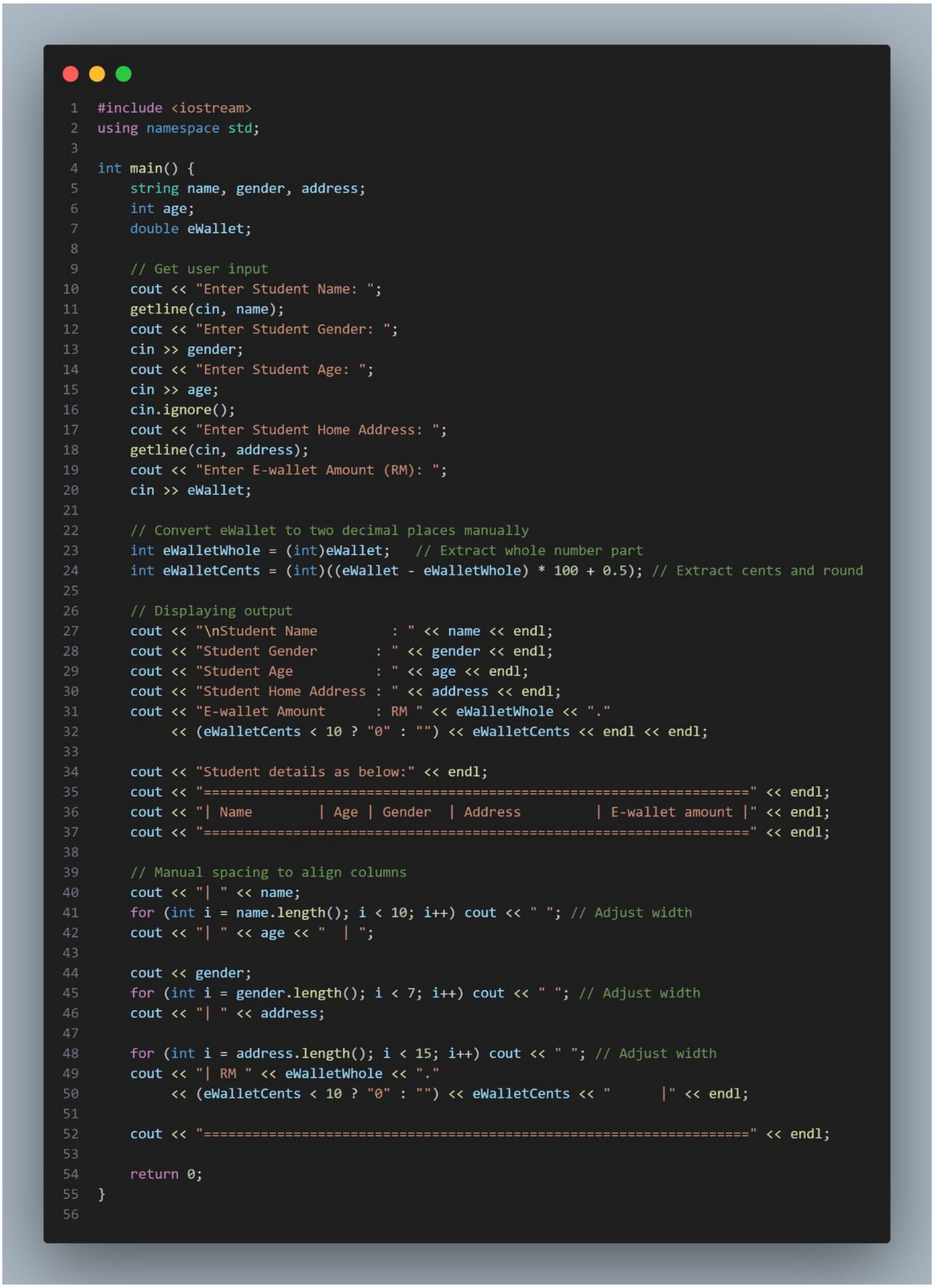
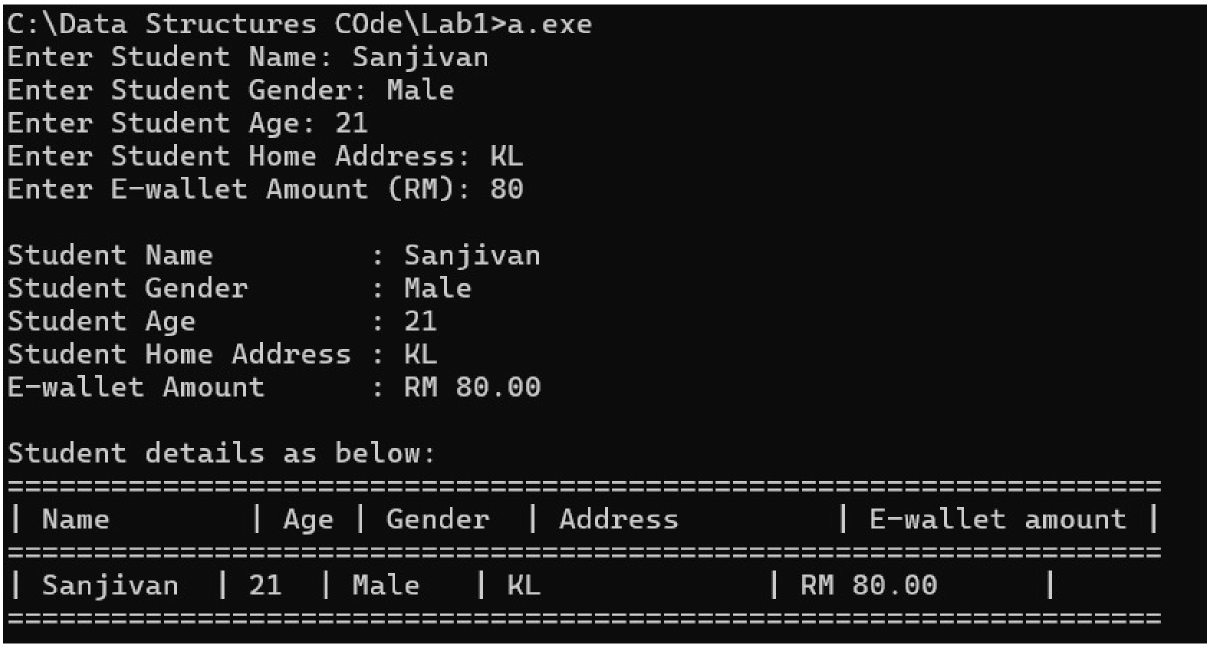
Part A

1)



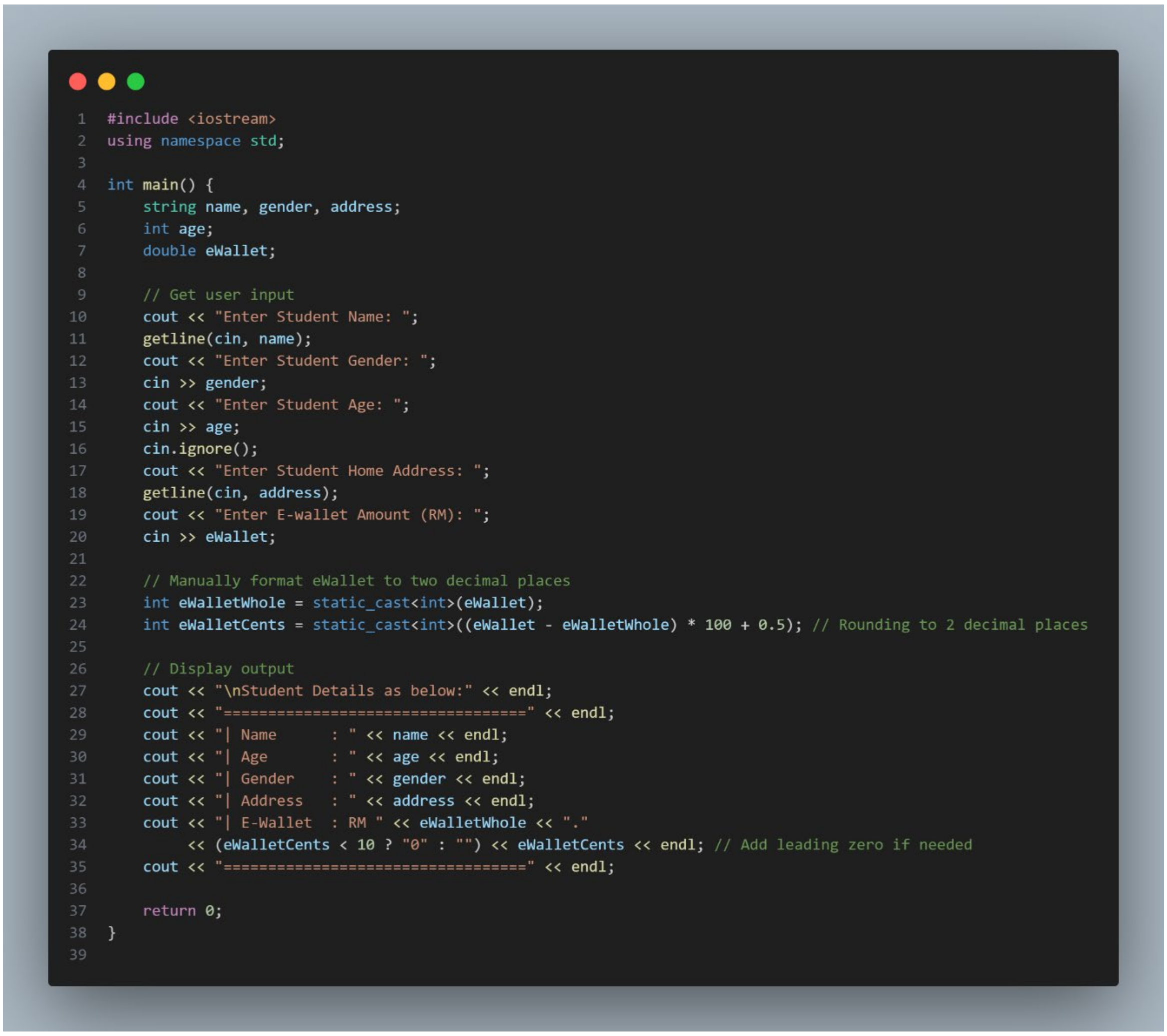
```cpp
#include <iostream>
using namespace std;

int main() {
    string name, gender, address;
    int age;
    double eWallet;

    // Get user input
    cout << "Enter Student Name: ";
    getline(cin, name);
    cout << "Enter Student Gender: ";
    cin >> gender;
    cout << "Enter Student Age: ";
    cin >> age;
    cin.ignore();
    cout << "Enter Student Home Address: ";
    getline(cin, address);
    cout << "Enter E-wallet Amount (RM): ";
    cin >> eWallet;

    // Convert eWallet to two decimal places manually
    int eWalletWhole = (int)eWallet;   // Extract whole number part
    int eWalletCents = (int)((eWallet - eWalletWhole) * 100 + 0.5); // Extract cents and round

    // Displaying output
    cout << "\nStudent Name         : " << name << endl;
    cout << "Student Gender       : " << gender << endl;
    cout << "Student Age          : " << age << endl;
    cout << "Student Home Address : " << address << endl;
    cout << "E-wallet Amount      : RM " << eWalletWhole << "."
         << (eWalletCents < 10 ? "0" : "") << eWalletCents << endl << endl;

    cout << "Student details as below:" << endl;
    cout << "===================================================================" << endl;
    cout << "| Name        | Age | Gender  | Address         | E-wallet amount |" << endl;
    cout << "===================================================================" << endl;

    // Manual spacing to align columns
    cout << "| " << name;
    for (int i = name.length(); i < 10; i++) cout << " "; // Adjust width
    cout << "| " << age << "  | ";

    cout << gender;
    for (int i = gender.length(); i < 7; i++) cout << " "; // Adjust width
    cout << "| " << address;

    for (int i = address.length(); i < 15; i++) cout << " "; // Adjust width
    cout << "| RM " << eWalletWhole << "."
         << (eWalletCents < 10 ? "0" : "") << eWalletCents << "      |" << endl;

    cout << "===================================================================" << endl;

    return 0;
}
```

Output:



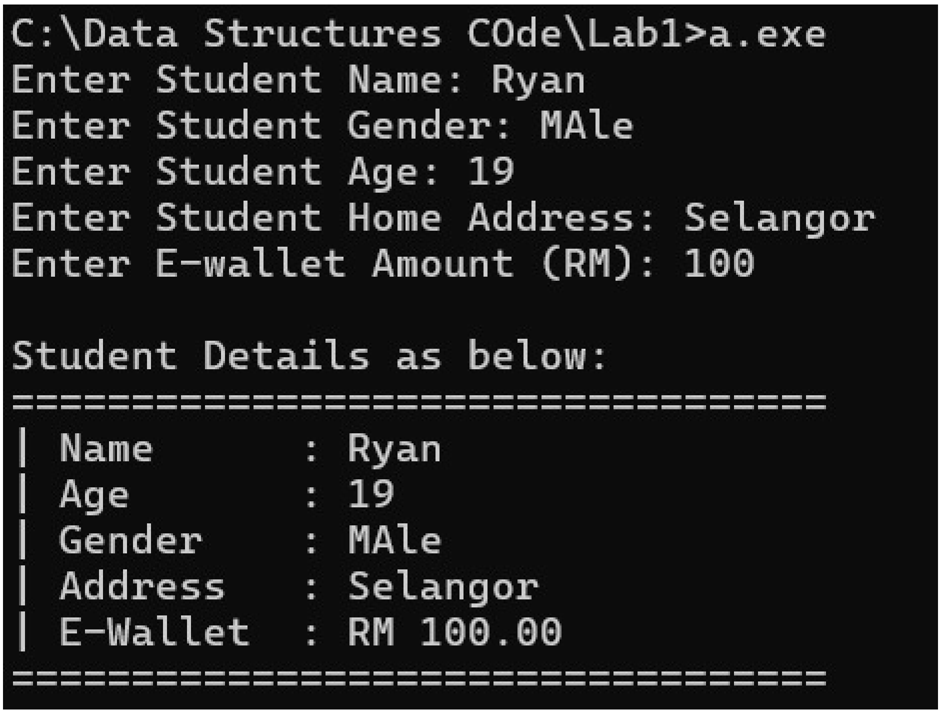
```
C:\Data Structures COde\Lab1>a.exe
Enter Student Name: Sanjivan
Enter Student Gender: Male
Enter Student Age: 21
Enter Student Home Address: KL
Enter E-wallet Amount (RM): 80

Student Name         : Sanjivan
Student Gender       : Male
Student Age          : 21
Student Home Address : KL
E-wallet Amount      : RM 80.00

Student details as below:
===================================================================
| Name          | Age | Gender   | Address         | E-wallet amount |
===================================================================
| Sanjivan  | 21  | Male    | KL              | RM 80.00       |
===================================================================
```

Q2



```cpp
#include <iostream>
using namespace std;

int main() {
    string name, gender, address;
    int age;
    double eWallet;

    // Get user input
    cout << "Enter Student Name: ";
    getline(cin, name);
    cout << "Enter Student Gender: ";
    cin >> gender;
    cout << "Enter Student Age: ";
    cin >> age;
    cin.ignore();
    cout << "Enter Student Home Address: ";
    getline(cin, address);
    cout << "Enter E-wallet Amount (RM): ";
    cin >> eWallet;

    // Manually format eWallet to two decimal places
    int eWalletWhole = static_cast<int>(eWallet);
    int eWalletCents = static_cast<int>((eWallet - eWalletWhole) * 100 + 0.5); // Rounding to 2 decimal places

    // Display output
    cout << "\nStudent Details as below:" << endl;
    cout << "==================================" << endl;
    cout << "| Name      : " << name << endl;
    cout << "| Age       : " << age << endl;
    cout << "| Gender    : " << gender << endl;
    cout << "| Address   : " << address << endl;
    cout << "| E-Wallet  : RM " << eWalletWhole << "."
         << (eWalletCents < 10 ? "0" : "") << eWalletCents << endl; // Add leading zero if needed
    cout << "==================================" << endl;

    return 0;
}
```

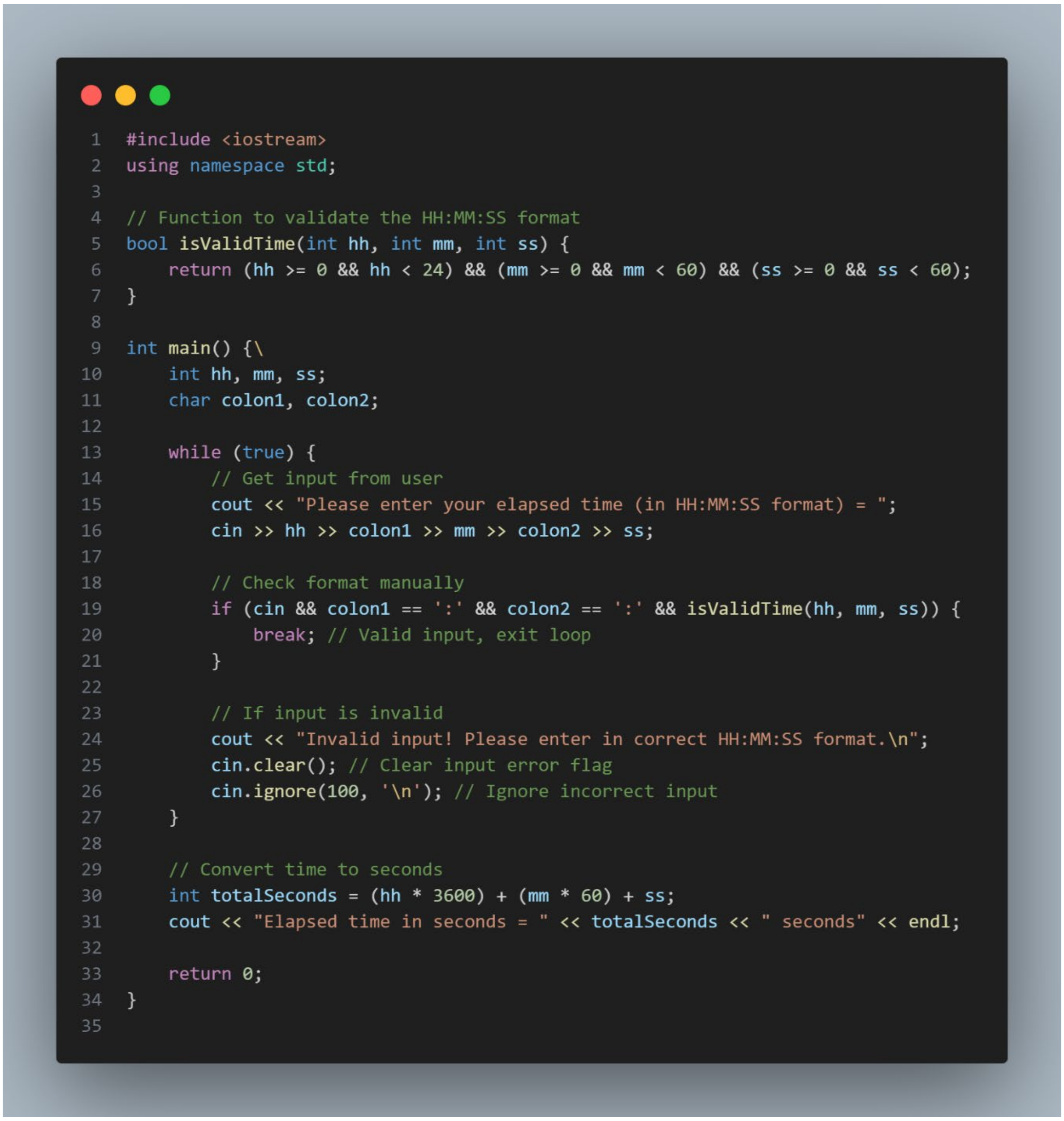
Output:



```
C:\Data Structures COde\Lab1>a.exe
Enter Student Name: Ryan
Enter Student Gender: MAle
Enter Student Age: 19
Enter Student Home Address: Selangor
Enter E-wallet Amount (RM): 100

Student Details as below:
==================================
| Name      : Ryan
| Age       : 19
| Gender    : MAle
| Address   : Selangor
| E-Wallet  : RM 100.00
==================================
```

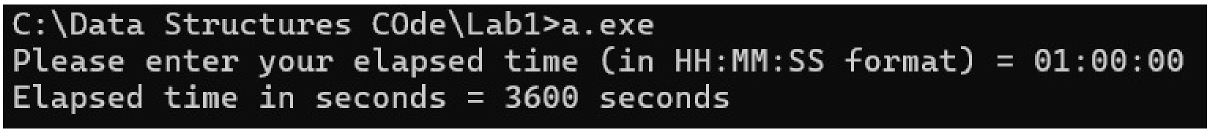
Part B

Q1:



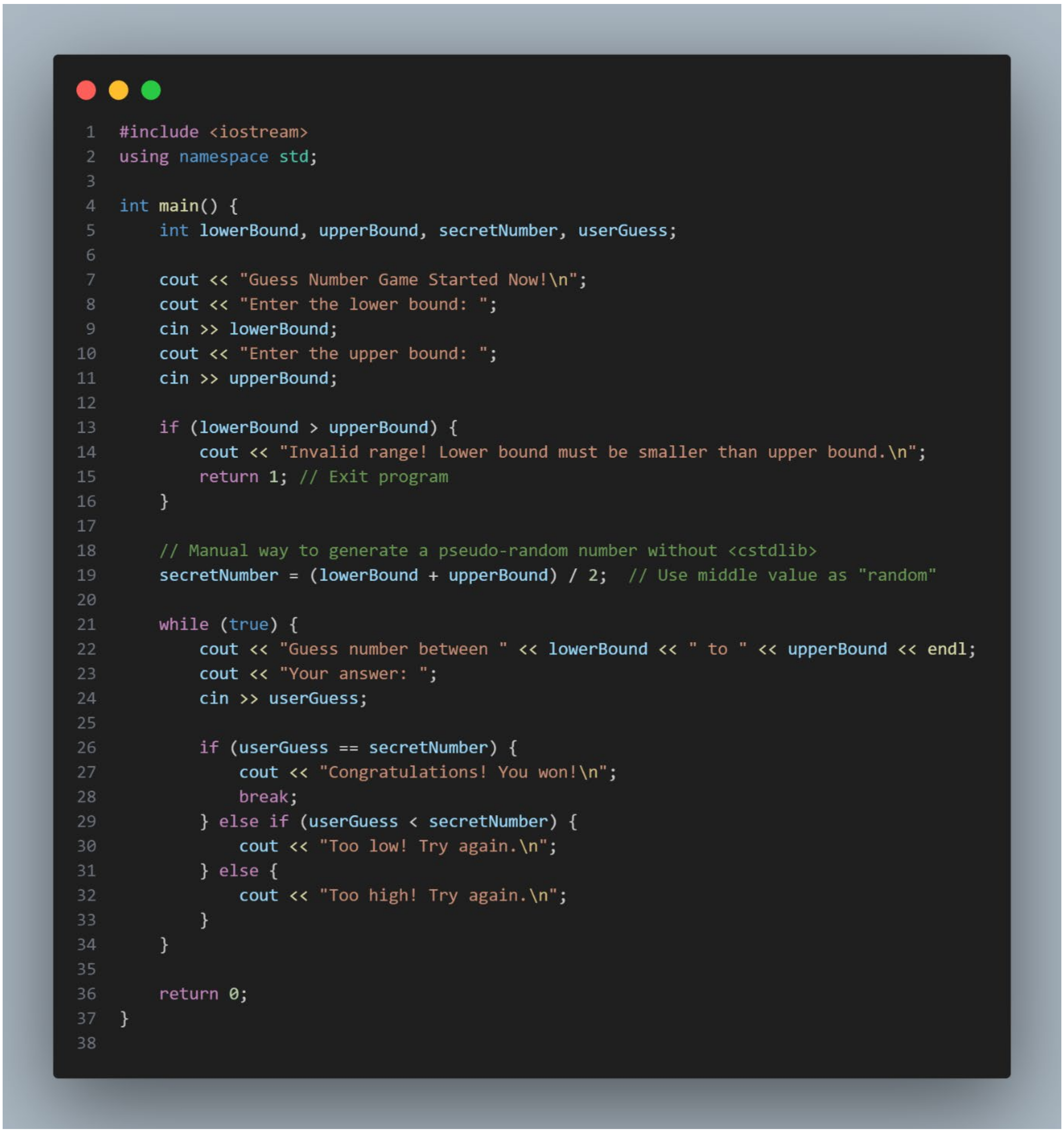
```cpp
#include <iostream>
using namespace std;

// Function to validate the HH:MM:SS format
bool isValidTime(int hh, int mm, int ss) {
    return (hh >= 0 && hh < 24) && (mm >= 0 && mm < 60) && (ss >= 0 && ss < 60);
}

int main() {\
    int hh, mm, ss;
    char colon1, colon2;

    while (true) {
        // Get input from user
        cout << "Please enter your elapsed time (in HH:MM:SS format) = ";
        cin >> hh >> colon1 >> mm >> colon2 >> ss;

        // Check format manually
        if (cin && colon1 == ':' && colon2 == ':' && isValidTime(hh, mm, ss)) {
            break; // Valid input, exit loop
        }

        // If input is invalid
        cout << "Invalid input! Please enter in correct HH:MM:SS format.\n";
        cin.clear(); // Clear input error flag
        cin.ignore(100, '\n'); // Ignore incorrect input
    }

    // Convert time to seconds
    int totalSeconds = (hh * 3600) + (mm * 60) + ss;
    cout << "Elapsed time in seconds = " << totalSeconds << " seconds" << endl;

    return 0;
}

```

Output:



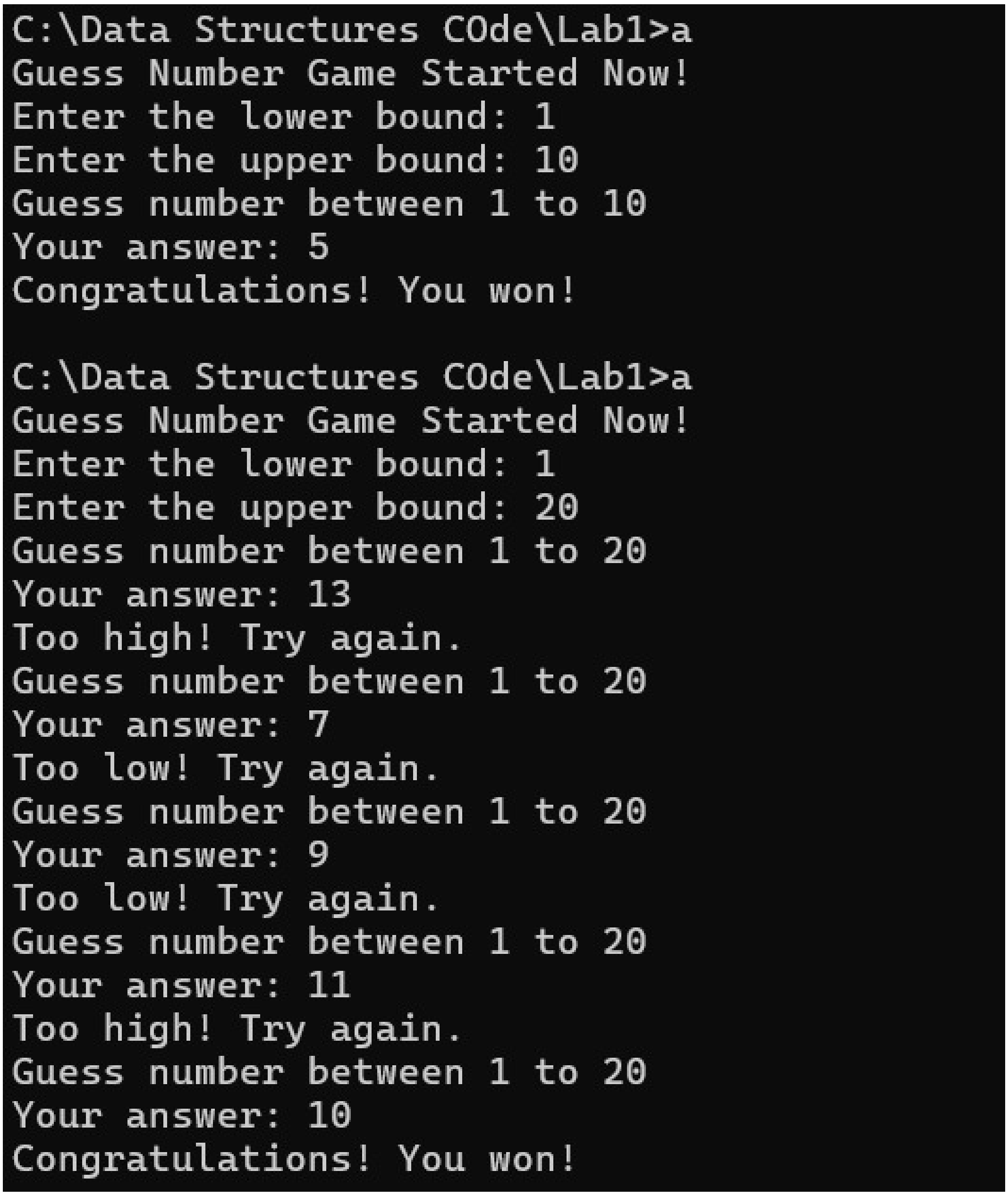
```
C:\Data Structures COde\Lab1>a.exe
Please enter your elapsed time (in HH:MM:SS format) = 01:00:00
Elapsed time in seconds = 3600 seconds
```

Q2:



```cpp
#include <iostream>
using namespace std;

int main() {
    int lowerBound, upperBound, secretNumber, userGuess;

    cout << "Guess Number Game Started Now!\n";
    cout << "Enter the lower bound: ";
    cin >> lowerBound;
    cout << "Enter the upper bound: ";
    cin >> upperBound;

    if (lowerBound > upperBound) {
        cout << "Invalid range! Lower bound must be smaller than upper bound.\n";
        return 1; // Exit program
    }

    // Manual way to generate a pseudo-random number without <cstdlib>
    secretNumber = (lowerBound + upperBound) / 2;  // Use middle value as "random"

    while (true) {
        cout << "Guess number between " << lowerBound << " to " << upperBound << endl;
        cout << "Your answer: ";
        cin >> userGuess;

        if (userGuess == secretNumber) {
            cout << "Congratulations! You won!\n";
            break;
        } else if (userGuess < secretNumber) {
            cout << "Too low! Try again.\n";
        } else {
            cout << "Too high! Try again.\n";
        }
    }

    return 0;
}

```

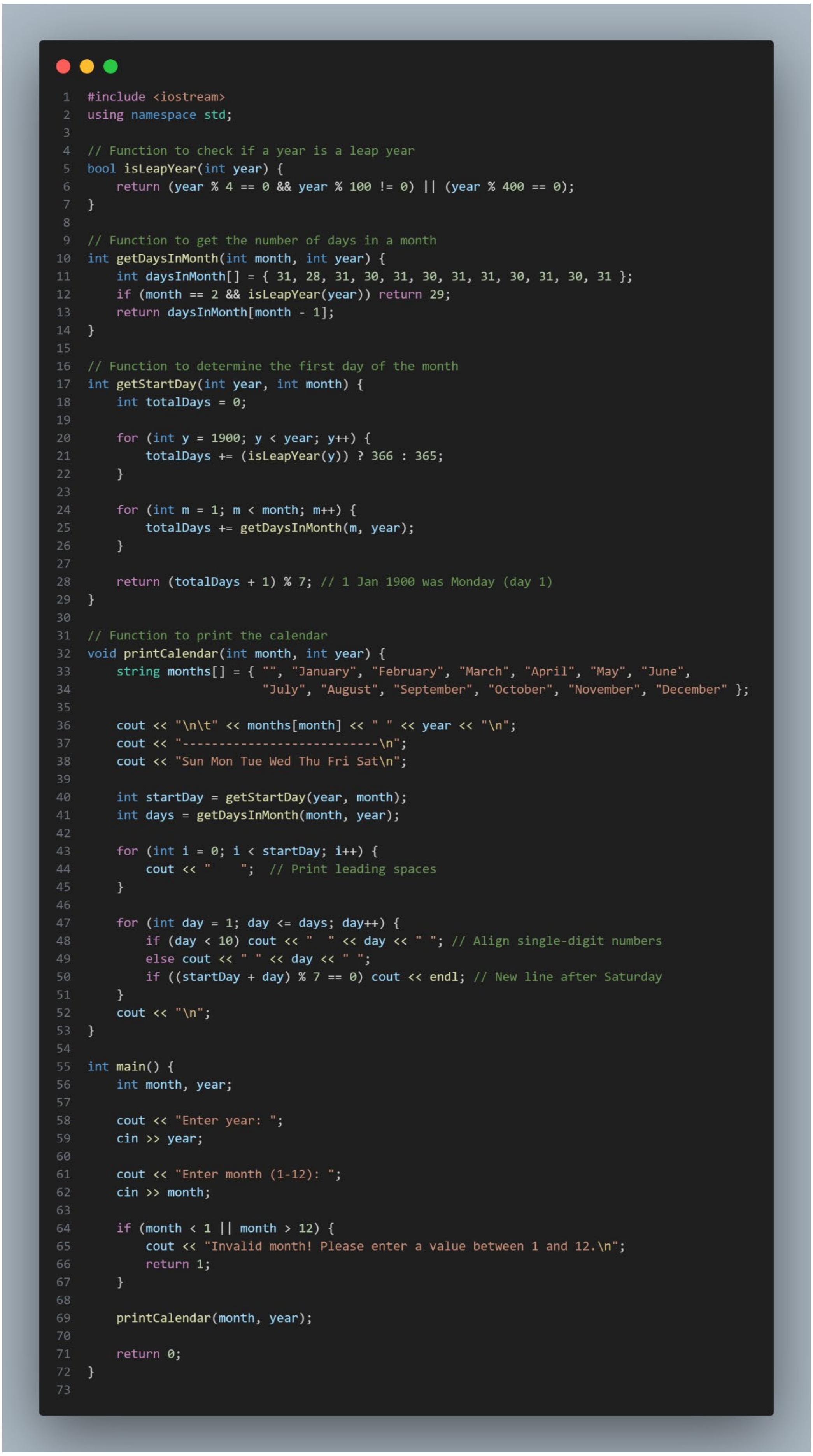
Output:



```
C:\Data Structures COde\Lab1>a
Guess Number Game Started Now!
Enter the lower bound: 1
Enter the upper bound: 10
Guess number between 1 to 10
Your answer: 5
Congratulations! You won!

C:\Data Structures COde\Lab1>a
Guess Number Game Started Now!
Enter the lower bound: 1
Enter the upper bound: 20
Guess number between 1 to 20
Your answer: 13
Too high! Try again.
Guess number between 1 to 20
Your answer: 7
Too low! Try again.
Guess number between 1 to 20
Your answer: 9
Too low! Try again.
Guess number between 1 to 20
Your answer: 11
Too high! Try again.
Guess number between 1 to 20
Your answer: 10
Congratulations! You won!
```

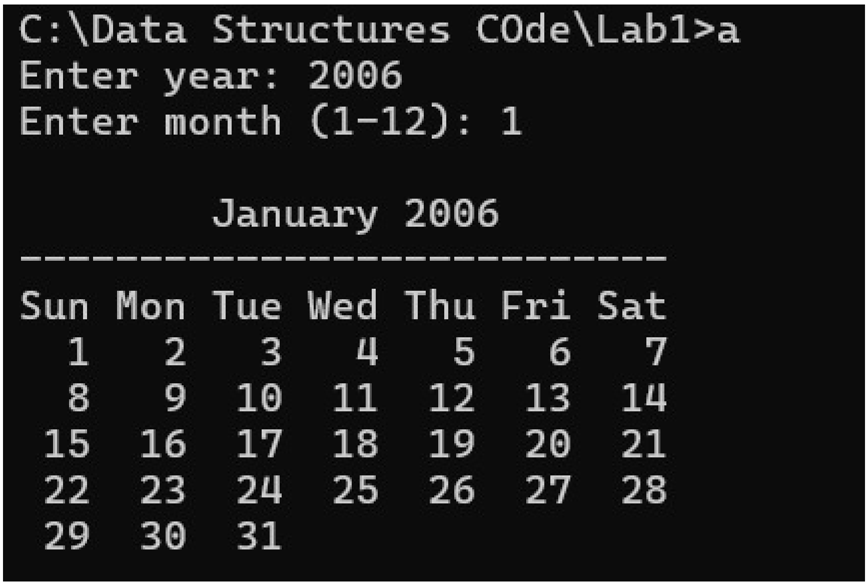
Part C

Q1:



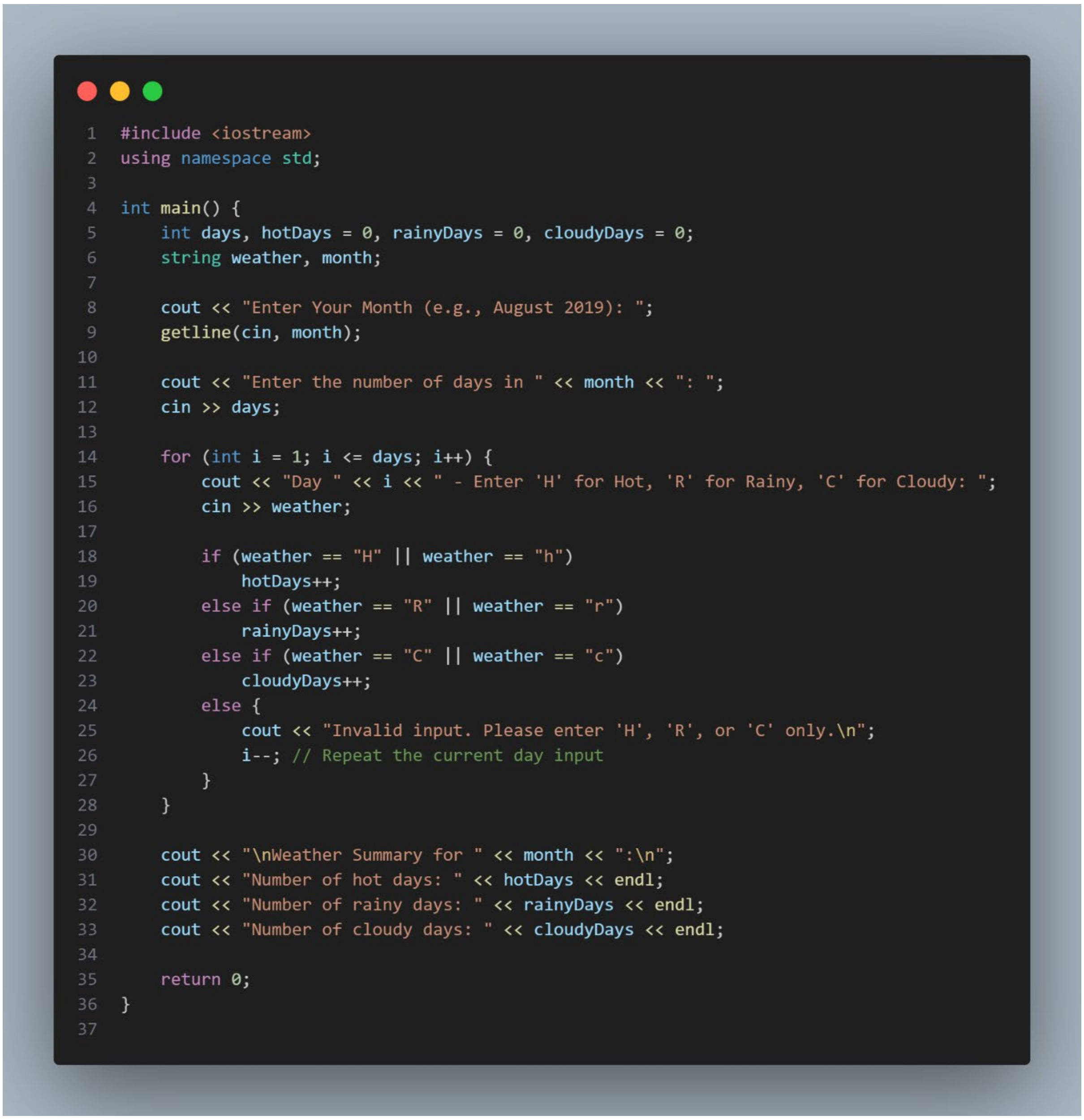
```cpp
#include <iostream>
using namespace std;

// Function to check if a year is a leap year
bool isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}

// Function to get the number of days in a month
int getDaysInMonth(int month, int year) {
    int daysInMonth[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    if (month == 2 && isLeapYear(year)) return 29;
    return daysInMonth[month - 1];
}

// Function to determine the first day of the month
int getStartDay(int year, int month) {
    int totalDays = 0;

    for (int y = 1900; y < year; y++) {
        totalDays += (isLeapYear(y)) ? 366 : 365;
    }

    for (int m = 1; m < month; m++) {
        totalDays += getDaysInMonth(m, year);
    }

    return (totalDays + 1) % 7; // 1 Jan 1900 was Monday (day 1)
}

// Function to print the calendar
void printCalendar(int month, int year) {
    string months[] = { "", "January", "February", "March", "April", "May", "June",
                        "July", "August", "September", "October", "November", "December" };

    cout << "\n\t" << months[month] << " " << year << "\n";
    cout << "---------------------------\n";
    cout << "Sun Mon Tue Wed Thu Fri Sat\n";

    int startDay = getStartDay(year, month);
    int days = getDaysInMonth(month, year);

    for (int i = 0; i < startDay; i++) {
        cout << "    ";  // Print leading spaces
    }

    for (int day = 1; day <= days; day++) {
        if (day < 10) cout << "  " << day << " "; // Align single-digit numbers
        else cout << " " << day << " ";
        if ((startDay + day) % 7 == 0) cout << endl; // New line after Saturday
    }
    cout << "\n";
}

int main() {
    int month, year;

    cout << "Enter year: ";
    cin >> year;

    cout << "Enter month (1-12): ";
    cin >> month;

    if (month < 1 || month > 12) {
        cout << "Invalid month! Please enter a value between 1 and 12.\n";
        return 1;
    }

    printCalendar(month, year);

    return 0;
}
```

Output:



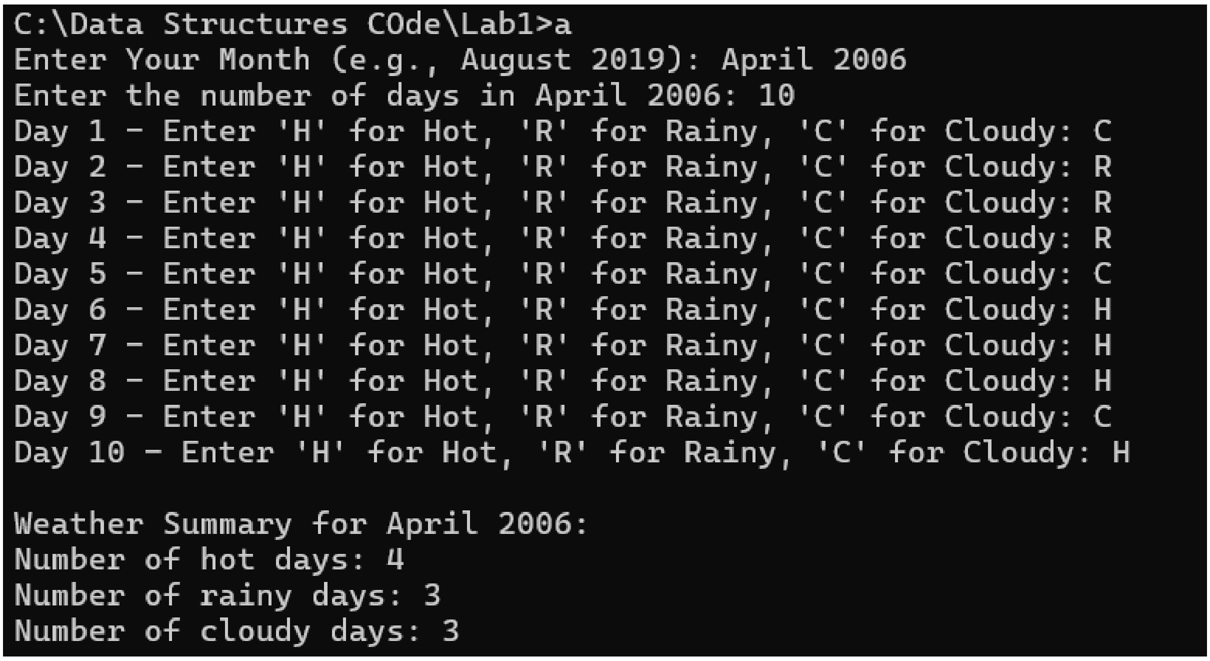
```
C:\Data Structures COde\Lab1>a
Enter year: 2006
Enter month (1-12): 1

        January 2006
---------------------------
Sun Mon Tue Wed Thu Fri Sat
  1   2   3   4   5   6   7
  8   9  10  11  12  13  14
 15  16  17  18  19  20  21
 22  23  24  25  26  27  28
 29  30  31
```

Q2:



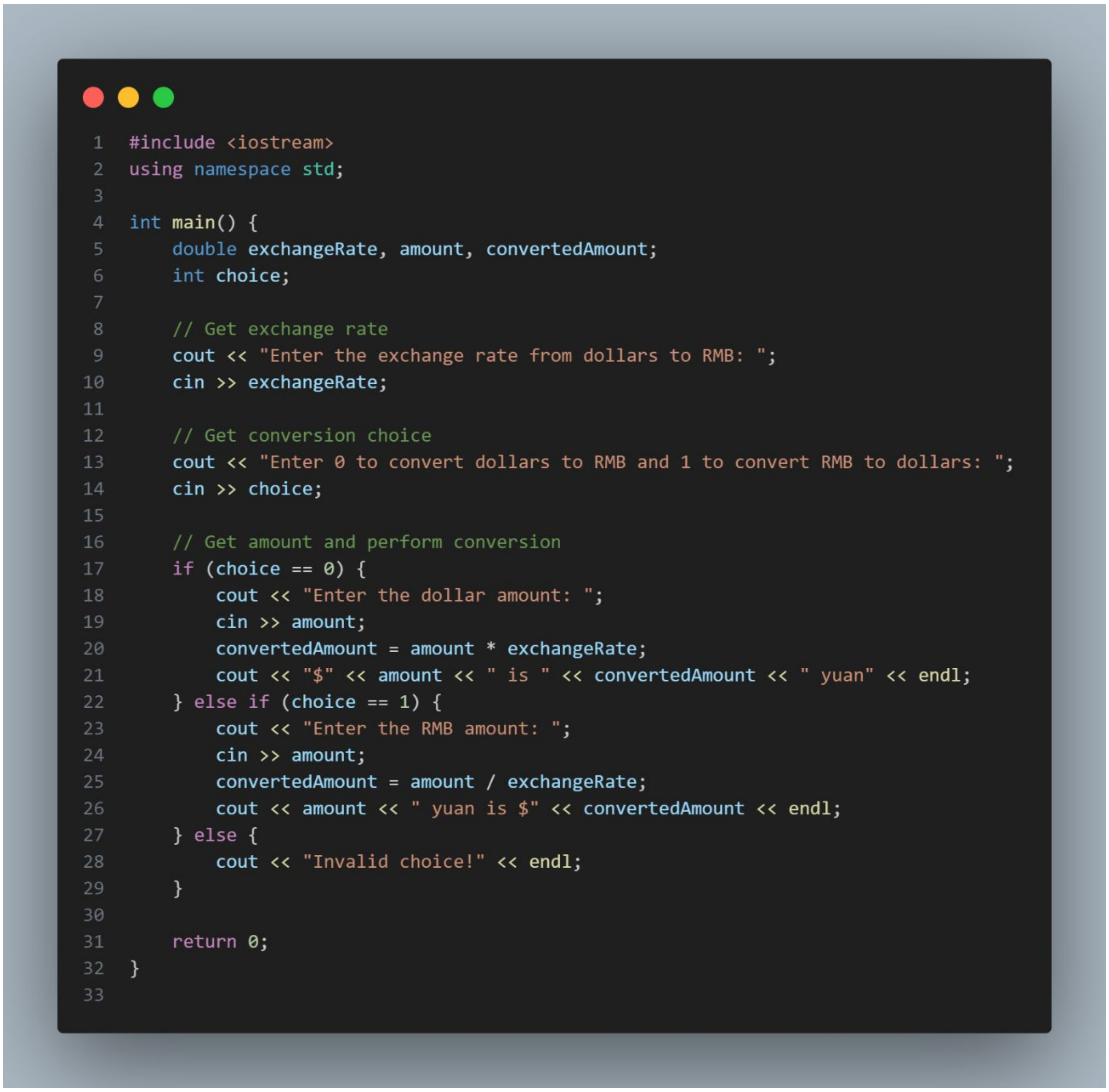
```cpp
#include <iostream>
using namespace std;

int main() {
    int days, hotDays = 0, rainyDays = 0, cloudyDays = 0;
    string weather, month;

    cout << "Enter Your Month (e.g., August 2019): ";
    getline(cin, month);

    cout << "Enter the number of days in " << month << ": ";
    cin >> days;

    for (int i = 1; i <= days; i++) {
        cout << "Day " << i << " - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: ";
        cin >> weather;

        if (weather == "H" || weather == "h")
            hotDays++;
        else if (weather == "R" || weather == "r")
            rainyDays++;
        else if (weather == "C" || weather == "c")
            cloudyDays++;
        else {
            cout << "Invalid input. Please enter 'H', 'R', or 'C' only.\n";
            i--; // Repeat the current day input
        }
    }

    cout << "\nWeather Summary for " << month << ":\n";
    cout << "Number of hot days: " << hotDays << endl;
    cout << "Number of rainy days: " << rainyDays << endl;
    cout << "Number of cloudy days: " << cloudyDays << endl;

    return 0;
}

```

Output:



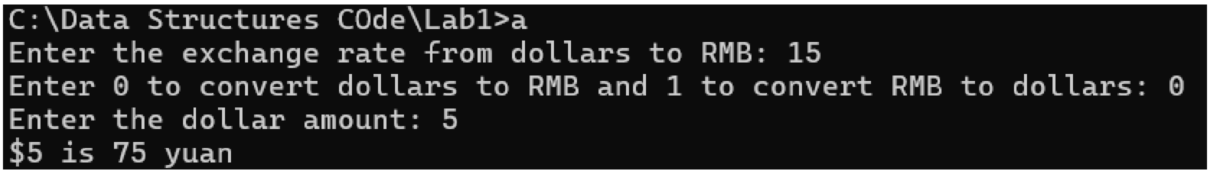
```
C:\Data Structures COde\Lab1>a
Enter Your Month (e.g., August 2019): April 2006
Enter the number of days in April 2006: 10
Day 1 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: C
Day 2 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: R
Day 3 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: R
Day 4 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: R
Day 5 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: C
Day 6 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: H
Day 7 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: H
Day 8 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: H
Day 9 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: C
Day 10 - Enter 'H' for Hot, 'R' for Rainy, 'C' for Cloudy: H

Weather Summary for April 2006:
Number of hot days: 4
Number of rainy days: 3
Number of cloudy days: 3
```

Q3:



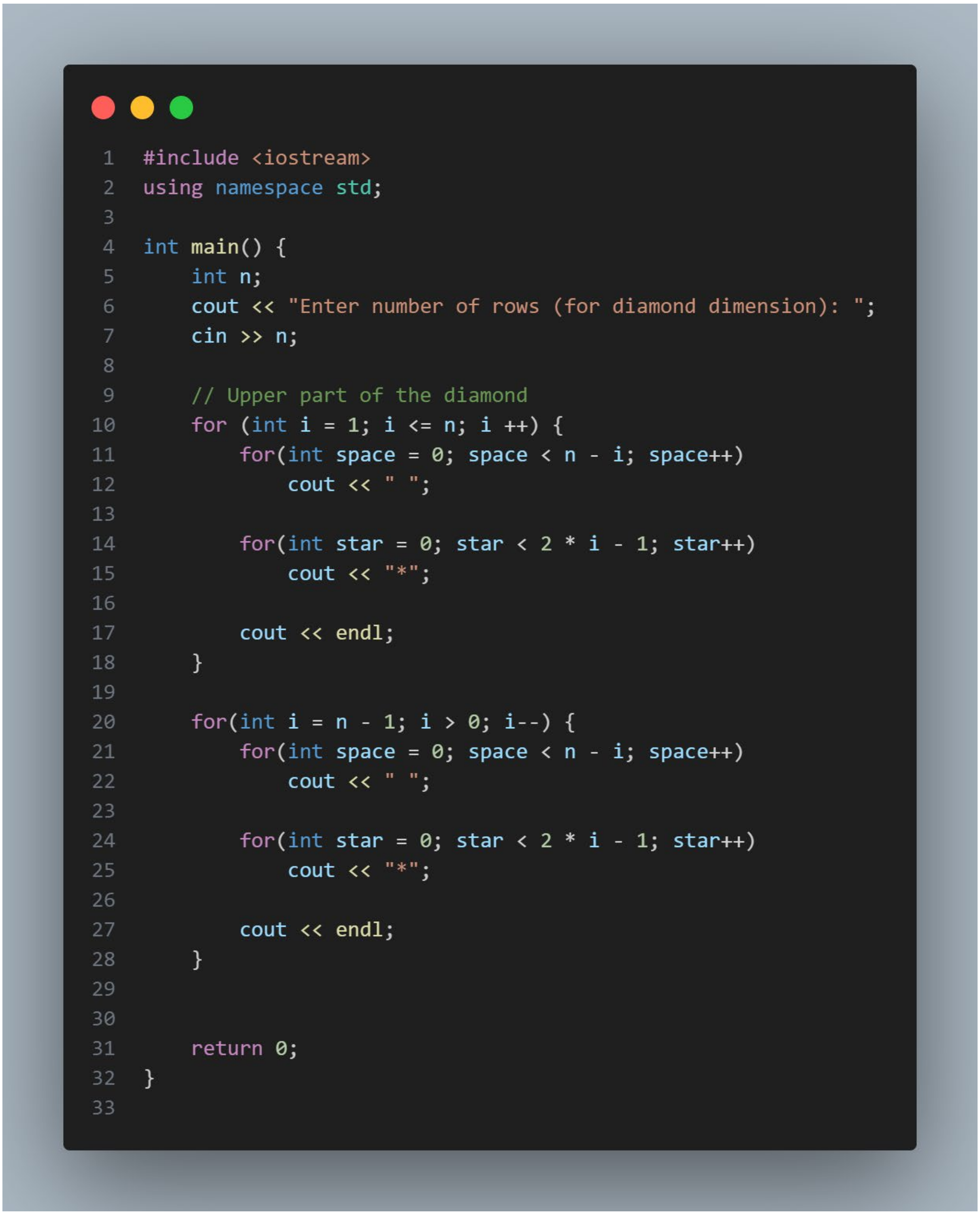
```cpp
#include <iostream>
using namespace std;

int main() {
    double exchangeRate, amount, convertedAmount;
    int choice;

    // Get exchange rate
    cout << "Enter the exchange rate from dollars to RMB: ";
    cin >> exchangeRate;

    // Get conversion choice
    cout << "Enter 0 to convert dollars to RMB and 1 to convert RMB to dollars: ";
    cin >> choice;

    // Get amount and perform conversion
    if (choice == 0) {
        cout << "Enter the dollar amount: ";
        cin >> amount;
        convertedAmount = amount * exchangeRate;
        cout << "$" << amount << " is " << convertedAmount << " yuan" << endl;
    } else if (choice == 1) {
        cout << "Enter the RMB amount: ";
        cin >> amount;
        convertedAmount = amount / exchangeRate;
        cout << amount << " yuan is $" << convertedAmount << endl;
    } else {
        cout << "Invalid choice!" << endl;
    }

    return 0;
}

```

Output:



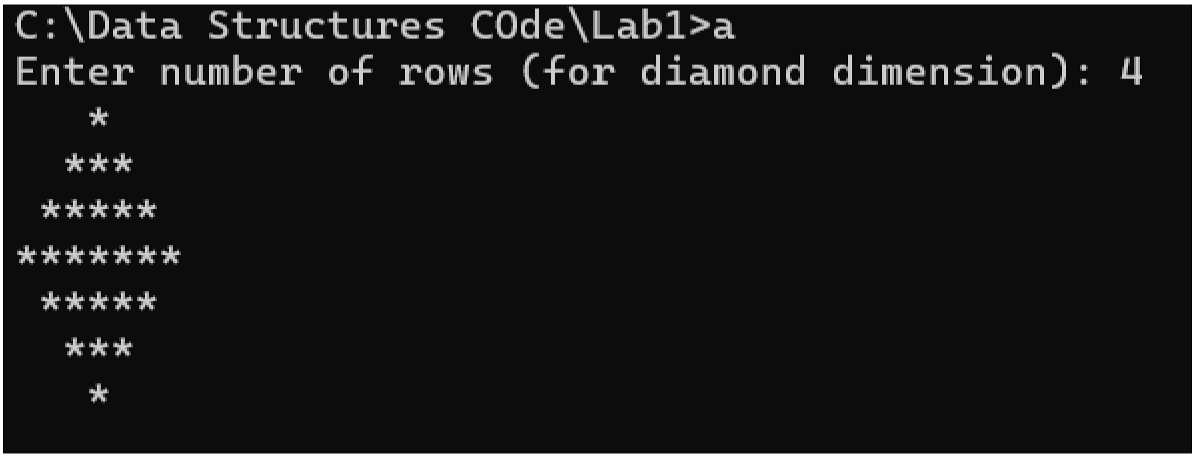
```
C:\Data Structures COde\Lab1>a
Enter the exchange rate from dollars to RMB: 15
Enter 0 to convert dollars to RMB and 1 to convert RMB to dollars: 0
Enter the dollar amount: 5
$5 is 75 yuan
```

Q4:



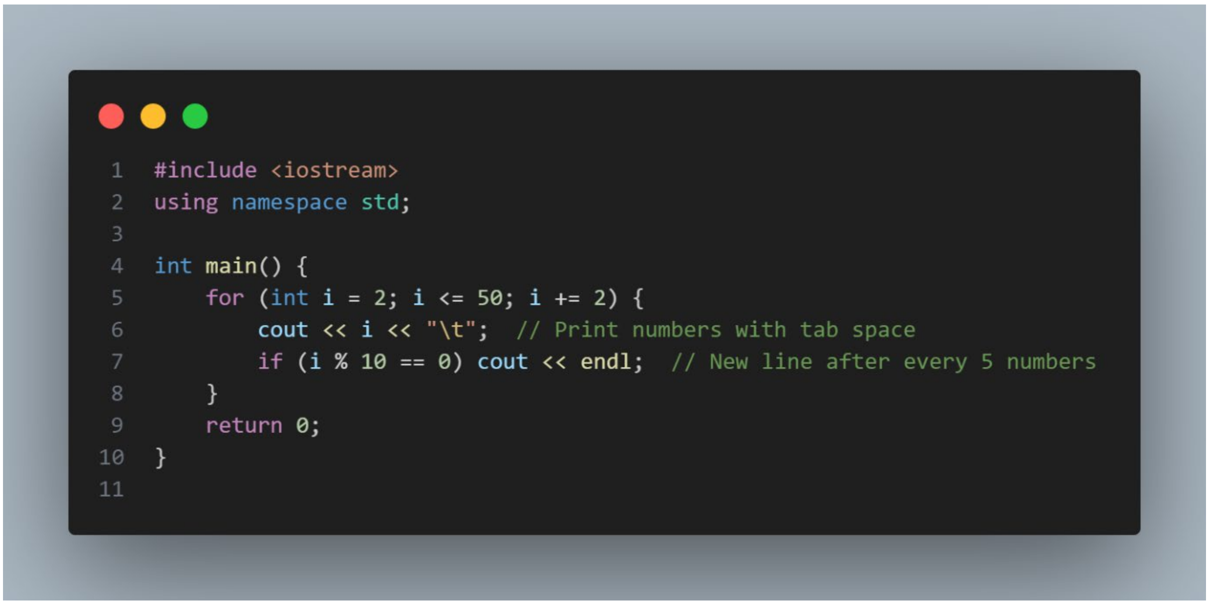
```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Enter number of rows (for diamond dimension): ";
    cin >> n;

    // Upper part of the diamond
    for (int i = 1; i <= n; i ++) {
        for(int space = 0; space < n - i; space++)
            cout << " ";

        for(int star = 0; star < 2 * i - 1; star++)
            cout << "*";

        cout << endl;
    }

    for(int i = n - 1; i > 0; i--) {
        for(int space = 0; space < n - i; space++)
            cout << " ";

        for(int star = 0; star < 2 * i - 1; star++)
            cout << "*";

        cout << endl;
    }


    return 0;
}
```

Output:



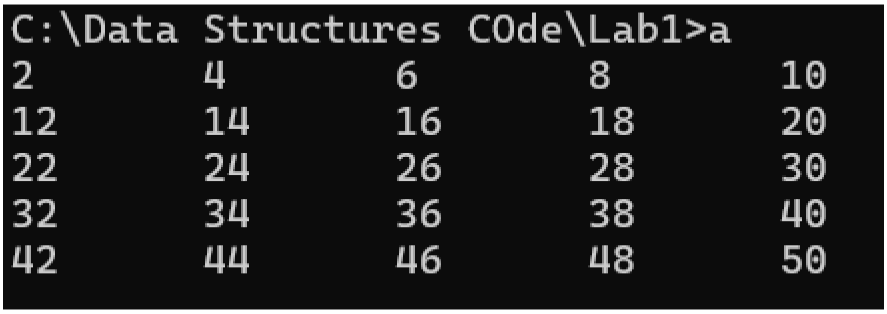
```
C:\Data Structures COde\Lab1>a
Enter number of rows (for diamond dimension): 4
   *
  ***
 *****
*******
 *****
  ***
   *
```

Q5:



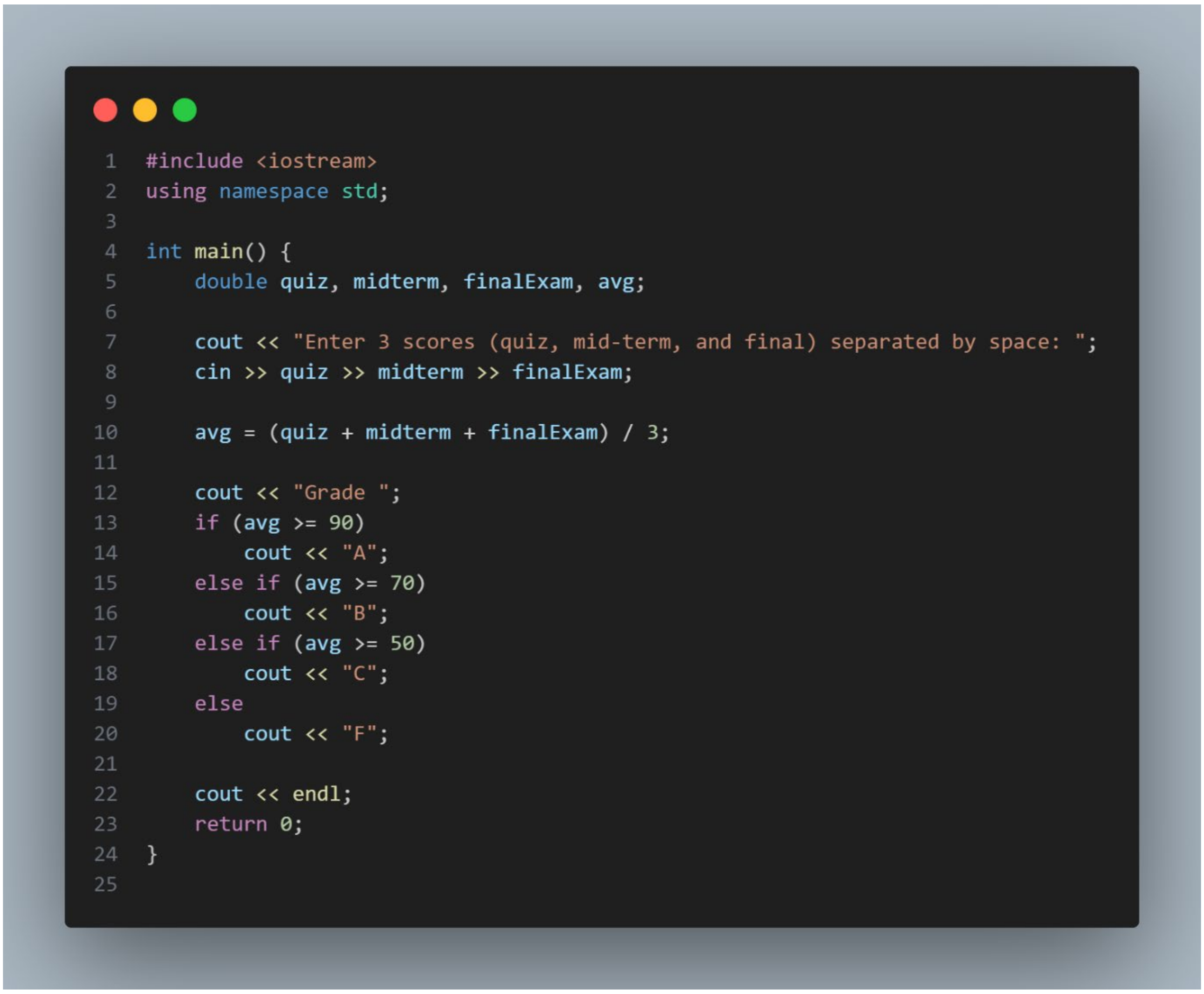
```cpp
#include <iostream>
using namespace std;

int main() {
    for (int i = 2; i <= 50; i += 2) {
        cout << i << "\t";  // Print numbers with tab space
        if (i % 10 == 0) cout << endl;  // New line after every 5 numbers
    }
    return 0;
}
```

Output:



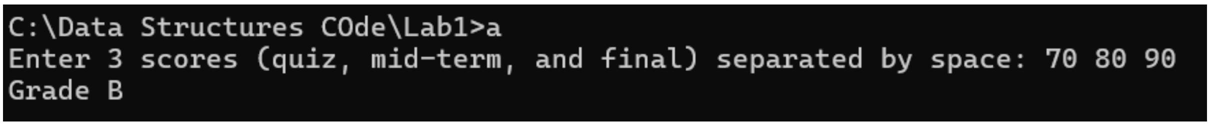
```
C:\Data Structures COde\Lab1>a
2       4       6       8       10
12      14      16      18      20
22      24      26      28      30
32      34      36      38      40
42      44      46      48      50
```

Q6:



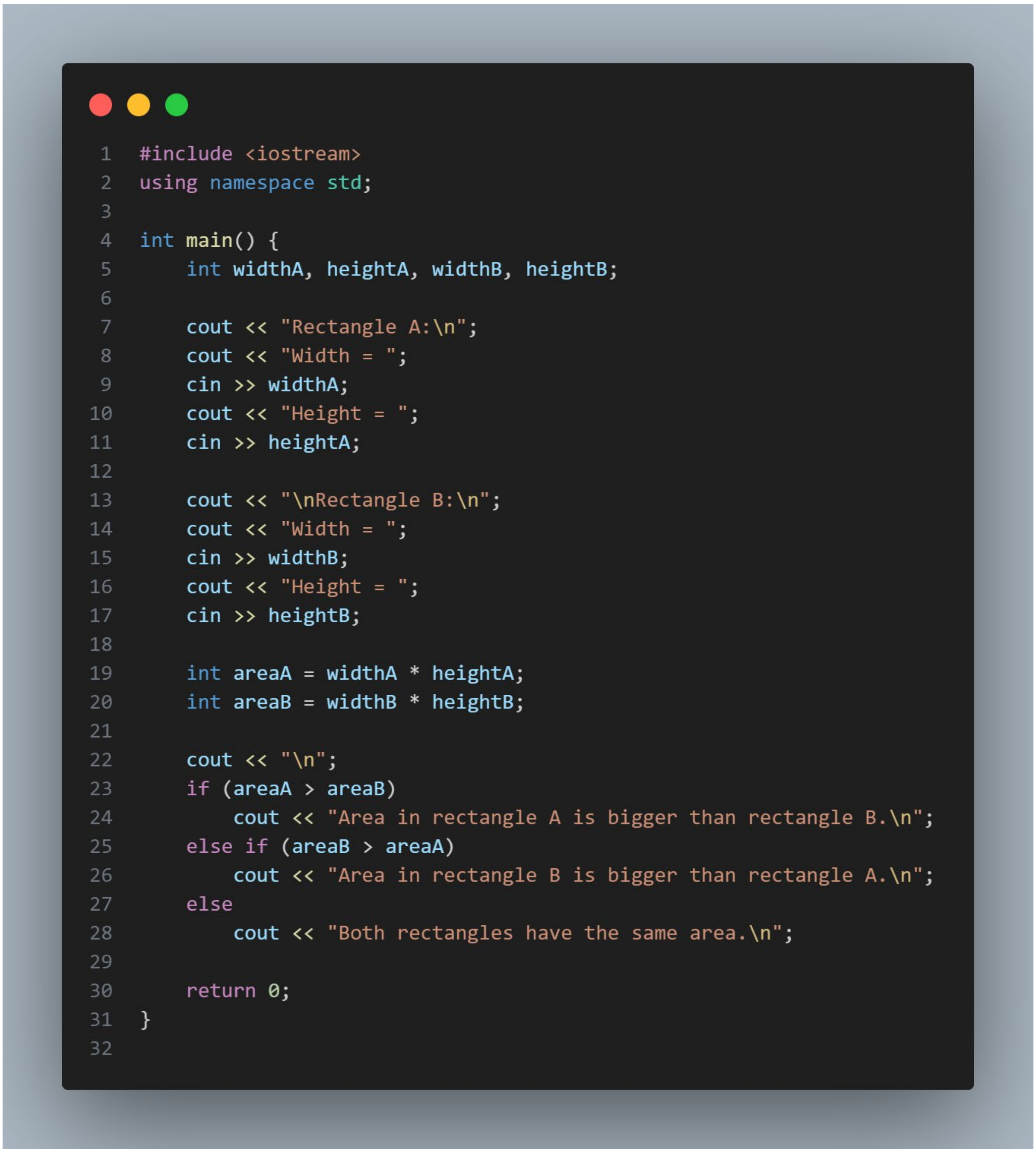
```cpp
#include <iostream>
using namespace std;

int main() {
    double quiz, midterm, finalExam, avg;

    cout << "Enter 3 scores (quiz, mid-term, and final) separated by space: ";
    cin >> quiz >> midterm >> finalExam;

    avg = (quiz + midterm + finalExam) / 3;

    cout << "Grade ";
    if (avg >= 90)
        cout << "A";
    else if (avg >= 70)
        cout << "B";
    else if (avg >= 50)
        cout << "C";
    else
        cout << "F";

    cout << endl;
    return 0;
}

```

Output:



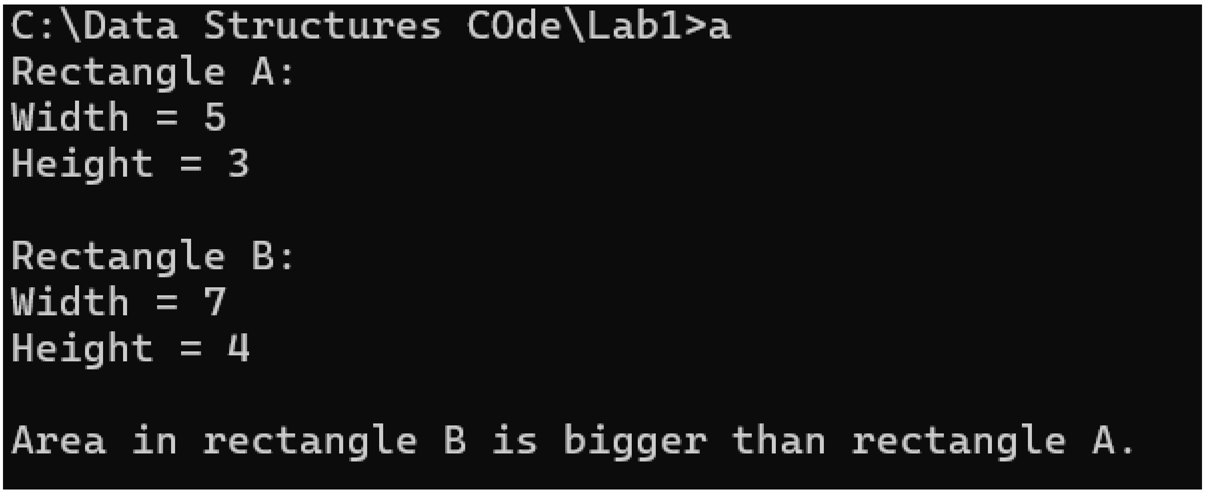
```
C:\Data Structures COde\Lab1>a
Enter 3 scores (quiz, mid-term, and final) separated by space: 70 80 90
Grade B
```

Q7:



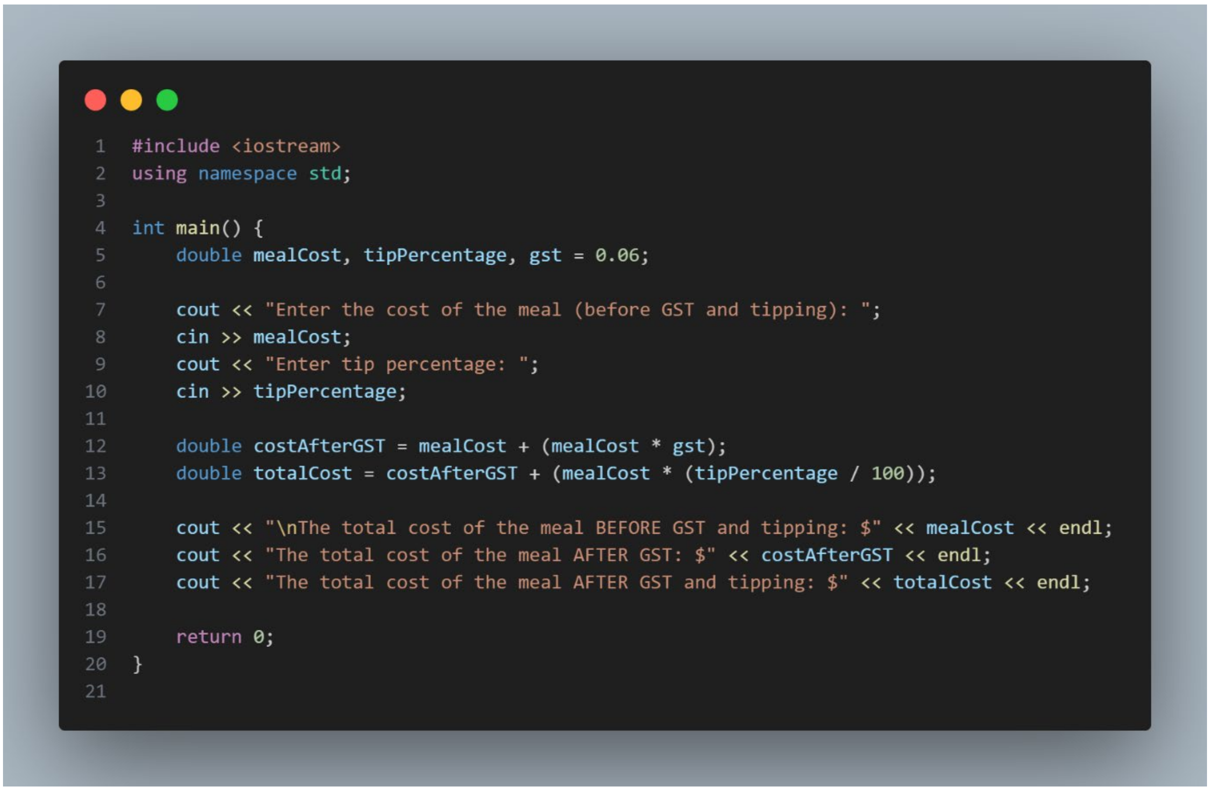
```
#include <iostream>
using namespace std;

int main() {
    int widthA, heightA, widthB, heightB;

    cout << "Rectangle A:\n";
    cout << "Width = ";
    cin >> widthA;
    cout << "Height = ";
    cin >> heightA;

    cout << "\nRectangle B:\n";
    cout << "Width = ";
    cin >> widthB;
    cout << "Height = ";
    cin >> heightB;

    int areaA = widthA * heightA;
    int areaB = widthB * heightB;

    cout << "\n";
    if (areaA > areaB)
        cout << "Area in rectangle A is bigger than rectangle B.\n";
    else if (areaB > areaA)
        cout << "Area in rectangle B is bigger than rectangle A.\n";
    else
        cout << "Both rectangles have the same area.\n";

    return 0;
}

```

Output:



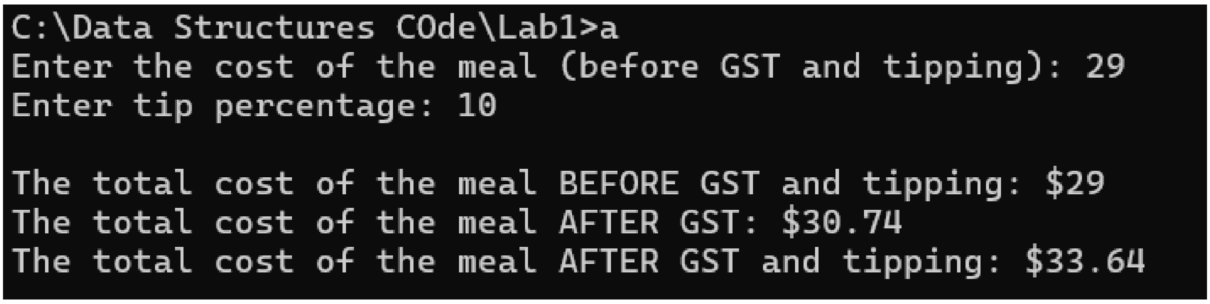
```
C:\Data Structures COde\Lab1>a
Rectangle A:
Width = 5
Height = 3

Rectangle B:
Width = 7
Height = 4

Area in rectangle B is bigger than rectangle A.
```

Q8:



```cpp
#include <iostream>
using namespace std;

int main() {
    double mealCost, tipPercentage, gst = 0.06;

    cout << "Enter the cost of the meal (before GST and tipping): ";
    cin >> mealCost;
    cout << "Enter tip percentage: ";
    cin >> tipPercentage;

    double costAfterGST = mealCost + (mealCost * gst);
    double totalCost = costAfterGST + (mealCost * (tipPercentage / 100));

    cout << "\nThe total cost of the meal BEFORE GST and tipping: $" << mealCost << endl;
    cout << "The total cost of the meal AFTER GST: $" << costAfterGST << endl;
    cout << "The total cost of the meal AFTER GST and tipping: $" << totalCost << endl;

    return 0;
}

```

Output:



```
C:\Data Structures COde\Lab1>a
Enter the cost of the meal (before GST and tipping): 29
Enter tip percentage: 10

The total cost of the meal BEFORE GST and tipping: $29
The total cost of the meal AFTER GST: $30.74
The total cost of the meal AFTER GST and tipping: $33.64
```