ASSIGNMENT 1

PROGRAMMING TECHNIQUE 1 (SECJ1013)

SEM 1 (2021/2022)

INSTRUCTIONS TO THE STUDENTS

- *This assignment must be done* ***in pairs*** *(a group consisting of 2 members).*
- *Please refer to the group list to find out your group members/ partner and your set of assignments.*
- *The application examples given in the figure in the question set can be used as a guide to design your solution (flow chart).*
- *Any form of plagiarisms is* ***NOT ALLOWED****. Students who copied other students' assignments will get* ***ZERO*** *marks (both parties, students who copied, and students that share their work).*
- *Please insert your* ***name and partner's name, matric number, and date*** *as a comment in your program.*

SUBMISSION PROCEDURE

- *Please submit this assignment no later than* ***October 29, 2024 (Tuesday, on / before 00:00 MYT)****.*
- *Only one submission per pair (group) that includes one file is required for the submission which is the flow chart (the file with the extension .pdf).*
- *Submit the assignment via the UTM's e-learning system.*

SET 1

Based on the problem given below, analyze the problem and design its solution using a **flow chart**. The flow chart must be drawn by using any appropriate drawing tools such as Microsoft Visio, draw.io (https://app.diagrams.net/), and Lucid chart (https://www.lucidchart.com/pages/examples/flowchart-maker).

You need to design a program that repeatedly accepts temperatures from the user in Celsius, converts them to Fahrenheit, and then classifies them into one of the following categories:

a) Cold if the Fahrenheit temperature is below 50°F,

b) Moderate if it's between 50°F and 77°F, and

c) Hot if it's above 77°F.

The program will continue until the user enters a sentinel value (-999) to stop. Alongside the main function, you must define and call a user-defined function to handle the temperature conversion from Celsius to Fahrenheit.

Please take note that in your solution (flow chart), you **MUST** apply:

a) Branching/ selection (if..else)

b) Loop/ repetition (repeat..until/ do..while)

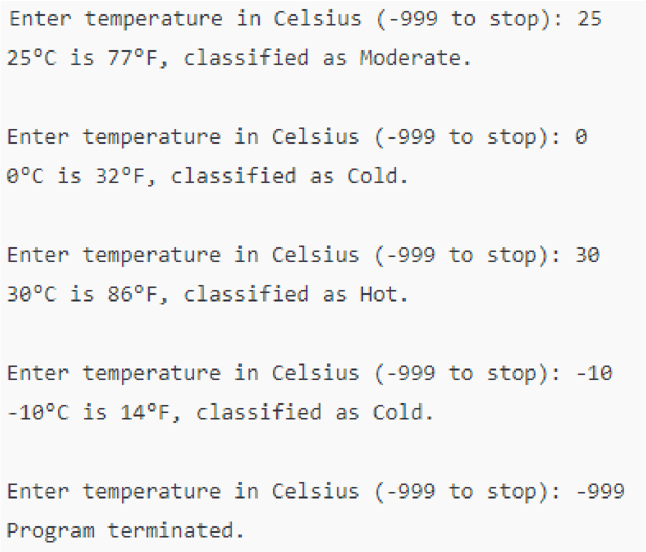
c) User-defined function flow chart. Besides the **main** function flow chart, your solution needs to design at least **ONE** more other function flow chart. Use appropriate arguments for the function.

Sample Input & Output

Sample 1
Input: 25°C
Conversion:
  Fahrenheit = (9/5 * 25) + 32 = 77°F
Classification: Moderate

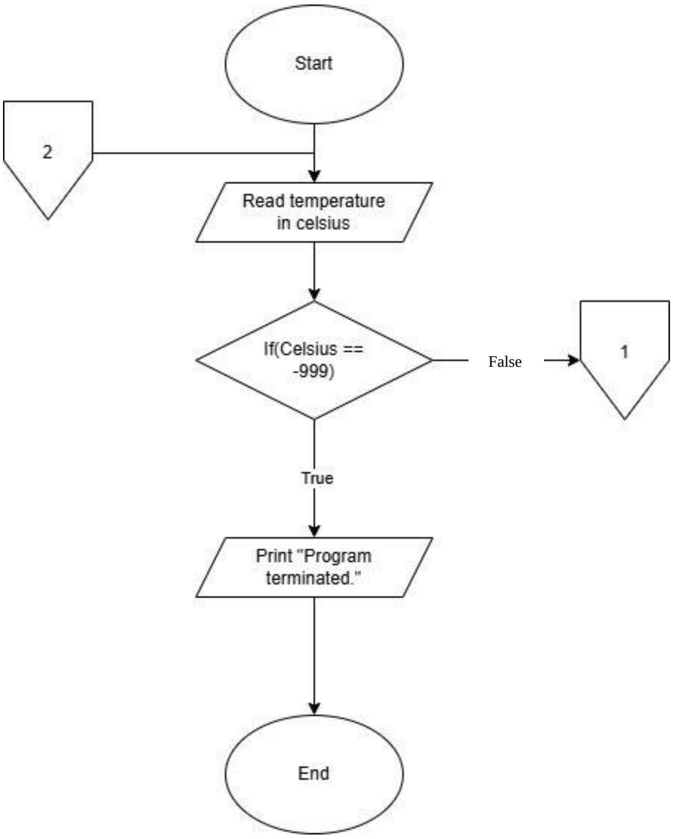
Sample 2
Input: -5°C
Conversion:
  Fahrenheit = (9/5 * -5) + 32 = 23°F
Classification: Cold

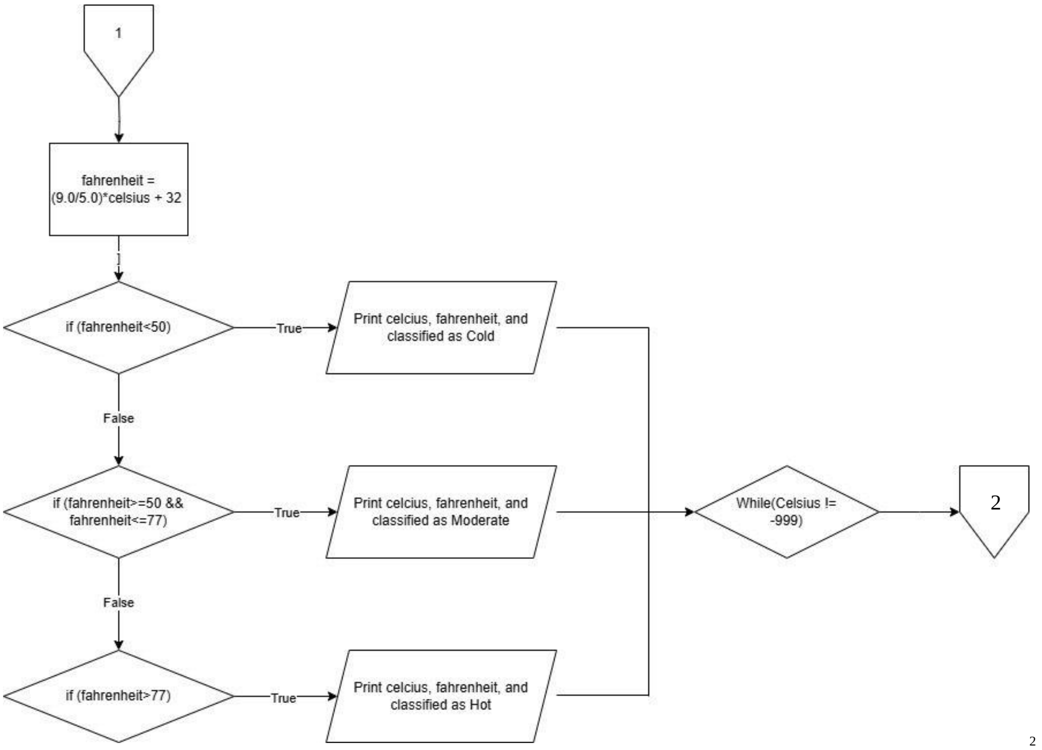
Display of input & output will be like below:



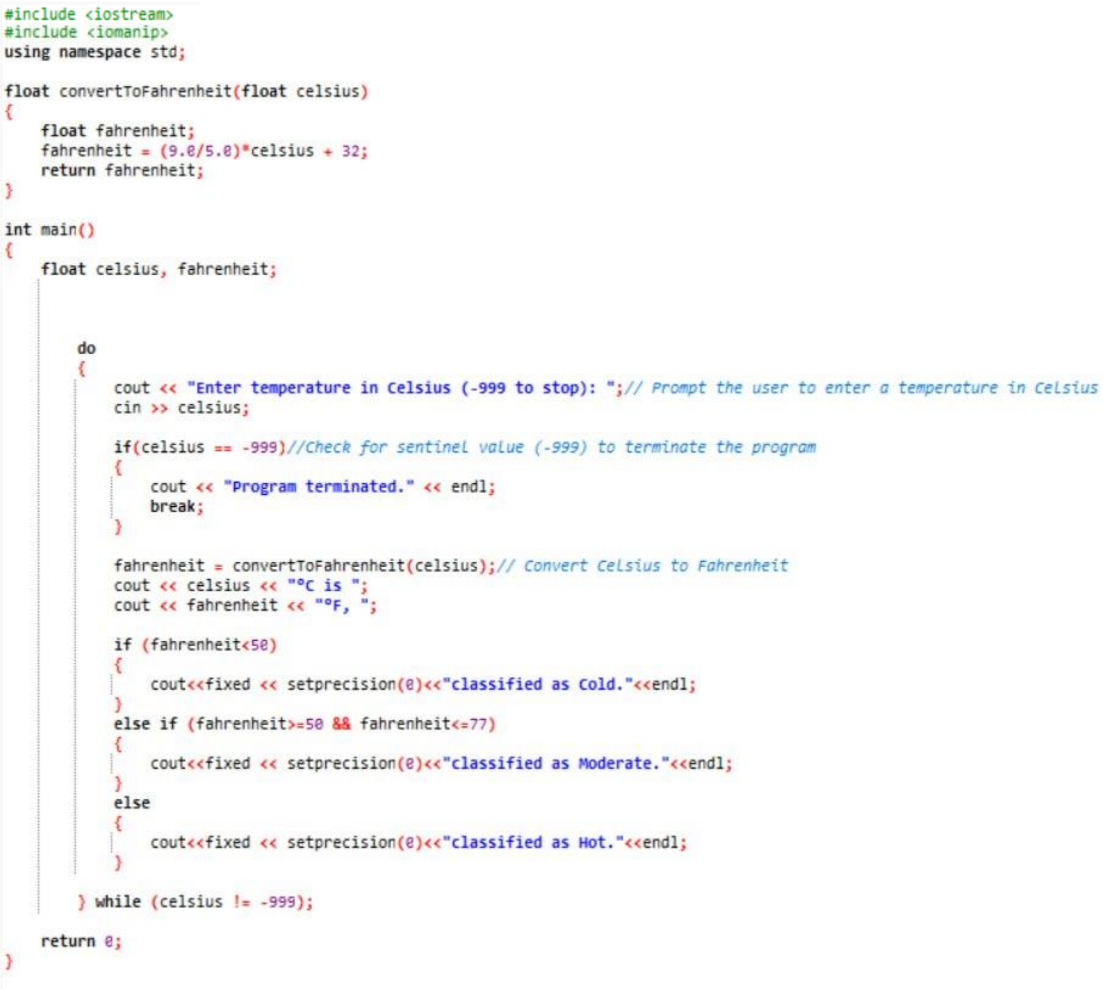
```
Enter temperature in Celsius (-999 to stop): 25
25°C is 77°F, classified as Moderate.

Enter temperature in Celsius (-999 to stop): 0
0°C is 32°F, classified as Cold.

Enter temperature in Celsius (-999 to stop): 30
30°C is 86°F, classified as Hot.

Enter temperature in Celsius (-999 to stop): -10
-10°C is 14°F, classified as Cold.

Enter temperature in Celsius (-999 to stop): -999
Program terminated.
```

Flowchart:



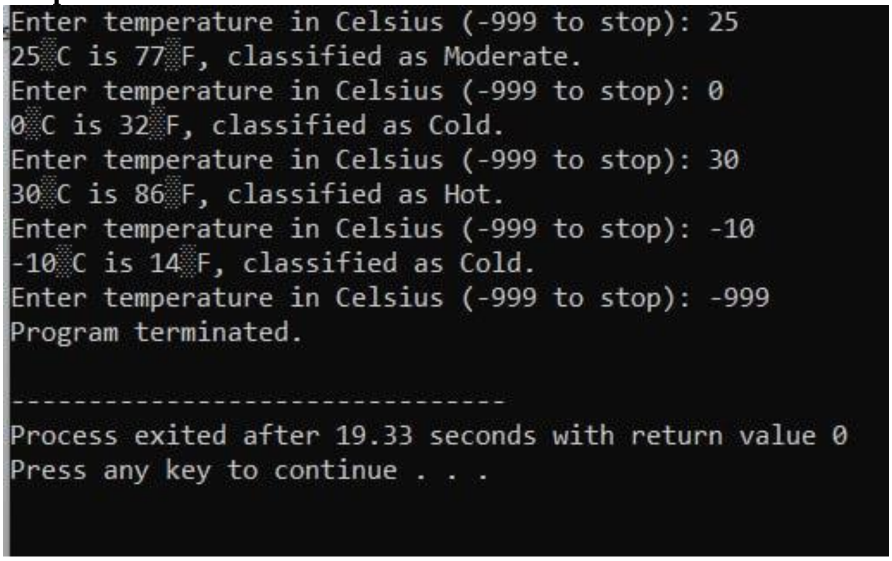

1
fahrenheit =
(9.0/5.0)*celsius + 32
if (fahrenheit<50)
True
Print celcius, fahrenheit, and classified as Cold
False
if (fahrenheit>=50 && fahrenheit<=77)
True
Print celcius, fahrenheit, and classified as Moderate
False
if (fahrenheit>77)
True
Print celcius, fahrenheit, and classified as Hot
While(Celsius != -999)
2
2

Coding:



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

float convertToFahrenheit(float celsius)
{
    float fahrenheit;
    fahrenheit = (9.0/5.0)*celsius + 32;
    return fahrenheit;
}

int main()
{
    float celsius, fahrenheit;



        do
        {
            cout << "Enter temperature in Celsius (-999 to stop): ";// Prompt the user to enter a temperature in Celsius
            cin >> celsius;

            if(celsius == -999)//Check for sentinel value (-999) to terminate the program
            {
                cout << "Program terminated." << endl;
                break;
            }

            fahrenheit = convertToFahrenheit(celsius);// Convert Celsius to Fahrenheit
            cout << celsius << "°C is ";
            cout << fahrenheit << "°F, ";

            if (fahrenheit<50)
            {
                cout<<fixed << setprecision(0)<<"classified as Cold."<<endl;
            }
            else if (fahrenheit>=50 && fahrenheit<=77)
            {
                cout<<fixed << setprecision(0)<<"classified as Moderate."<<endl;
            }
            else
            {
                cout<<fixed << setprecision(0)<<"classified as Hot."<<endl;
            }

        } while (celsius != -999);

    return 0;
}
```

Output:



```
Enter temperature in Celsius (-999 to stop): 25
25░C is 77░F, classified as Moderate.
Enter temperature in Celsius (-999 to stop): 0
0░C is 32░F, classified as Cold.
Enter temperature in Celsius (-999 to stop): 30
30░C is 86░F, classified as Hot.
Enter temperature in Celsius (-999 to stop): -10
-10░C is 14░F, classified as Cold.
Enter temperature in Celsius (-999 to stop): -999
Program terminated.

--------------------------------
Process exited after 19.33 seconds with return value 0
Press any key to continue . . .
```

SET 2

Based on the problem given below, analyze the problem and design its solution using a <u>**flow chart**</u>. The flow chart must be drawn by using any appropriate drawing tools such as Microsoft Visio, draw.io (https://app.diagrams.net/), and Lucid chart (https://www.lucidchart.com/pages/examples/flowchart-maker).

You need to design a program that allows users to calculate either Simple Interest or Compound Interest based on user input. The program will:

a) Ask the user whether they want to calculate Simple Interest or Compound Interest.

b) For both interest types, the user will input the Principal (P), Rate of Interest (R) (as a percentage), and Time (T) (in years).

c) If the user chooses Simple Interest, calculate it using the formula:
Simple Interest $(SI) = \frac{P \times R \times T}{100}$

d) If the user chooses Compound Interest, ask for the number of times interest is compounded per year (n) and calculate it using the formula:
Compound Interest (CI) =

$$CI = P \times \left(1 + \frac{R}{100 \times n}\right)^{n \times T} - P$$

e) The program will allow the user to perform multiple calculations and will terminate when the user enters a sentinel value (0 for interest type).

Please take note that in your solution (flow chart), you <u>**MUST**</u> apply:

a) Branching/ selection (if..else)

b) Loop/ repetition (repeat..until/ do..while)

c) User-defined function flow chart. Besides the <u>**main**</u> function flow chart, your solution needs to design at least <u>**ONE**</u> more other function flow chart. Use appropriate arguments for the function.

<u>**Sample Input & Output**</u>

Sample 1 (Simple Interest Example):
Input Principal (P): 1000
Input Rate (R): 5%
Input Time (T): 3 years
Calculation:

$$SI = \frac{1000 \times 5 \times 3}{100}$$

Result: Simple Interest = 150

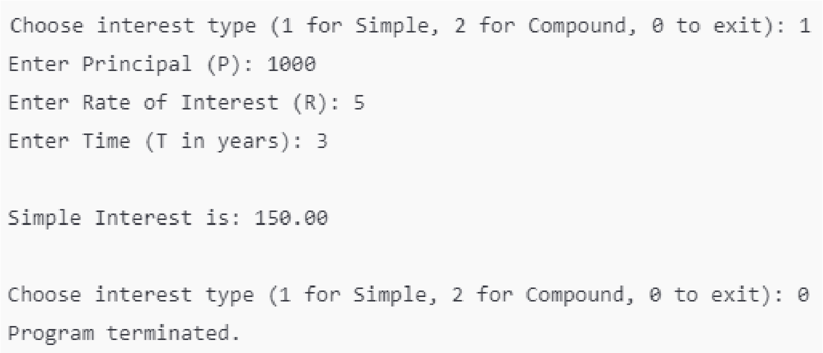
Sample 2 (Compound Interest Example):
Input Principal (P): 1000
Input Rate (R): 5%
Input Time (T): 3 years
Input Number of Times Compounded Per Year (n): 4 (quarterly)
Calculation:

$$CI = 1000 \times \left(1 + \frac{5}{100 \times 4}\right)^{4 \times 3} - 1000$$

Result: Simple Interest = 159.27

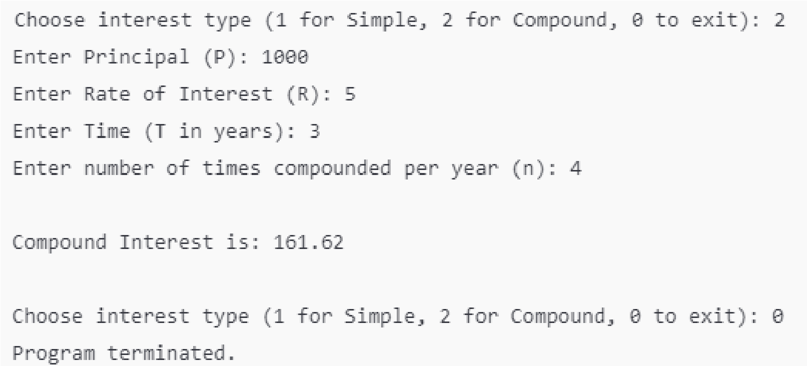
Display of input & output will be like below:

Sample Execution (Simple Interest):



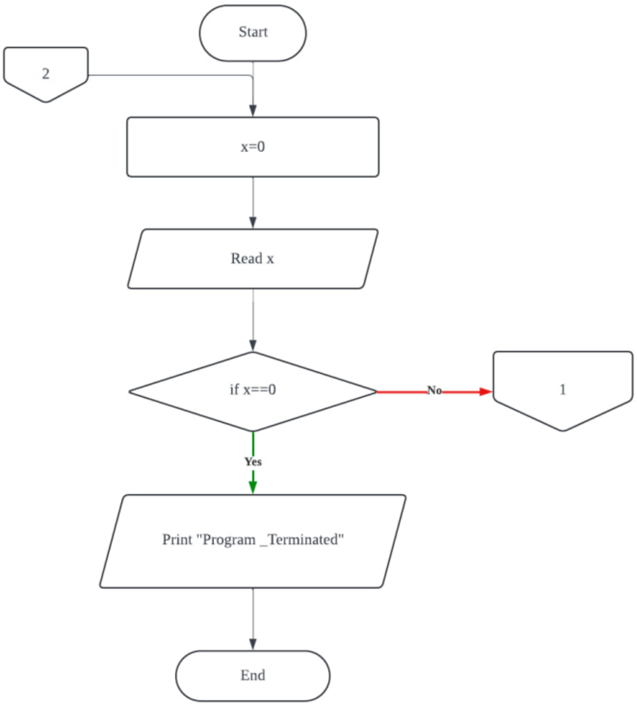
```
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 1
Enter Principal (P): 1000
Enter Rate of Interest (R): 5
Enter Time (T in years): 3

Simple Interest is: 150.00

Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 0
Program terminated.
```

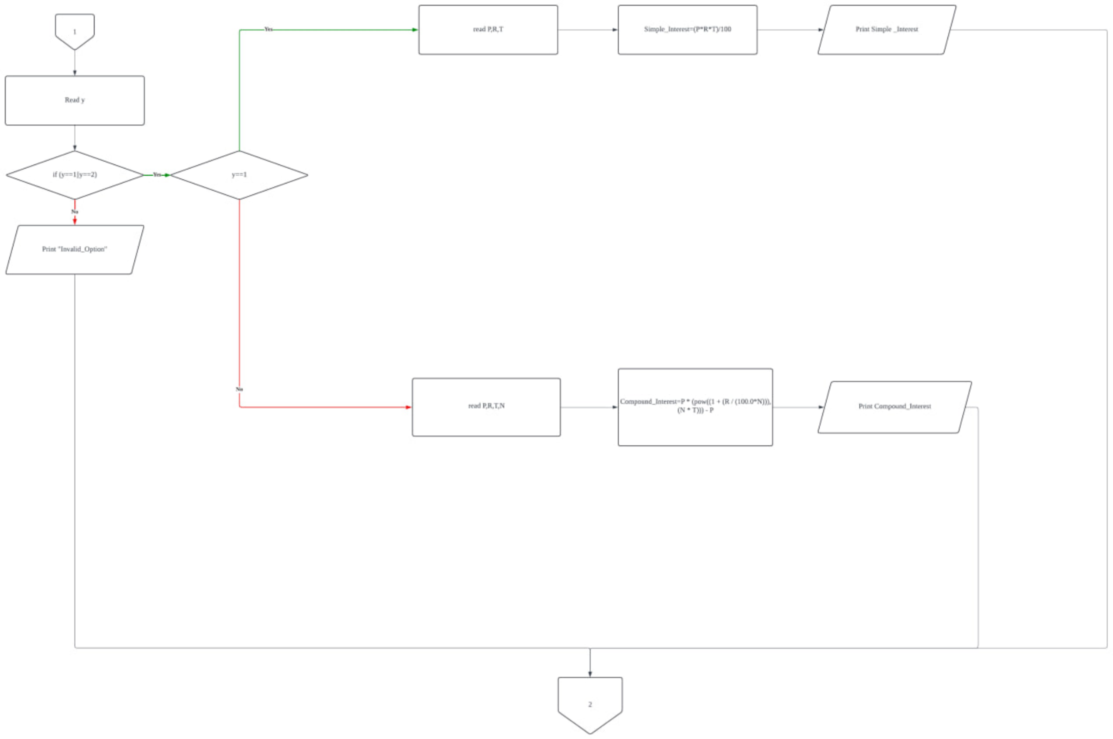
Sample Execution (Compound Interest):



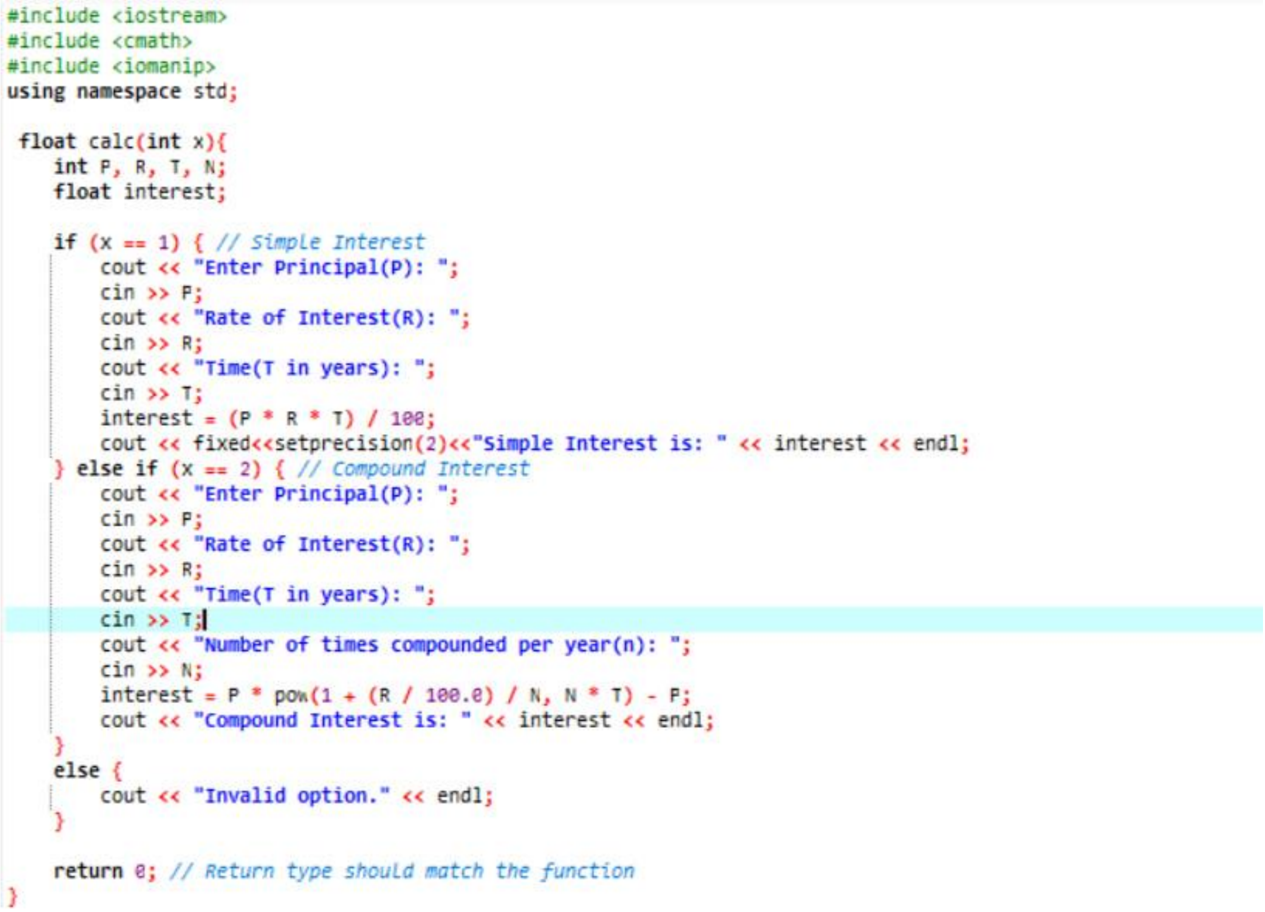
```
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 2
Enter Principal (P): 1000
Enter Rate of Interest (R): 5
Enter Time (T in years): 3
Enter number of times compounded per year (n): 4

Compound Interest is: 161.62

Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 0
Program terminated.
```

Flowchart:

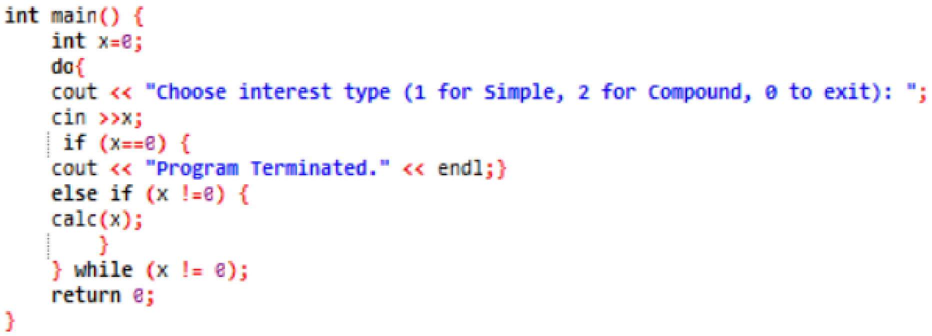




Coding:

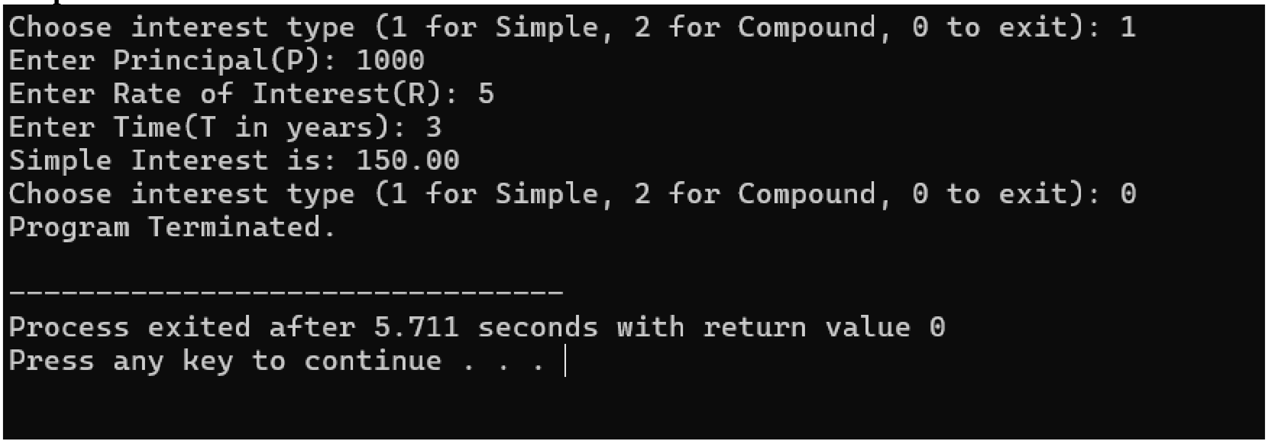


```
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

 float calc(int x){
    int P, R, T, N;
    float interest;

    if (x == 1) { // Simple Interest
        cout << "Enter Principal(P): ";
        cin >> P;
        cout << "Rate of Interest(R): ";
        cin >> R;
        cout << "Time(T in years): ";
        cin >> T;
        interest = (P * R * T) / 100;
        cout << fixed<<setprecision(2)<<"Simple Interest is: " << interest << endl;
    } else if (x == 2) { // Compound Interest
        cout << "Enter Principal(P): ";
        cin >> P;
        cout << "Rate of Interest(R): ";
        cin >> R;
        cout << "Time(T in years): ";
        cin >> T;
        cout << "Number of times compounded per year(n): ";
        cin >> N;
        interest = P * pow(1 + (R / 100.0) / N, N * T) - P;
        cout << "Compound Interest is: " << interest << endl;
    }
    else {
        cout << "Invalid option." << endl;
    }

    return 0; // Return type should match the function
}
```

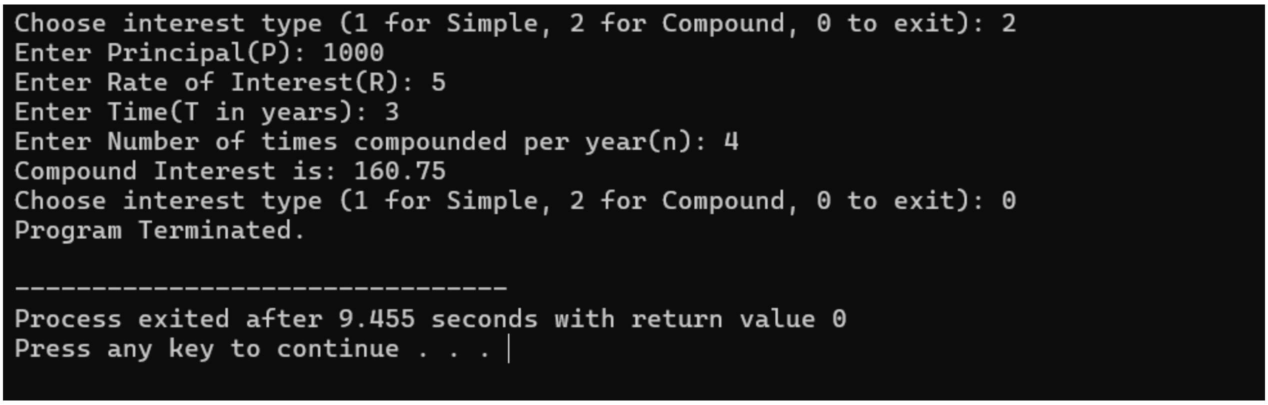


```
int main() {
    int x=0;
    do{
    cout << "Choose interest type (1 for Simple, 2 for Compound, 0 to exit): ";
    cin >>x;
     if (x==0) {
    cout << "Program Terminated." << endl;}
    else if (x !=0) {
    calc(x);
         }
    } while (x != 0);
    return 0;
}
```

Output:



```
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 1
Enter Principal(P): 1000
Enter Rate of Interest(R): 5
Enter Time(T in years): 3
Simple Interest is: 150.00
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 0
Program Terminated.

--------------------------------
Process exited after 5.711 seconds with return value 0
Press any key to continue . . .
```



```
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 2
Enter Principal(P): 1000
Enter Rate of Interest(R): 5
Enter Time(T in years): 3
Enter Number of times compounded per year(n): 4
Compound Interest is: 160.75
Choose interest type (1 for Simple, 2 for Compound, 0 to exit): 0
Program Terminated.

--------------------------------
Process exited after 9.455 seconds with return value 0
Press any key to continue . . .
```