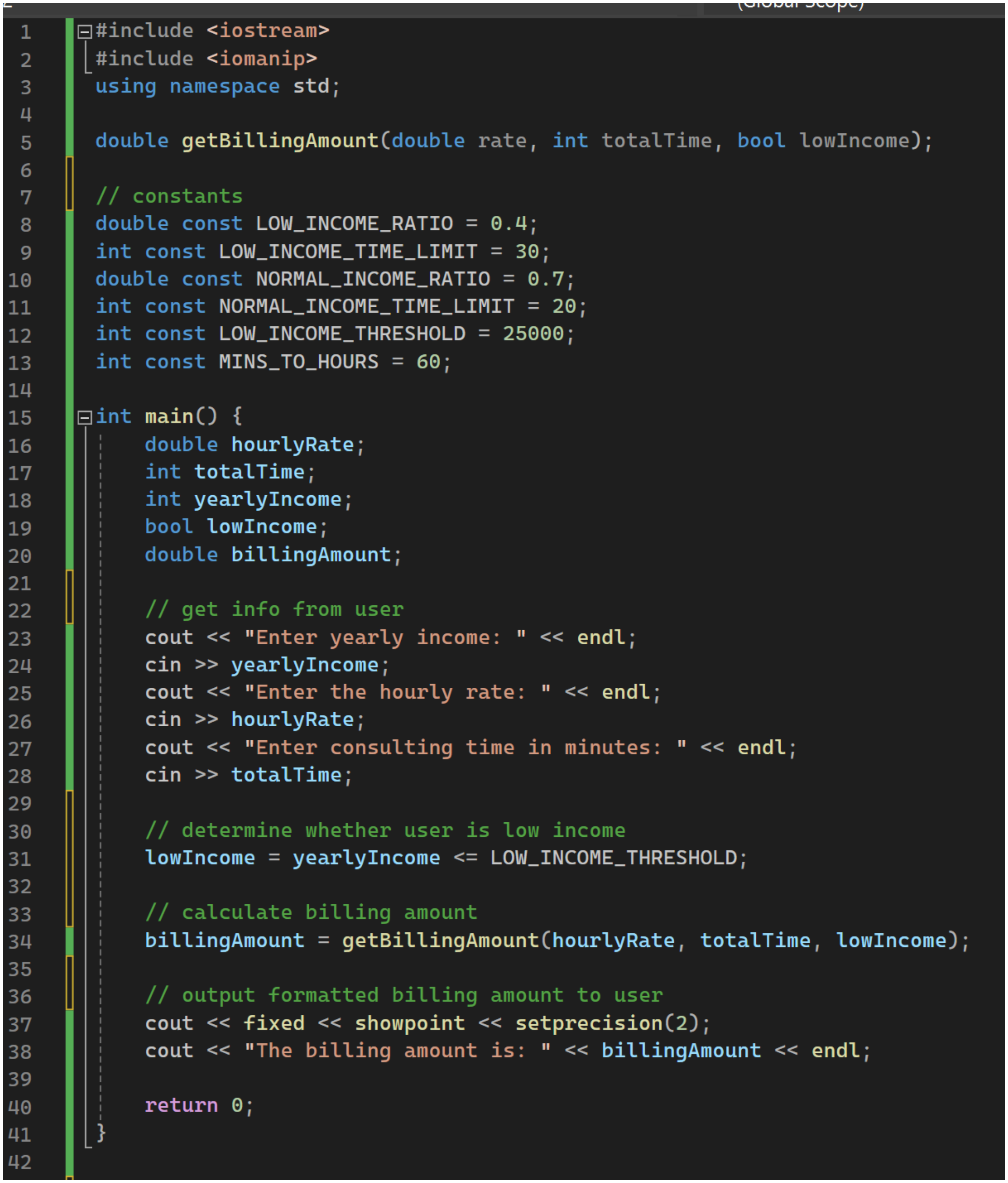
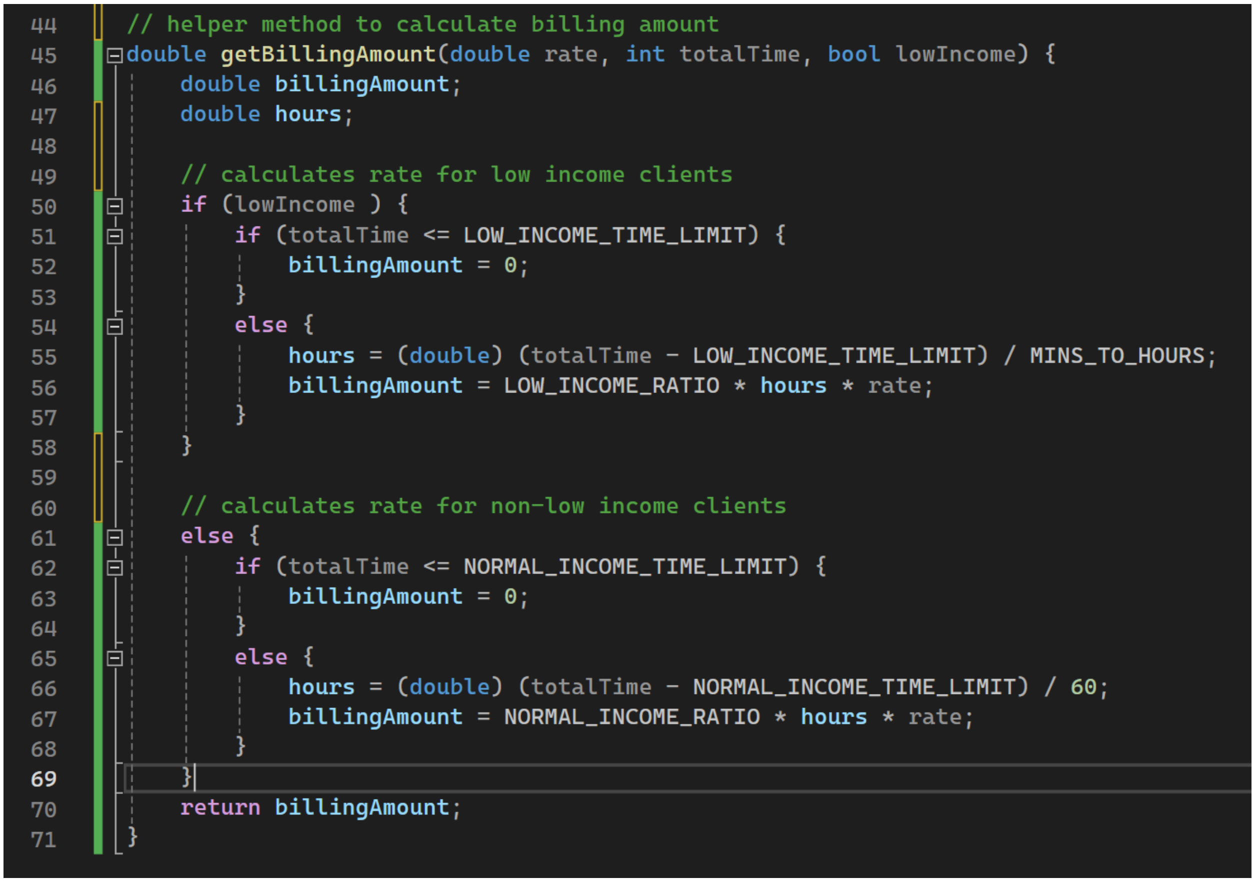
Ex 1
Code:

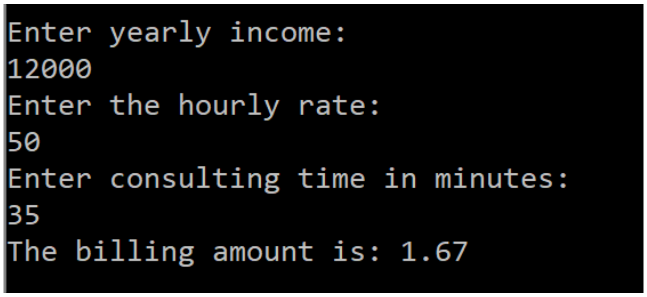


```cpp
#include <iostream>
#include <iomanip>
using namespace std;

double getBillingAmount(double rate, int totalTime, bool lowIncome);

// constants
double const LOW_INCOME_RATIO = 0.4;
int const LOW_INCOME_TIME_LIMIT = 30;
double const NORMAL_INCOME_RATIO = 0.7;
int const NORMAL_INCOME_TIME_LIMIT = 20;
int const LOW_INCOME_THRESHOLD = 25000;
int const MINS_TO_HOURS = 60;

int main() {
    double hourlyRate;
    int totalTime;
    int yearlyIncome;
    bool lowIncome;
    double billingAmount;

    // get info from user
    cout << "Enter yearly income: " << endl;
    cin >> yearlyIncome;
    cout << "Enter the hourly rate: " << endl;
    cin >> hourlyRate;
    cout << "Enter consulting time in minutes: " << endl;
    cin >> totalTime;

    // determine whether user is low income
    lowIncome = yearlyIncome <= LOW_INCOME_THRESHOLD;

    // calculate billing amount
    billingAmount = getBillingAmount(hourlyRate, totalTime, lowIncome);

    // output formatted billing amount to user
    cout << fixed << showpoint << setprecision(2);
    cout << "The billing amount is: " << billingAmount << endl;

    return 0;
}

```

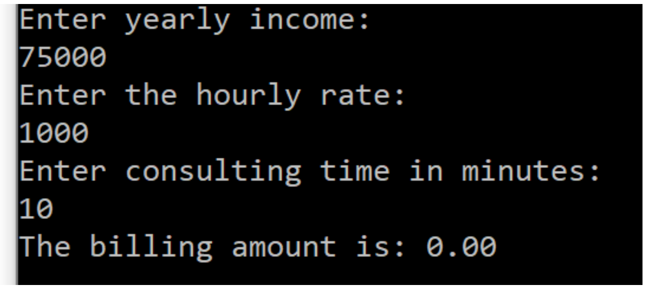


```
// helper method to calculate billing amount
double getBillingAmount(double rate, int totalTime, bool lowIncome) {
    double billingAmount;
    double hours;

    // calculates rate for low income clients
    if (lowIncome ) {
        if (totalTime <= LOW_INCOME_TIME_LIMIT) {
            billingAmount = 0;
        }
        else {
            hours = (double) (totalTime - LOW_INCOME_TIME_LIMIT) / MINS_TO_HOURS;
            billingAmount = LOW_INCOME_RATIO * hours * rate;
        }
    }

    // calculates rate for non-low income clients
    else {
        if (totalTime <= NORMAL_INCOME_TIME_LIMIT) {
            billingAmount = 0;
        }
        else {
            hours = (double) (totalTime - NORMAL_INCOME_TIME_LIMIT) / 60;
            billingAmount = NORMAL_INCOME_RATIO * hours * rate;
        }
    }
    return billingAmount;
}
```

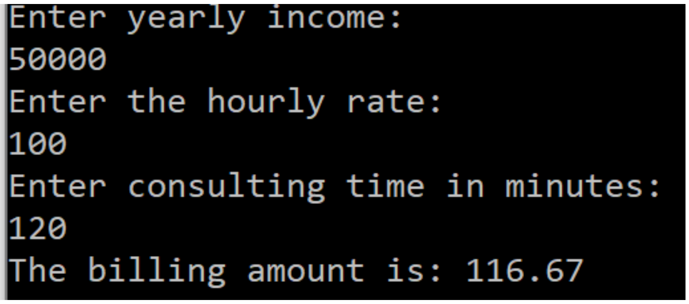
Output:



```
Enter yearly income:
12000
Enter the hourly rate:
50
Enter consulting time in minutes:
35
The billing amount is: 1.67
```



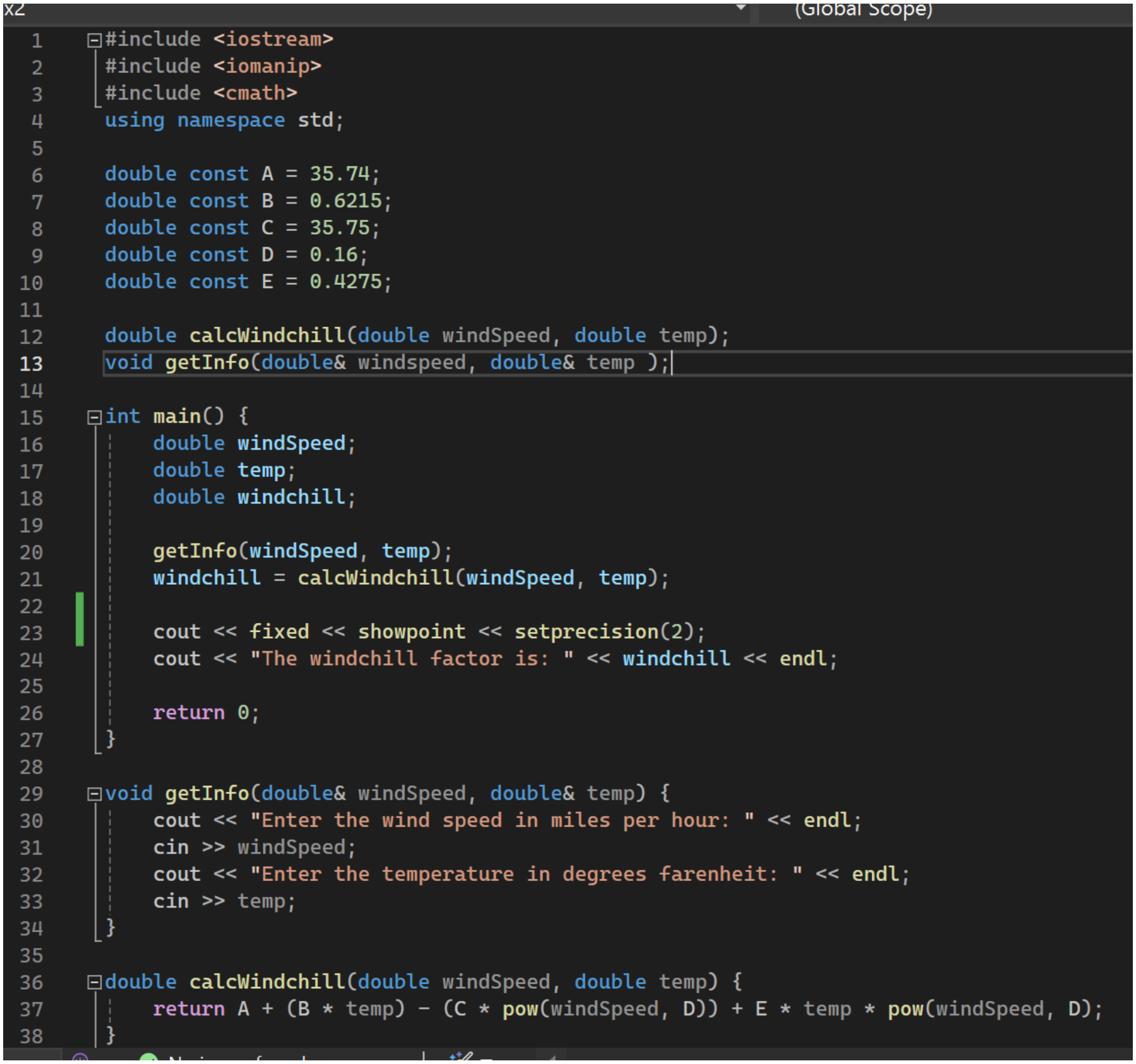
```
Enter yearly income:
75000
Enter the hourly rate:
1000
Enter consulting time in minutes:
10
The billing amount is: 0.00
```



```
Enter yearly income:
50000
Enter the hourly rate:
100
Enter consulting time in minutes:
120
The billing amount is: 116.67
```

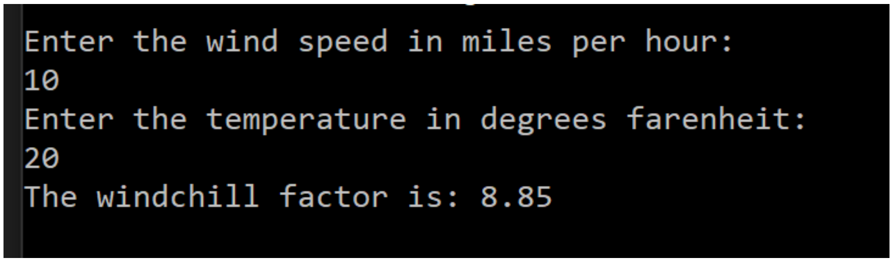
Ex 2

Code:



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

double const A = 35.74;
double const B = 0.6215;
double const C = 35.75;
double const D = 0.16;
double const E = 0.4275;

double calcWindchill(double windSpeed, double temp);
void getInfo(double& windspeed, double& temp );

int main() {
    double windSpeed;
    double temp;
    double windchill;

    getInfo(windSpeed, temp);
    windchill = calcWindchill(windSpeed, temp);

    cout << fixed << showpoint << setprecision(2);
    cout << "The windchill factor is: " << windchill << endl;

    return 0;
}

void getInfo(double& windSpeed, double& temp) {
    cout << "Enter the wind speed in miles per hour: " << endl;
    cin >> windSpeed;
    cout << "Enter the temperature in degrees farenheit: " << endl;
    cin >> temp;
}

double calcWindchill(double windSpeed, double temp) {
    return A + (B * temp) - (C * pow(windSpeed, D)) + E * temp * pow(windSpeed, D);
}
```

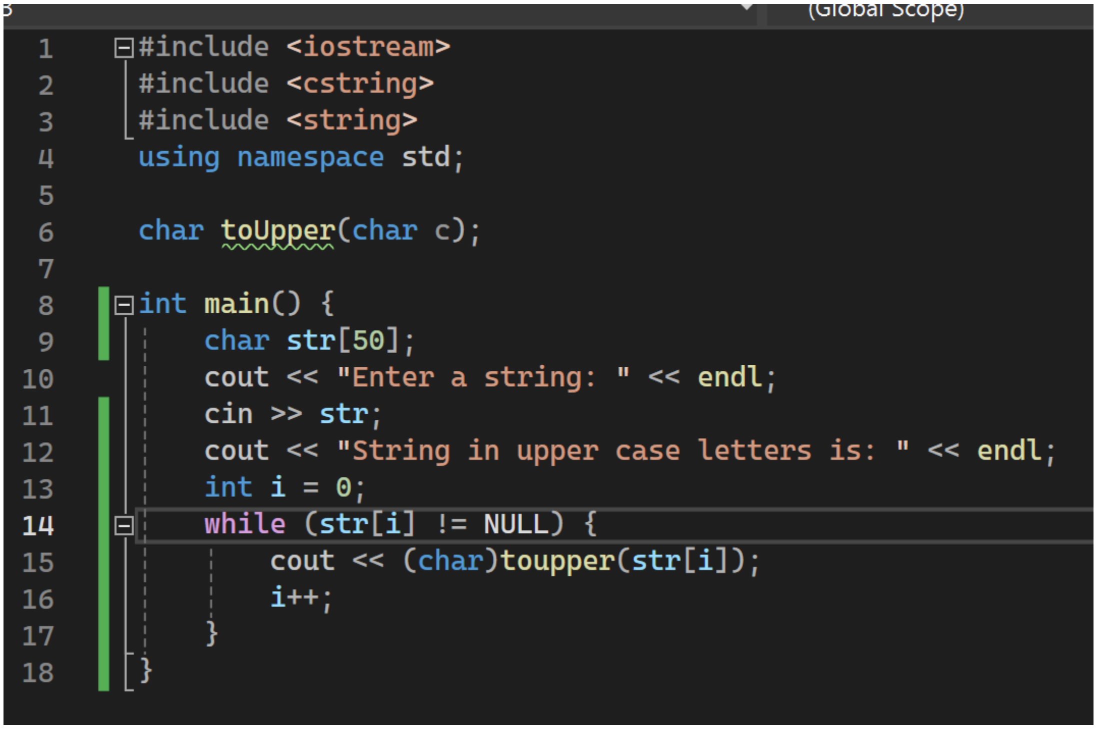
Output:



```
Enter the wind speed in miles per hour:
10
Enter the temperature in degrees farenheit:
20
The windchill factor is: 8.85
```

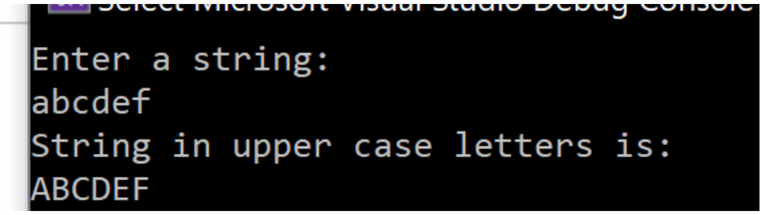
Ex 3

Code:

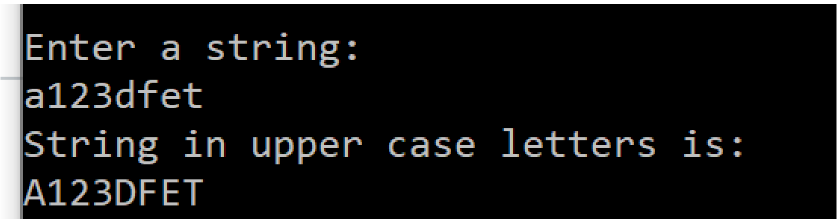


```cpp
#include <iostream>
#include <cstring>
#include <string>
using namespace std;

char toUpper(char c);

int main() {
    char str[50];
    cout << "Enter a string: " << endl;
    cin >> str;
    cout << "String in upper case letters is: " << endl;
    int i = 0;
    while (str[i] != NULL) {
        cout << (char)toupper(str[i]);
        i++;
    }
}
```

Output:



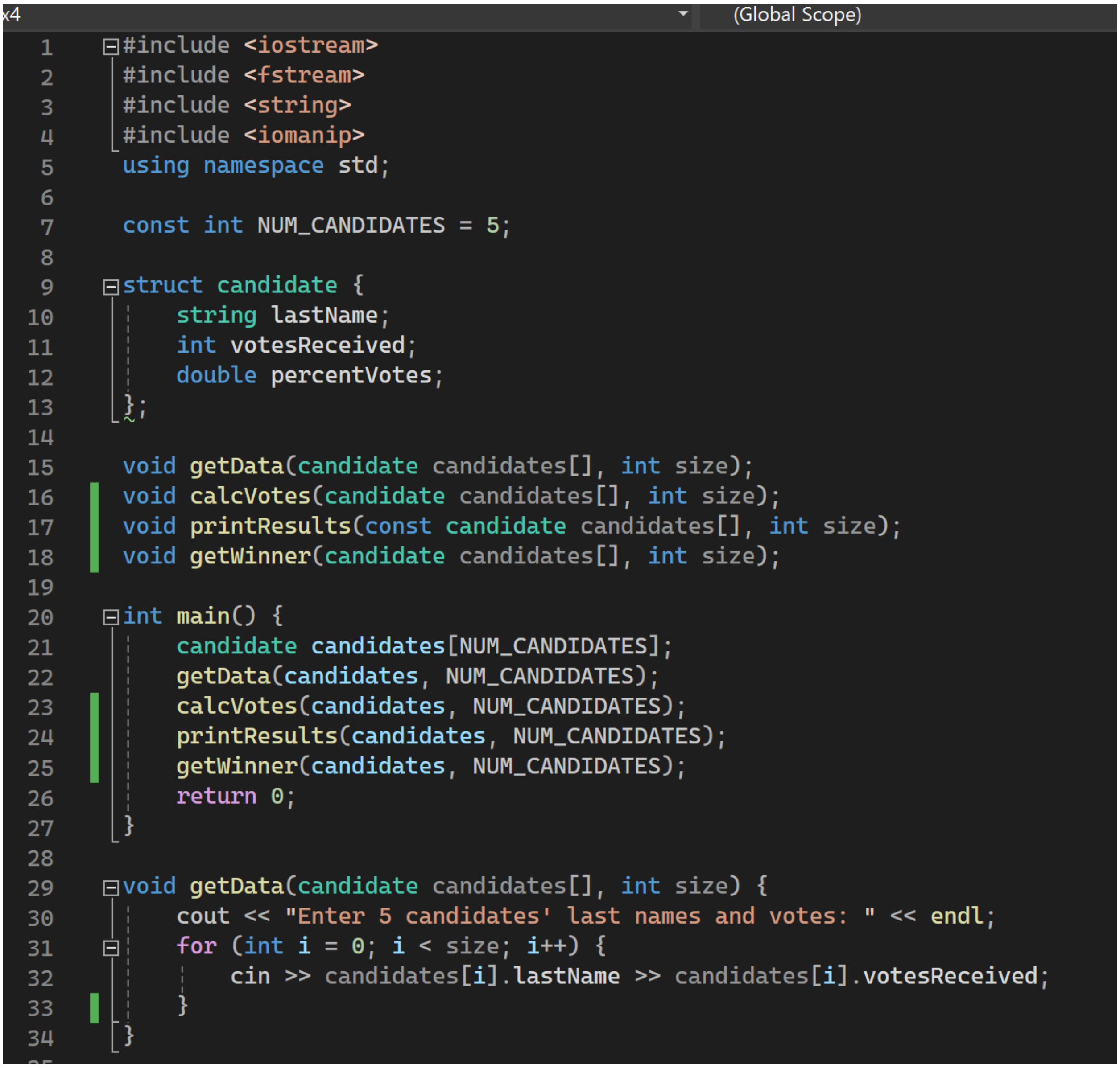
```
Enter a string:
abcdef
String in upper case letters is:
ABCDEF
```



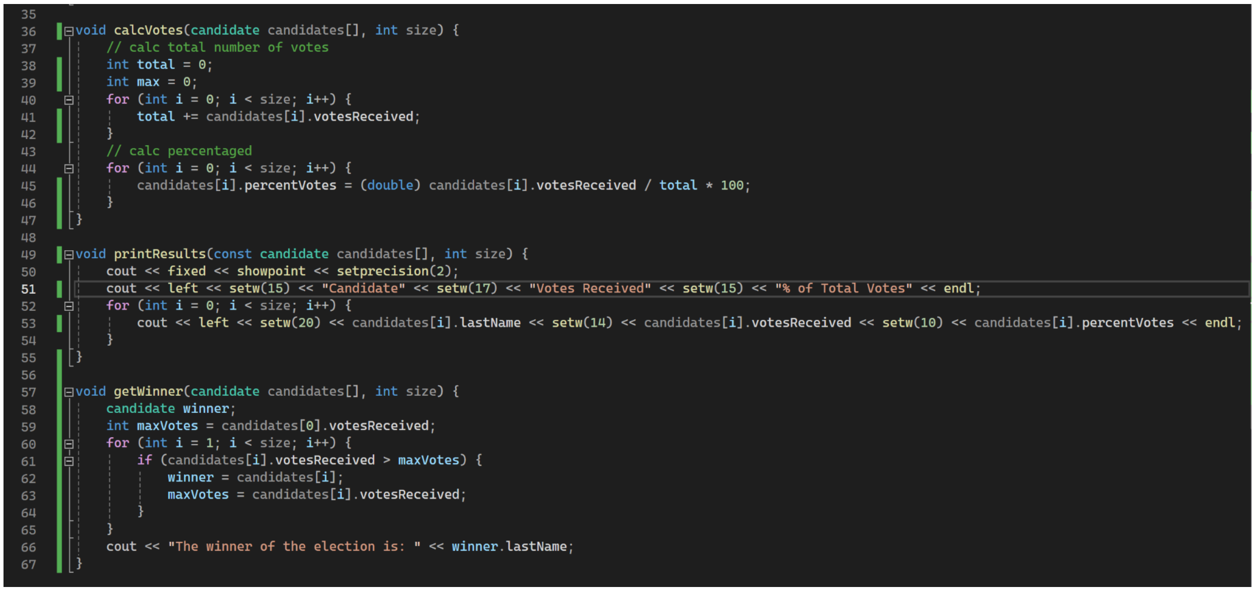
```
Enter a string:
a123dfet
String in upper case letters is:
A123DFET
```

Ex 4

Code:

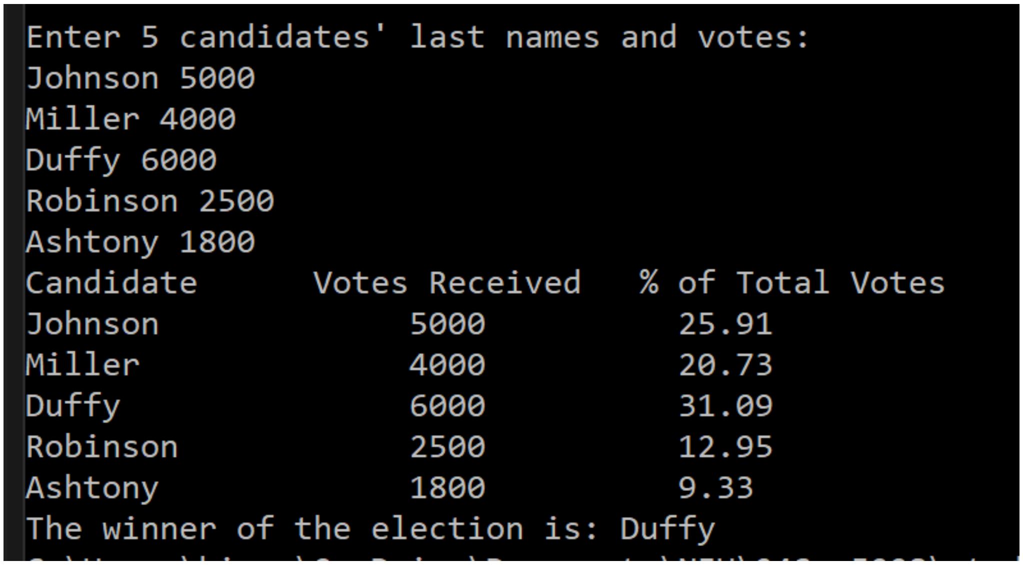


```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
using namespace std;

const int NUM_CANDIDATES = 5;

struct candidate {
    string lastName;
    int votesReceived;
    double percentVotes;
};

void getData(candidate candidates[], int size);
void calcVotes(candidate candidates[], int size);
void printResults(const candidate candidates[], int size);
void getWinner(candidate candidates[], int size);

int main() {
    candidate candidates[NUM_CANDIDATES];
    getData(candidates, NUM_CANDIDATES);
    calcVotes(candidates, NUM_CANDIDATES);
    printResults(candidates, NUM_CANDIDATES);
    getWinner(candidates, NUM_CANDIDATES);
    return 0;
}

void getData(candidate candidates[], int size) {
    cout << "Enter 5 candidates' last names and votes: " << endl;
    for (int i = 0; i < size; i++) {
        cin >> candidates[i].lastName >> candidates[i].votesReceived;
    }
}
```



```cpp

void calcVotes(candidate candidates[], int size) {
    // calc total number of votes
    int total = 0;
    int max = 0;
    for (int i = 0; i < size; i++) {
        total += candidates[i].votesReceived;
    }
    // calc percentaged
    for (int i = 0; i < size; i++) {
        candidates[i].percentVotes = (double) candidates[i].votesReceived / total * 100;
    }
}

void printResults(const candidate candidates[], int size) {
    cout << fixed << showpoint << setprecision(2);
    cout << left << setw(15) << "Candidate" << setw(17) << "Votes Received" << setw(15) << "% of Total Votes" << endl;
    for (int i = 0; i < size; i++) {
        cout << left << setw(20) << candidates[i].lastName << setw(14) << candidates[i].votesReceived << setw(10) << candidates[i].percentVotes << endl;
    }
}

void getWinner(candidate candidates[], int size) {
    candidate winner;
    int maxVotes = candidates[0].votesReceived;
    for (int i = 1; i < size; i++) {
        if (candidates[i].votesReceived > maxVotes) {
            winner = candidates[i];
            maxVotes = candidates[i].votesReceived;
        }
    }
    cout << "The winner of the election is: " << winner.lastName;
}
```

Output:



```
Enter 5 candidates' last names and votes:
Johnson 5000
Miller 4000
Duffy 6000
Robinson 2500
Ashtony 1800
Candidate      Votes Received   % of Total Votes
Johnson             5000          25.91
Miller              4000          20.73
Duffy               6000          31.09
Robinson            2500          12.95
Ashtony             1800          9.33
The winner of the election is: Duffy
```