ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់ GIC

កិច្ចការស្រាវជ្រាវក្រុម(I2-GIC-1A)

មុខវិជ្ជា ៖ SDP Ⅱ

និស្សិតឈ្មោះ	លេខសំគាល់ខ្លួននិស្សិត	ពិន្ទុ
1. SEAN Manutnithya	e20240477	……….

បង្រៀនមេរៀនដោយ ៖ លោកគ្រូ BOU Channa

បង្រៀនអនុវត្តលំហាត់ដោយ ៖អ្នកគ្រូ SEAK Leng

ឆ្នាំសិក្សា ២០២៥~២០២៦

1. Write programs using C++ to solve each problem below: 1. Write a C++ program to ask information from a student such as name, gender, major, age then display the greeting message on screen. Say he/she is eligible to vote the election if the age is at least 18. Hi, Mr. name! your age is age year old and you learn major! : for male (M) Hi, Mrs. name!, you are age year old and you are majoring in major : for female (F) For either Male or Female: You can vote. If age is at least 18. You can not vote Otherwise.

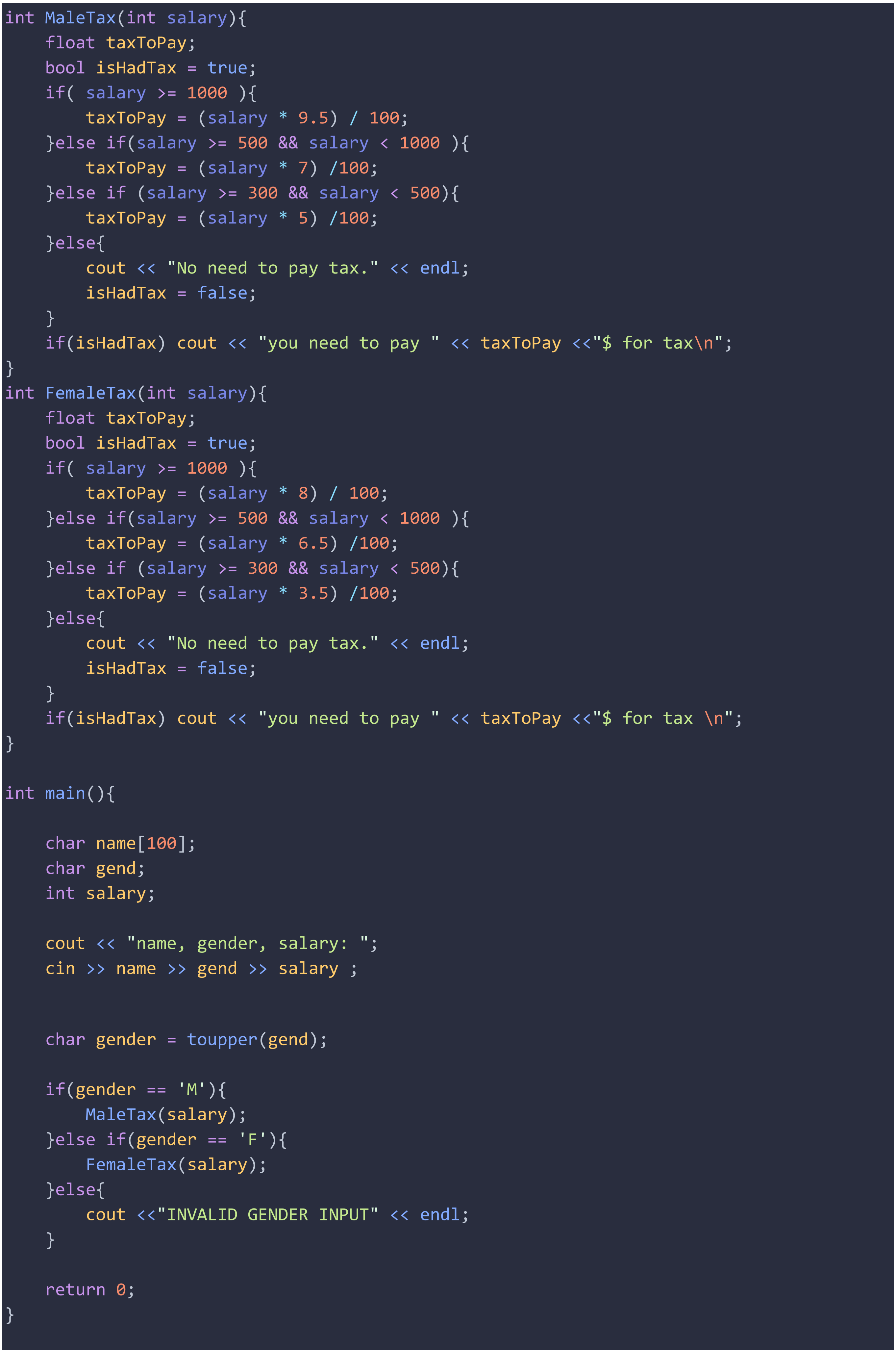
```
#include <iostream>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
using namespace std;
int main(){

    char name[100];
    char gender;
    char major[100];
    int age;
    const int valid_age = 18;
    bool valid_gender = false;


    cout << "Enter your name, gender, major, age: ";
    cin >>  name >> gender >> major >> age;

    char GENDER = toupper(gender);


    if(GENDER == 'M'){
        cout << "Hi, Mr."<<name << "! your age is " << age  << " years old and you
learn " << major <<  "!" << endl;
        valid_gender = true;
    }else if(GENDER == 'F'){
        cout << "Hi, Mrs."<<name << "! your age is " << age  << " years old and you
learn " << major <<  "!" << endl;
        valid_gender = true;
    }else{
        cout << "INVALID GENDER INPUT" << endl;
        valid_gender = false;
    }

    if(valid_gender){
    if(age >= valid_age) {
        cout << "you can vote" << endl;
    }else{
        cout << "you can not vote" << endl;
    }}

    return 0;
}
```

2. Write a C++ program to check whether an input character is a vowel or a consonant. Hint: Use ASCII code to test condition. ▪ Uppercase letters from: 65 to 90, lowercase letters from 97 to 122 ▪ Vowel and its ASCII code: a = 97, e = 101, i = 105, o = 111, u = 117

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

using namespace std;
int main(){

    char ch;
    cout << "Enter charater: ";
    cin >> ch;
    int ascii = (int)ch;

    const int isLowercaseCh = (ascii >= 65 || ascii <= 90) ;
    const int isUppercaseCh = (ascii >= 97 || ascii <= 122);
    // const int setIsVowel = ascii = {97 || 101 || 105 || 111 || 117};
    const int isLowercaseVowel = (ascii == 97 || ascii ==  101 || ascii== 105 ||
ascii ==  111 || ascii == 117 );
    const int isUppercaseVowel = (ascii == 65 || ascii ==  69 || ascii== 73 ||
ascii ==  79 || ascii == 85 );
    if(isLowercaseCh || isUppercaseCh ){
        if(isLowercaseVowel || isUppercaseVowel){
            cout << ch << " is a vowel\n";
        }else{
            cout << ch << " is a consonant\n";
        }
    }else {
        cout << "INVALID Chataracter" << endl;
    }



    return 0;
}
```

3. A program to compute tax salary of a person. The program asks for name, gender and salary of a person and tell him/her how much tax he/she is required to pay. The tax is computed based on the rules below: ▪ For males • Salary more than 1000 USD, pay tax 9.5% • Salary 500 – 1000 USD, pay tax 7% • Salary 300 – 500 USD, pay tax 5% • Salary less than 300, no need to pay tax ▪ For females • Salary more than 1000 USD, pay tax 8% • Salary 500 – 1000 USD, pay tax 6.5% • Salary 300 – 500 USD, pay tax 3.5% • Salary less than 300, no need to pay tax

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

using namespace std;
```



```
int MaleTax(int salary){
    float taxToPay;
    bool isHadTax = true;
    if( salary >= 1000 ){
        taxToPay = (salary * 9.5) / 100;
    }else if(salary >= 500 && salary < 1000 ){
        taxToPay = (salary * 7) /100;
    }else if (salary >= 300 && salary < 500){
        taxToPay = (salary * 5) /100;
    }else{
        cout << "No need to pay tax." << endl;
        isHadTax = false;
    }
    if(isHadTax) cout << "you need to pay " << taxToPay <<"$ for tax\n";
}
int FemaleTax(int salary){
    float taxToPay;
    bool isHadTax = true;
    if( salary >= 1000 ){
        taxToPay = (salary * 8) / 100;
    }else if(salary >= 500 && salary < 1000 ){
        taxToPay = (salary * 6.5) /100;
    }else if (salary >= 300 && salary < 500){
        taxToPay = (salary * 3.5) /100;
    }else{
        cout << "No need to pay tax." << endl;
        isHadTax = false;
    }
    if(isHadTax) cout << "you need to pay " << taxToPay <<"$ for tax \n";
}

int main(){

    char name[100];
    char gend;
    int salary;

    cout << "name, gender, salary: ";
    cin >> name >> gend >> salary ;


    char gender = toupper(gend);

    if(gender == 'M'){
        MaleTax(salary);
    }else if(gender == 'F'){
        FemaleTax(salary);
    }else{
        cout <<"INVALID GENDER INPUT" << endl;
    }

    return 0;
}
```

4. Create a C++ program that can play a game Rock-Paper-Scissor. The game plays between the user against the computer.

```
#include <iostream>
#include <string.h>
#include <ctime>
#include <cstdlib>
#include <random>
#include <stdlib.h>
#include <stdio.h>
using namespace std;
int main(){

    random_device rd;
    // mersenne twister
    mt19937 gen(rd());
    uniform_real_distribution<float> dist(0.0, 1.0);
    float randomNumber = dist(gen);

    // bool scissor = randomNumber >= 0.0 && randomNumber < 1.0/3.0;
    // bool rock = randomNumber >= 1.0/3.0 && randomNumber < 2.0/3.0;
    // bool paper = randomNumber >=2.0/3.0  && randomNumber <= 1.0;

    string bot_move;
    if(randomNumber < 1.0 / 3.0){
        bot_move = "scissor";
    }else if(randomNumber < 2.0/3.0){
        bot_move = "rock";
    }else{
        bot_move = "paper";
    }

    string player_move;
    cout << "your move (rock, paper, scissor):";
    cin >> player_move;

    for(char &c : player_move)
        c = tolower(c);

    bool isTie = strcasecmp(player_move.c_str(),bot_move.c_str()) ;
    bool isBotWin = (bot_move == "rock" && player_move == "scissor") || (bot_move ==
"paper" && player_move == "rock") || (bot_move == "scissor" && player_move == "paper");
    bool isPlayerWin = (bot_move == "rock" && player_move == "paper") || (bot_move ==
"paper" && player_move == "scissor") || (bot_move == "scissor" && player_move ==
"rock");

    if( isTie == 0){
        cout << "you picked " << player_move << " computer picked " << bot_move << ".
Tie" << endl;
    }else if(isBotWin){
        cout << "you picked " << player_move << " computer picked " << bot_move << ".
Computer win" << endl;
    }else if(isPlayerWin) {
        cout << "you picked " << player_move << " computer picked " << bot_move << ". you win" << endl;
    } else{
        cout <<"INVALID MOVE" << endl;
    }

    return 0;
}
```

5. Write a C++ program to convert a minute to a time format which consists of hour, minute, and second (h:m:s). A user is required to input a minute. ▪ Hours = minutes / 60 ▪ Remainderminutes = minutes % 60 ▪ Seconds = remainderminutes * 60

```
#include <iostream>
#include <cctype>
#include <string>
#include <stdlib.h>

using namespace std;

struct Time {
    int hour;
    int minute;
    int seconde;
};

void convertTime(float minute) {
    Time t;
    t.hour = (int) minute / 60;
    t.minute =(int) minute % 60;
    t.seconde = (minute - (int) minute) * 60;

    printf("%dh : %dmn : %ds\n",t.hour, t.minute, t.seconde);
}

int main(){

    float minute;
    cout << "minute :";

    if(!(cin >> minute) || minute < 0){
        cout << "INVALID INPUT" << endl;
        return 0;
    } else{
        convertTime(minute);
    }

    // cout <<  minute << endl;
    // cout << (int)  minute << endl;
    // cout << minute - (int) minute<< endl;
    // cout << t.seconde << endl;

    return 0;
}
```

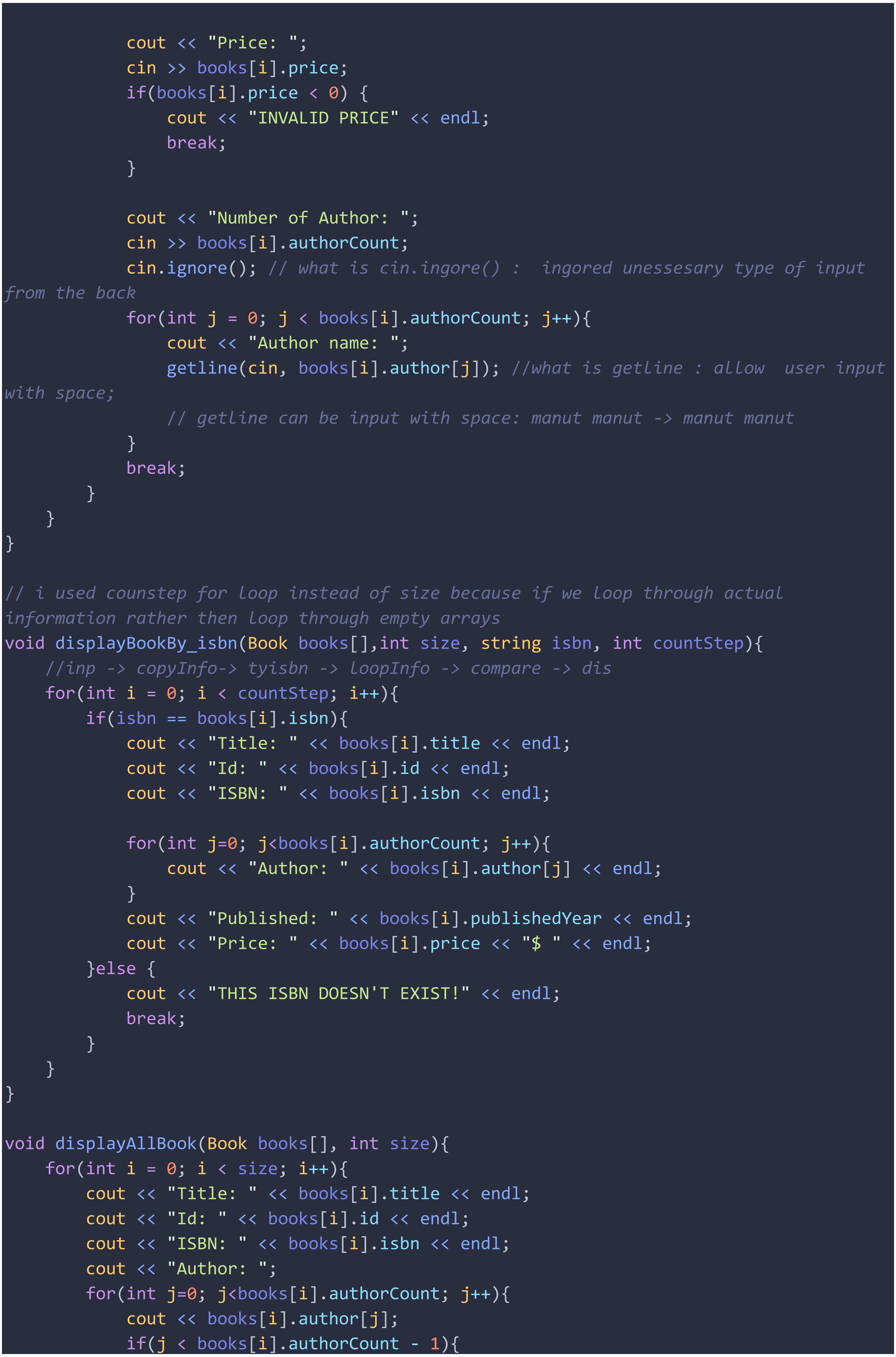
6. Write a C++ program to find the summation of numbers from 1 to n except number 10 and 30, where n is a number input by a user and n should be greater than 50.

```cpp
#include <iostream>
#include <string>
#include <stdlib.h>
#include <cctype>

using namespace std;

int Summatation(int inp_n){

    int sum = 0;

    if(inp_n < 50){
        cout << "ENTER VALUE GREATER THAN 50!" << endl;
        return -1;
    }

    for (int stp = 0; stp < inp_n; stp++){
        if(stp == 10 || stp == 30) continue;
        sum+=stp;
    }

    return sum;


}

int main(){
    int inp_n;
    cout << "ENTER: ";
    cin >> inp_n;
    int result = Summatation(inp_n);
    if(result != -1){
        cout << result << endl;
    }
    return 0;
}
```

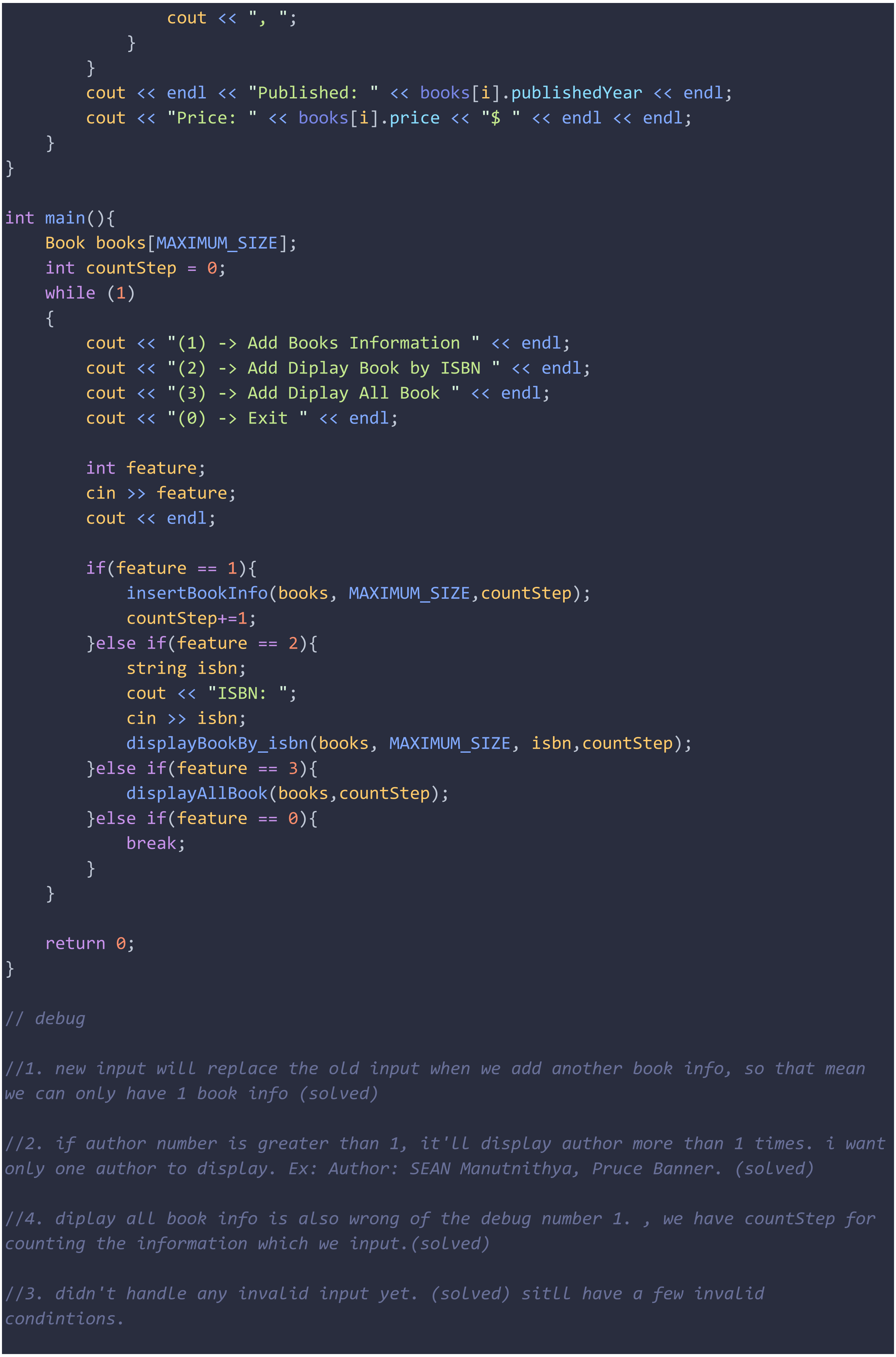
7. Write a C++ program to create a new data structure for storing info of book (book ID, book ISBN, book title, published year, author names and price ($)). Each book could have more than one author. To do: ▪ Create an array that can store 5 books' info. ▪ Create a function to display a book info based on ISBN. This function takes a parameter which is an ISBN of a book. void displayBookByISBN(Book books[], int size, string isbn){.... } ▪ Create a function to display information of all books. void displayAllBooks(Book books[], int size) { ... }

```cpp
#include <iostream>
#include <string>
#include <stdlib.h>
#include <ctime>
#define MAXIMUM_SIZE 5
#define MAX_AUTHOR 5

using namespace std;
```

```
struct Book{
    int id;
    int publishedYear;
    float price;
    string isbn;
    string title;

    int authorCount;
    string author[MAX_AUTHOR];
};

int getCurrentYear(){
    time_t current_time = time(NULL);
    struct tm local_time = *localtime(&current_time);
    int currentYear = local_time.tm_year + 1900;

    return currentYear;
}

void insertBookInfo(Book books[], int size, int countStep){
    if(countStep >= size){
        cout << "THE STORAGE IS FULL! "<< endl;
    }else{
        for(int i = countStep; i < size; i++){
            // 1st book.id -> idx 0, 2nd book.id -> idx 1, check previouse idx use
coutStep
            cout << "Book ID: ";
            cin >> books[i].id;
            bool isExisted = false;
            int existedId = books[i].id;
            for(int c = 0; c < countStep; c++){
                if(existedId == books[c].id){
                    cout << "THIS ID IS REALDY EXISTED!" << endl;
                    isExisted = true;
                    break;
                }
            }
            if(isExisted)break;

            cout << "Book ISBN: ";
            cin >> books[i].isbn;

            cin.ignore();
            cout << "Book Title: ";
            // cin >> books[i].title;
            getline(cin, books[i].title);

            cout << "Published year: ";
            cin >> books[i].publishedYear;
            int py = books[i].publishedYear;
            if(py < 0 || py > 2026) {
                cout << "INVALID YEAR!" << endl;
                break;
            };
```



```
            cout << "Price: ";
            cin >> books[i].price;
            if(books[i].price < 0) {
                cout << "INVALID PRICE" << endl;
                break;
            }

            cout << "Number of Author: ";
            cin >> books[i].authorCount;
            cin.ignore(); // what is cin.ingore() :  ingored unessesary type of input from the back
            for(int j = 0; j < books[i].authorCount; j++){
                cout << "Author name: ";
                getline(cin, books[i].author[j]); //what is getline : allow  user input with space;
                // getline can be input with space: manut manut -> manut manut
            }
            break;
        }
    }
}

// i used counstep for loop instead of size because if we loop through actual information rather then loop through empty arrays
void displayBookBy_isbn(Book books[],int size, string isbn, int countStep){
    //inp -> copyInfo-> tyisbn -> loopInfo -> compare -> dis
    for(int i = 0; i < countStep; i++){
        if(isbn == books[i].isbn){
            cout << "Title: " << books[i].title << endl;
            cout << "Id: " << books[i].id << endl;
            cout << "ISBN: " << books[i].isbn << endl;

            for(int j=0; j<books[i].authorCount; j++){
                cout << "Author: " << books[i].author[j] << endl;
            }
            cout << "Published: " << books[i].publishedYear << endl;
            cout << "Price: " << books[i].price << "$ " << endl;
        }else {
            cout << "THIS ISBN DOESN'T EXIST!" << endl;
            break;
        }
    }
}

void displayAllBook(Book books[], int size){
    for(int i = 0; i < size; i++){
        cout << "Title: " << books[i].title << endl;
        cout << "Id: " << books[i].id << endl;
        cout << "ISBN: " << books[i].isbn << endl;
        cout << "Author: ";
        for(int j=0; j<books[i].authorCount; j++){
            cout << books[i].author[j];
            if(j < books[i].authorCount - 1){
```



```
                cout << ", ";
            }
        }
        cout << endl << "Published: " << books[i].publishedYear << endl;
        cout << "Price: " << books[i].price << "$ " << endl << endl;
    }
}

int main(){
    Book books[MAXIMUM_SIZE];
    int countStep = 0;
    while (1)
    {
        cout << "(1) -> Add Books Information " << endl;
        cout << "(2) -> Add Diplay Book by ISBN " << endl;
        cout << "(3) -> Add Diplay All Book " << endl;
        cout << "(0) -> Exit " << endl;

        int feature;
        cin >> feature;
        cout << endl;

        if(feature == 1){
            insertBookInfo(books, MAXIMUM_SIZE,countStep);
            countStep+=1;
        }else if(feature == 2){
            string isbn;
            cout << "ISBN: ";
            cin >> isbn;
            displayBookBy_isbn(books, MAXIMUM_SIZE, isbn,countStep);
        }else if(feature == 3){
            displayAllBook(books,countStep);
        }else if(feature == 0){
            break;
        }
    }

    return 0;
}

// debug

//1. new input will replace the old input when we add another book info, so that mean we can only have 1 book info (solved)

//2. if author number is greater than 1, it'll display author more than 1 times. i want only one author to display. Ex: Author: SEAN Manutnithya, Pruce Banner. (solved)

//4. diplay all book info is also wrong of the debug number 1. , we have countStep for counting the information which we input.(solved)

//3. didn't handle any invalid input yet. (solved) sitll have a few invalid condintions.
```

```
//optional task
// after input invalid information instead of breaking and stopping a whole operatoin, it should re-ask for invalid input
```

8. Create 5 functions to: ▪ i) convert temperature Celsius to Fahrenheit: Fahrenheit = (Celsius * 9.0 / 5.0) + 32 ▪ ii) convert temperature Fahrenheit to Celsius: Celsius = (Fahrenheit – 32) * 50 / 9.0 ▪ iii) find root of quadratic equation ax2+bx+c=0. ▪ iv) compute BMI of a person's weight and height and tell whether he/she is overweight, underweight and other terms according to BMI list: - bmi = weight / (height * height) - - - - bmi < 18.5, underweight bmi < 25, normal weight bmi < 30, overweight else, obese ▪ v) sum numbers from 1 to n except those numbers that are divisible by 3, n is a parameter of the function. Design a menu program to demonstrate these 5 functions with different test cases during the run time of the program.

```cpp
#include <iostream>
#include <string>
#include <stdlib.h>
#include <math.h>
#include <iomanip>
using namespace std;

void ConvertCelsiusToFahrenhite(){
    cout << endl << "Convert celsius to fahrenhite." << endl;
    int cel;
    cin >> cel;
    float fah = (cel * 9.0 / 5.0) + 32;
    cout << fah << " F" << endl;
}

void ConvertFahrenhiteToCelsius(){
    cout << endl << "Convert fahrenhite to celsius." << endl;
    int fah;
    cin >> fah;
    float cel = (fah - 32)*50.0 / 9.0;
    cout  << cel << " C" << endl;
}

void FindRootOfQuadraticEqs(){
    cout << "Find root of quadratic equation." << endl;
    int a,b,c;
    cout << "Enter a, b, c: ";
    cin >> a >> b >> c;
    int delta = pow(b,2) -( 4 * a * c);
    float root_1,root_2;

    if(delta > 0){
        cout << "Delta > 0" << endl;
        root_1 = ((-pow(b,2) + sqrt(delta))) /( 2*a);
        root_2 = ((-pow(b,2) - sqrt(delta))) /( 2*a);
        cout << "x1 = " << setprecision(2) << root_1 << endl << "x2 = " << root_2 << endl;
    }else if(delta == 0){
        cout << "Delta = 0" << endl;
        root_1 = -(pow(b,2)) / ( 2*a);
        root_2 = root_1;
        cout << setprecision(2) <<"x1 = x2 = " << root_1  << endl;
    }else {
        delta = -1 * delta;
        float imaPart = (sqrt(delta)) / (2 * a);
        float realPart = -(pow(b,2)) / ( 2 * a);

        cout << "delta < 0" << endl;
        cout << "x1 = " << setprecision(2) << realPart <<" + "<< imaPart << "i" <<endl;
        cout << "x2 = " << setprecision(2) << realPart <<"-"<< imaPart << "i" <<endl;
    }
}

void CalcuateBMI(){
    cout << "Compute BMI." << endl;
    float h;
    float bmi;
    bool isValid = true;
    cout << "Height in meter: ";
    cin >> h;
    if(h <=0){
        cout << "INVALID...!" << endl;
        isValid = false;
    }
    float w;
    cout << "Weight in Kg: ";
    cin >> w;
    if(w <= 0){
        cout << "INVALID...!" << endl;
        isValid = false;
    }
    if(isValid){
        bmi = w / ( h * h );
        if(bmi < 18.5){
            cout << "Underweight" << endl;
        }else if(bmi < 25){
            cout << "Normalweight" << endl;
        }else if(bmi < 30){
            cout << "Overweight" << endl;
        }else{
            cout << "obese" << endl;
        }
    }

}

void Summation(int n){
    cout << "Sum numbers from 1 to n, Execpt number division by 3." << endl;
    int result = 0;
    for(int l = 1; l <= n; l++){
        if(l % 3 == 0) continue;
        result += l;
        }
        cout << result << endl;
```

```
}

int main(){
    int feature;
    while (true)
    {
        cout << endl << "(1) Convert celsius to fahrenhite." << endl;
        cout << "(2) Convert fahrenhite to celsius." << endl;
        cout << "(3) Find root of quadratic equation." << endl;
        cout << "(4) Compute BMI." << endl;
        cout << "(5) Sum numbers from 1 to n, Execpt number division by 3." << endl;
        cout << "(0) To Exit!" << endl ;
        cout << "Enter your choice:: " ;
        cin >> feature;
        if(feature == 0){
            break;
        }else if(feature == 1){
            ConvertCelsiusToFahrenhite();
        }else if(feature == 2){
            ConvertFahrenhiteToCelsius();
        }else if(feature == 3){
            FindRootOfQuadraticEqs();
        }else if(feature == 4){
            CalcuateBMI();
        }else if(feature == 5){
            int n;
            cout << "n: ";
            cin >> n;
            if(n < 1){
                cout << "INVALID...!" << endl;
            }else{
                Summation(n);
            }
        }else{
            cout << "INVALID CHOICE...!" << endl;
        }
    }
    return 0;
}
```