int interest(int principle, int rate, int time) {
    return (principle * rate * time) / 100;
}

float division(float a, float b) {
    return a / b;
}

int addition(int x, int y) {
    return x + y;
}

int product(int x, int y) {
    return x * y;
```

```
}

// Function definitions for new log and tan
functions
double naturalLog(double val) {
    return log(val);
}

double tangent(double angle_radians) {
    return tan(angle_radians); // tan()
calculates tangent, expects radians
}
```