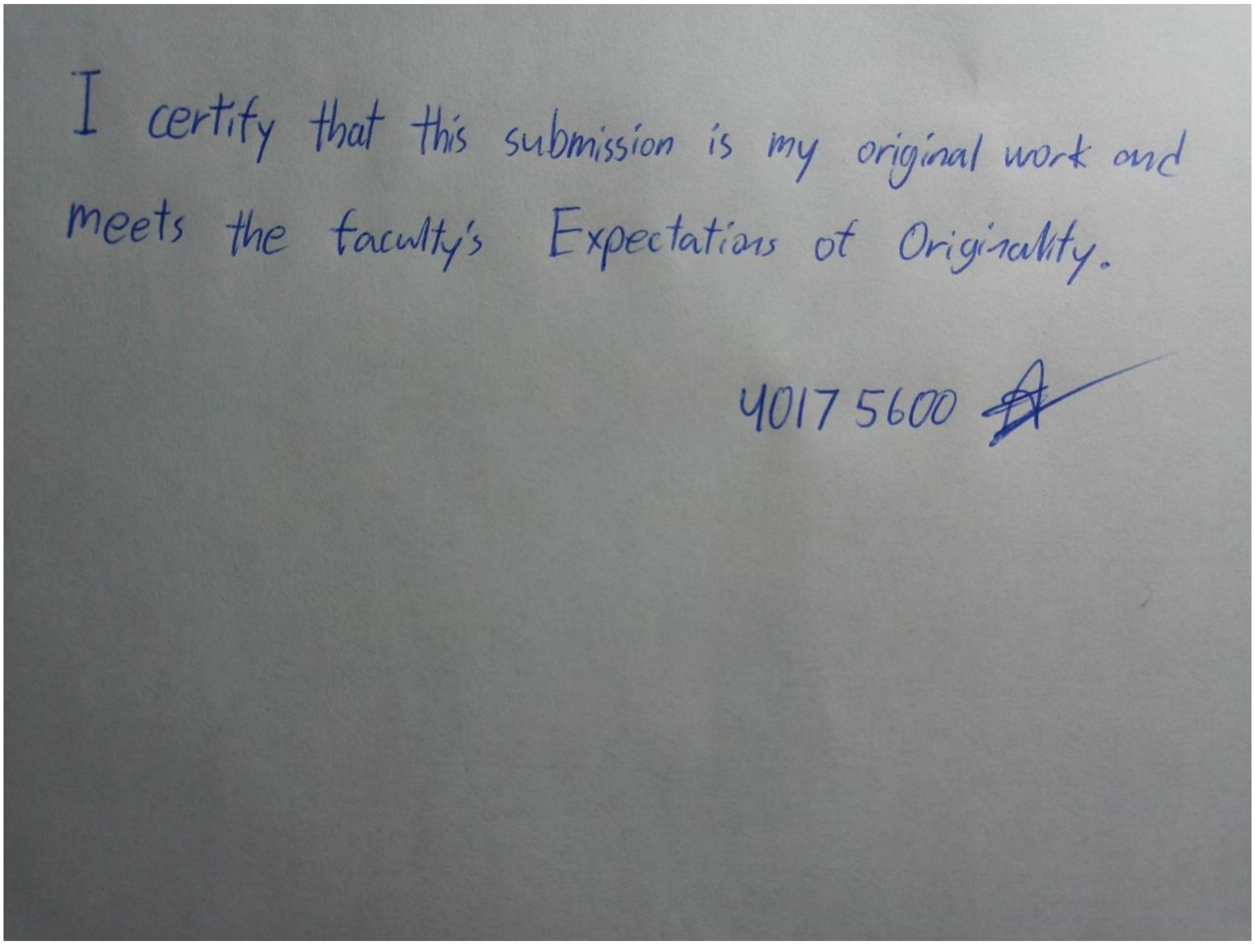
Hei Wang Andre Law

4017 5600

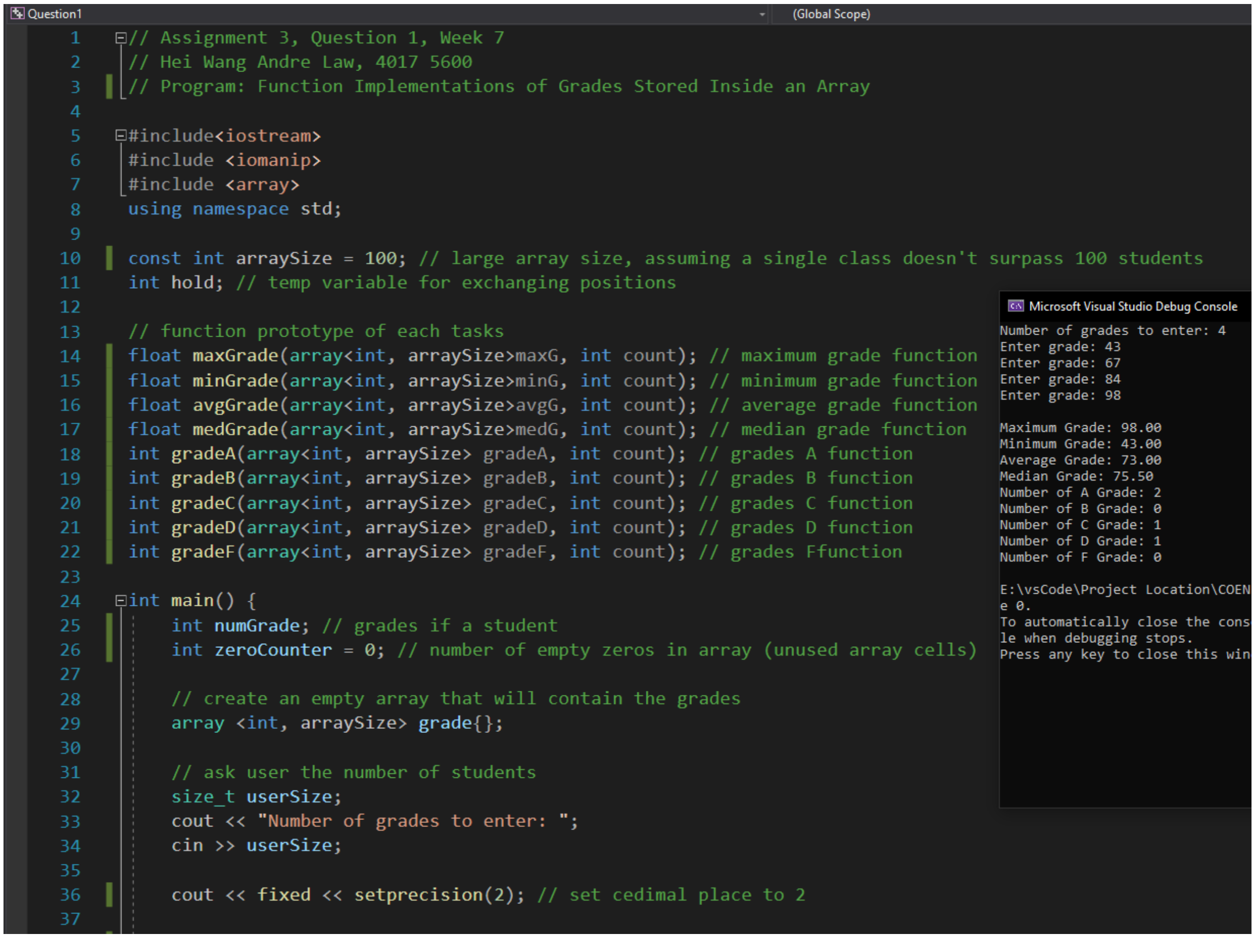
19/03/2021



I certify that this submission is my original work and meets the faculty's Expectations of Originality.

4017 5600

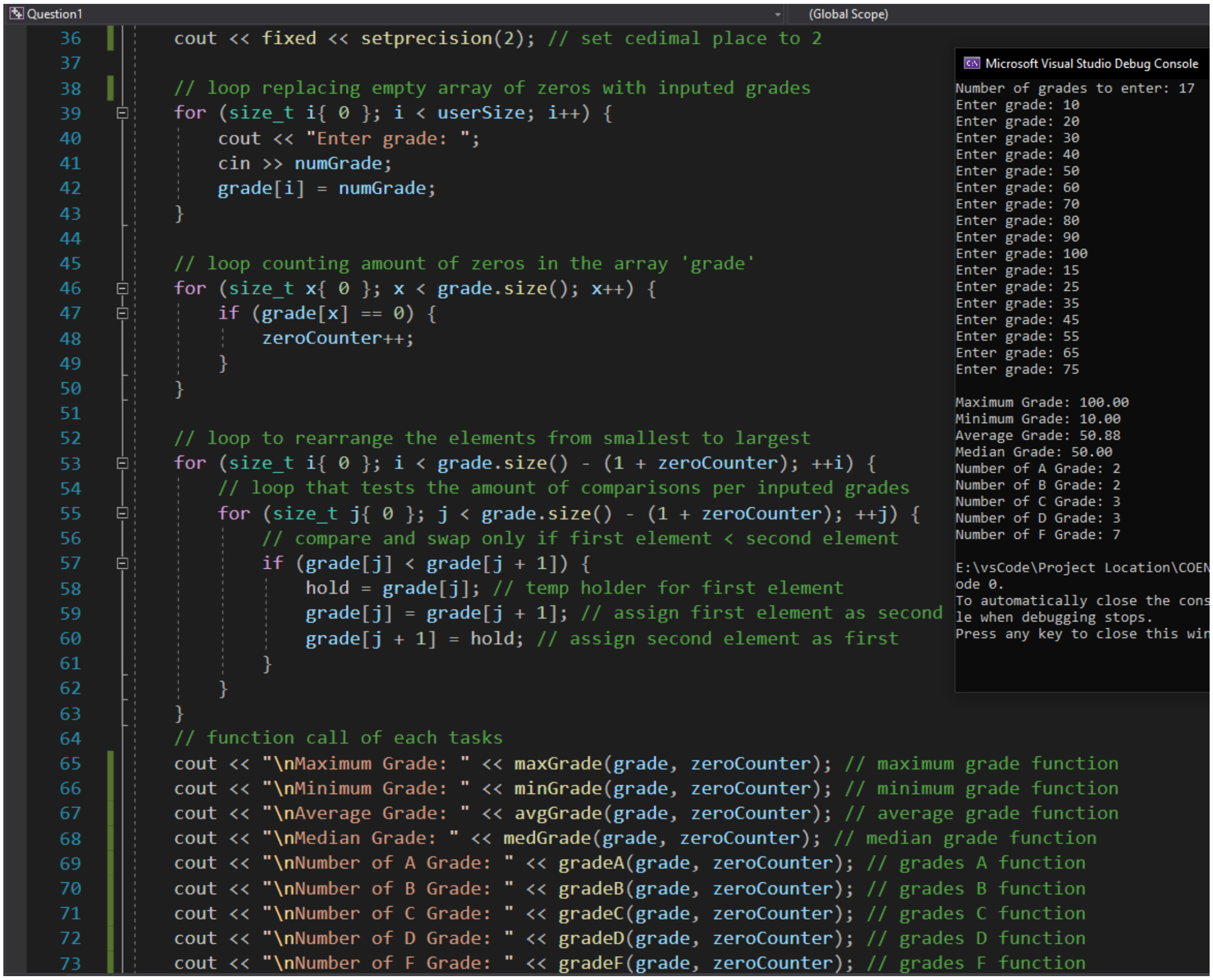
Question 1: (Part 1)



```cpp
// Assignment 3, Question 1, Week 7
// Hei Wang Andre Law, 4017 5600
// Program: Function Implementations of Grades Stored Inside an Array

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

const int arraySize = 100; // large array size, assuming a single class doesn't surpass 100 students
int hold; // temp variable for exchanging positions

// function prototype of each tasks
float maxGrade(array<int, arraySize>maxG, int count); // maximum grade function
float minGrade(array<int, arraySize>minG, int count); // minimum grade function
float avgGrade(array<int, arraySize>avgG, int count); // average grade function
float medGrade(array<int, arraySize>medG, int count); // median grade function
int gradeA(array<int, arraySize> gradeA, int count); // grades A function
int gradeB(array<int, arraySize> gradeB, int count); // grades B function
int gradeC(array<int, arraySize> gradeC, int count); // grades C function
int gradeD(array<int, arraySize> gradeD, int count); // grades D function
int gradeF(array<int, arraySize> gradeF, int count); // grades Ffunction

int main() {
    int numGrade; // grades if a student
    int zeroCounter = 0; // number of empty zeros in array (unused array cells)

    // create an empty array that will contain the grades
    array <int, arraySize> grade{};

    // ask user the number of students
    size_t userSize;
    cout << "Number of grades to enter: ";
    cin >> userSize;

    cout << fixed << setprecision(2); // set cedimal place to 2

```

```
Number of grades to enter: 4
Enter grade: 43
Enter grade: 67
Enter grade: 84
Enter grade: 98

Maximum Grade: 98.00
Minimum Grade: 43.00
Average Grade: 73.00
Median Grade: 75.50
Number of A Grade: 2
Number of B Grade: 0
Number of C Grade: 1
Number of D Grade: 1
Number of F Grade: 0

E:\vsCode\Project Location\COEN
e 0.
To automatically close the cons
le when debugging stops.
Press any key to close this win
```

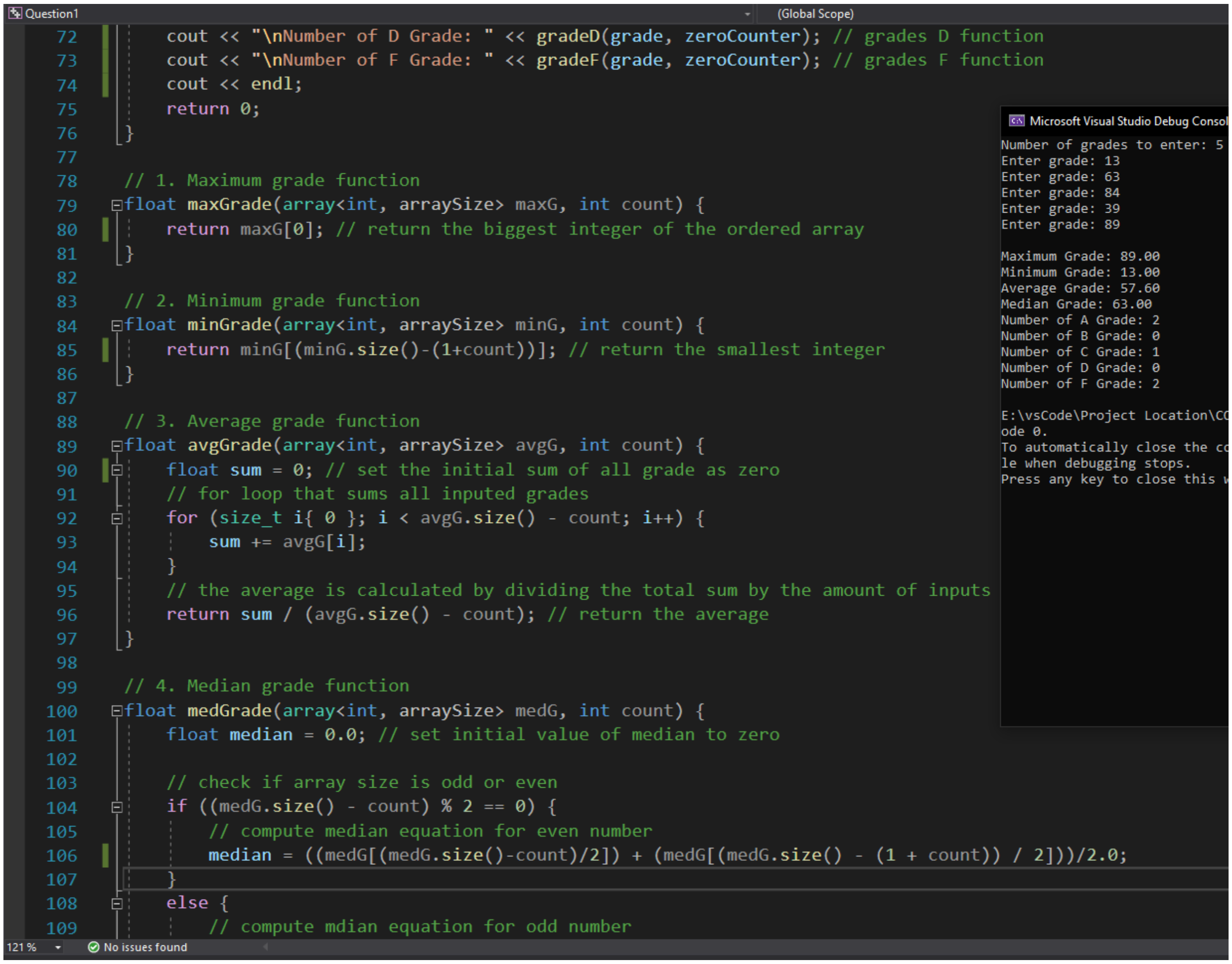
Question 1: (Part 2)



```cpp
    cout << fixed << setprecision(2); // set cedimal place to 2

    // loop replacing empty array of zeros with inputed grades
    for (size_t i{ 0 }; i < userSize; i++) {
        cout << "Enter grade: ";
        cin >> numGrade;
        grade[i] = numGrade;
    }

    // loop counting amount of zeros in the array 'grade'
    for (size_t x{ 0 }; x < grade.size(); x++) {
        if (grade[x] == 0) {
            zeroCounter++;
        }
    }

    // loop to rearrange the elements from smallest to largest
    for (size_t i{ 0 }; i < grade.size() - (1 + zeroCounter); ++i) {
        // loop that tests the amount of comparisons per inputed grades
        for (size_t j{ 0 }; j < grade.size() - (1 + zeroCounter); ++j) {
            // compare and swap only if first element < second element
            if (grade[j] < grade[j + 1]) {
                hold = grade[j]; // temp holder for first element
                grade[j] = grade[j + 1]; // assign first element as second
                grade[j + 1] = hold; // assign second element as first
            }
        }
    }
    // function call of each tasks
    cout << "\nMaximum Grade: " << maxGrade(grade, zeroCounter); // maximum grade function
    cout << "\nMinimum Grade: " << minGrade(grade, zeroCounter); // minimum grade function
    cout << "\nAverage Grade: " << avgGrade(grade, zeroCounter); // average grade function
    cout << "\nMedian Grade: " << medGrade(grade, zeroCounter); // median grade function
    cout << "\nNumber of A Grade: " << gradeA(grade, zeroCounter); // grades A function
    cout << "\nNumber of B Grade: " << gradeB(grade, zeroCounter); // grades B function
    cout << "\nNumber of C Grade: " << gradeC(grade, zeroCounter); // grades C function
    cout << "\nNumber of D Grade: " << gradeD(grade, zeroCounter); // grades D function
    cout << "\nNumber of F Grade: " << gradeF(grade, zeroCounter); // grades F function
```

```
Number of grades to enter: 17
Enter grade: 10
Enter grade: 20
Enter grade: 30
Enter grade: 40
Enter grade: 50
Enter grade: 60
Enter grade: 70
Enter grade: 80
Enter grade: 90
Enter grade: 100
Enter grade: 15
Enter grade: 25
Enter grade: 35
Enter grade: 45
Enter grade: 55
Enter grade: 65
Enter grade: 75

Maximum Grade: 100.00
Minimum Grade: 10.00
Average Grade: 50.88
Median Grade: 50.00
Number of A Grade: 2
Number of B Grade: 2
Number of C Grade: 3
Number of D Grade: 3
Number of F Grade: 7

E:\vsCode\Project Location\COEN
ode 0.
To automatically close the cons
le when debugging stops.
Press any key to close this wir
```

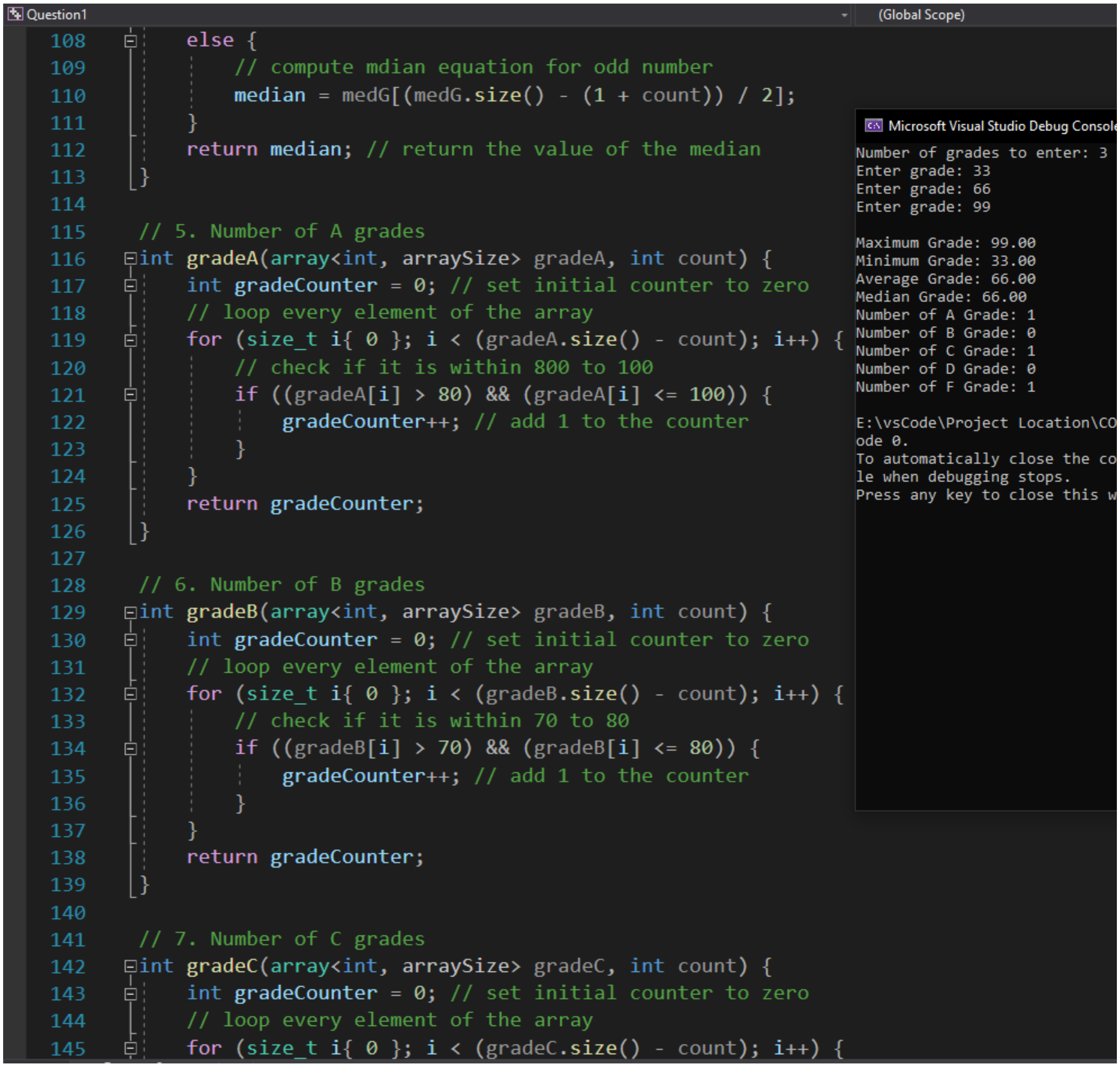
Question 1: (Part 3)



```cpp
    cout << "\nNumber of D Grade: " << gradeD(grade, zeroCounter); // grades D function
    cout << "\nNumber of F Grade: " << gradeF(grade, zeroCounter); // grades F function
    cout << endl;
    return 0;
}

// 1. Maximum grade function
float maxGrade(array<int, arraySize> maxG, int count) {
    return maxG[0]; // return the biggest integer of the ordered array
}

// 2. Minimum grade function
float minGrade(array<int, arraySize> minG, int count) {
    return minG[(minG.size()-(1+count))]; // return the smallest integer
}

// 3. Average grade function
float avgGrade(array<int, arraySize> avgG, int count) {
    float sum = 0; // set the initial sum of all grade as zero
    // for loop that sums all inputed grades
    for (size_t i{ 0 }; i < avgG.size() - count; i++) {
        sum += avgG[i];
    }
    // the average is calculated by dividing the total sum by the amount of inputs
    return sum / (avgG.size() - count); // return the average
}

// 4. Median grade function
float medGrade(array<int, arraySize> medG, int count) {
    float median = 0.0; // set initial value of median to zero

    // check if array size is odd or even
    if ((medG.size() - count) % 2 == 0) {
        // compute median equation for even number
        median = ((medG[(medG.size()-count)/2]) + (medG[(medG.size() - (1 + count)) / 2]))/2.0;
    }
    else {
        // compute mdian equation for odd number
```

```
Number of grades to enter: 5
Enter grade: 13
Enter grade: 63
Enter grade: 84
Enter grade: 39
Enter grade: 89

Maximum Grade: 89.00
Minimum Grade: 13.00
Average Grade: 57.60
Median Grade: 63.00
Number of A Grade: 2
Number of B Grade: 0
Number of C Grade: 1
Number of D Grade: 0
Number of F Grade: 2

E:\vsCode\Project Location\CC
ode 0.
To automatically close the co
le when debugging stops.
Press any key to close this w
```

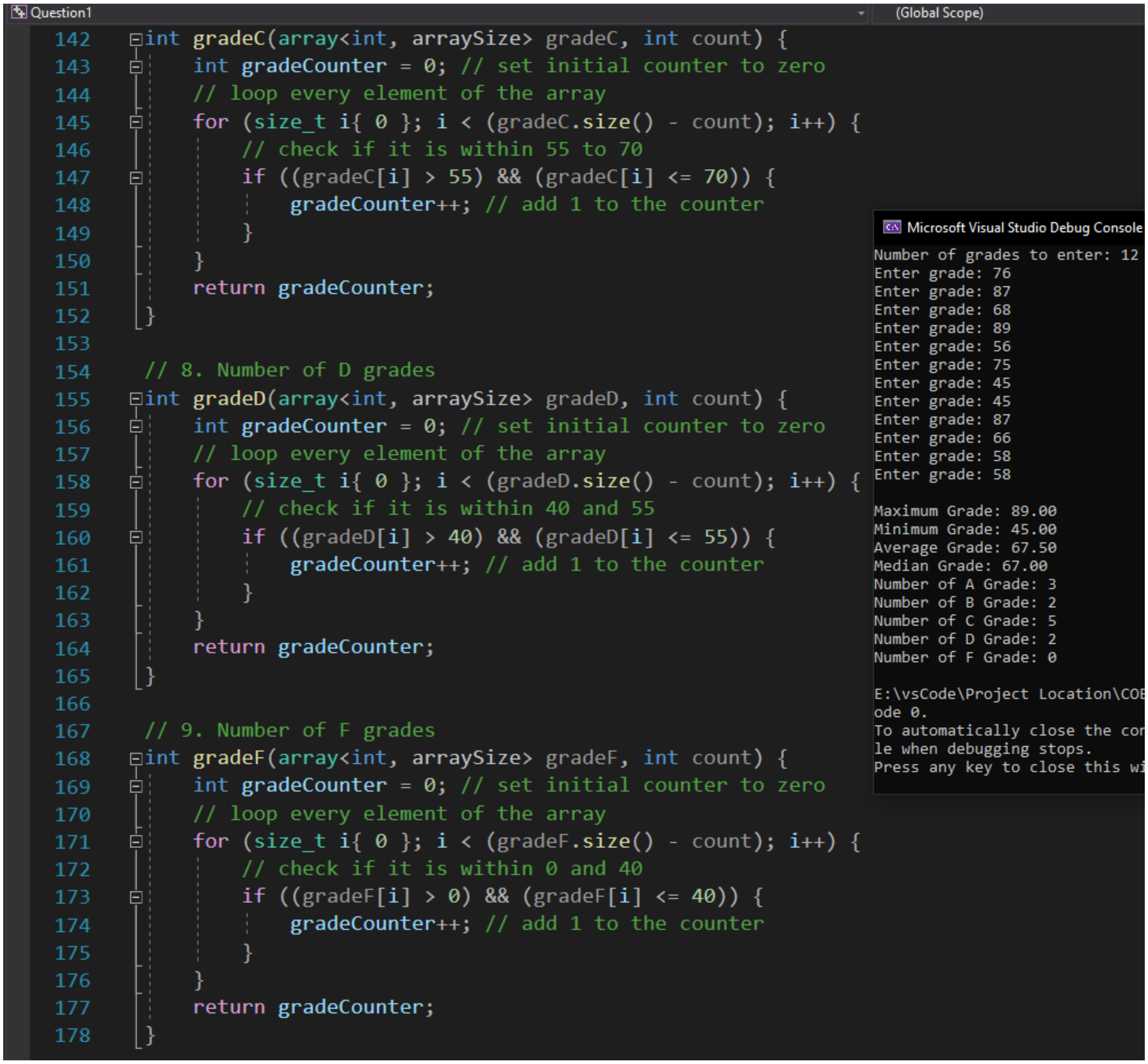
Question 1: (Part 4)



```
    else {
        // compute mdian equation for odd number
        median = medG[(medG.size() - (1 + count)) / 2];
    }
    return median; // return the value of the median
}

// 5. Number of A grades
int gradeA(array<int, arraySize> gradeA, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeA.size() - count); i++) {
        // check if it is within 800 to 100
        if ((gradeA[i] > 80) && (gradeA[i] <= 100)) {
            gradeCounter++; // add 1 to the counter
        }
    }
    return gradeCounter;
}

// 6. Number of B grades
int gradeB(array<int, arraySize> gradeB, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeB.size() - count); i++) {
        // check if it is within 70 to 80
        if ((gradeB[i] > 70) && (gradeB[i] <= 80)) {
            gradeCounter++; // add 1 to the counter
        }
    }
    return gradeCounter;
}

// 7. Number of C grades
int gradeC(array<int, arraySize> gradeC, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeC.size() - count); i++) {
```

```
Number of grades to enter: 3
Enter grade: 33
Enter grade: 66
Enter grade: 99

Maximum Grade: 99.00
Minimum Grade: 33.00
Average Grade: 66.00
Median Grade: 66.00
Number of A Grade: 1
Number of B Grade: 0
Number of C Grade: 1
Number of D Grade: 0
Number of F Grade: 1

E:\vsCode\Project Location\CO
ode 0.
To automatically close the co
le when debugging stops.
Press any key to close this w
```

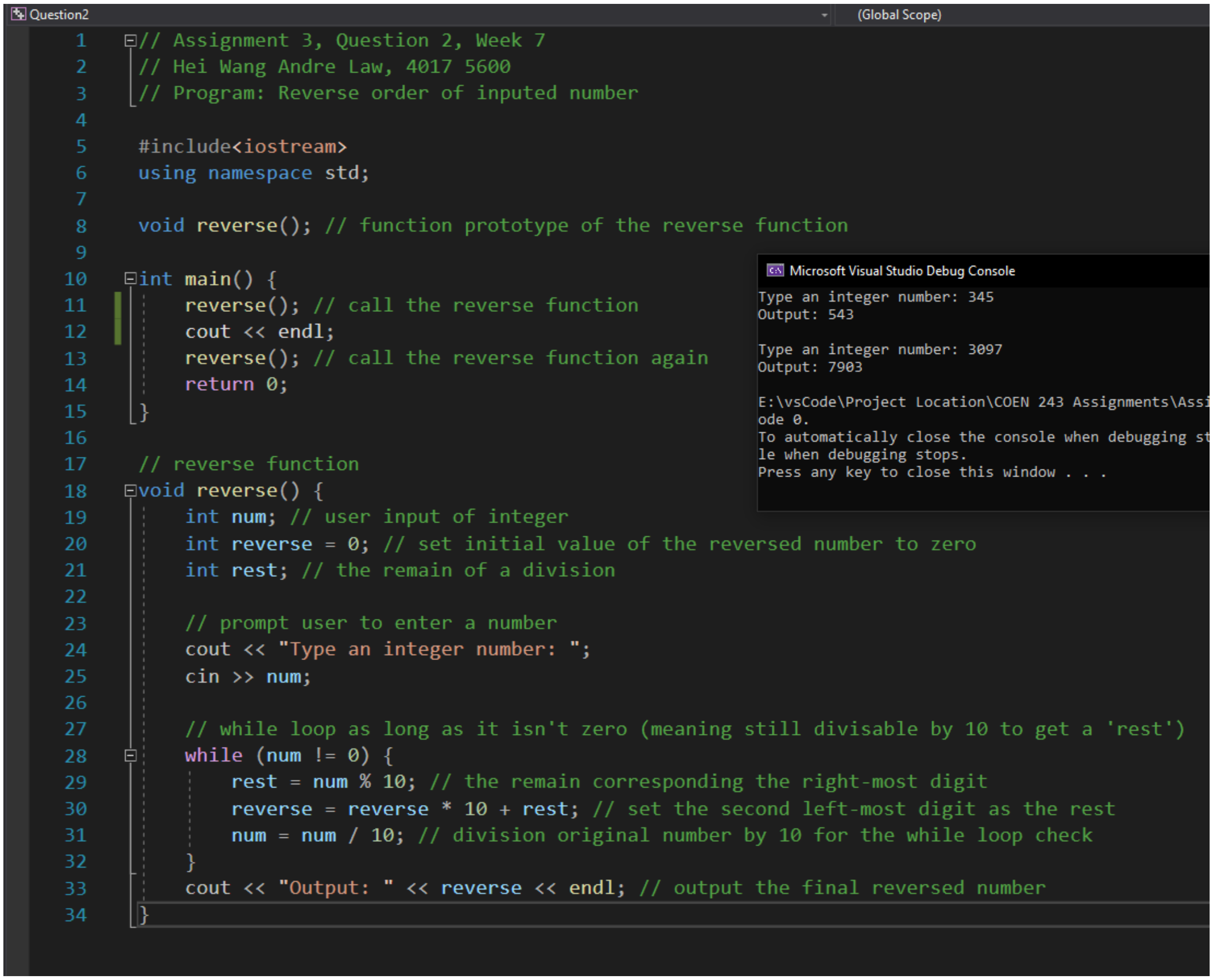
Question 1: (Part 5)



```cpp
int gradeC(array<int, arraySize> gradeC, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeC.size() - count); i++) {
        // check if it is within 55 to 70
        if ((gradeC[i] > 55) && (gradeC[i] <= 70)) {
            gradeCounter++; // add 1 to the counter
        }
    }
    return gradeCounter;
}

// 8. Number of D grades
int gradeD(array<int, arraySize> gradeD, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeD.size() - count); i++) {
        // check if it is within 40 and 55
        if ((gradeD[i] > 40) && (gradeD[i] <= 55)) {
            gradeCounter++; // add 1 to the counter
        }
    }
    return gradeCounter;
}

// 9. Number of F grades
int gradeF(array<int, arraySize> gradeF, int count) {
    int gradeCounter = 0; // set initial counter to zero
    // loop every element of the array
    for (size_t i{ 0 }; i < (gradeF.size() - count); i++) {
        // check if it is within 0 and 40
        if ((gradeF[i] > 0) && (gradeF[i] <= 40)) {
            gradeCounter++; // add 1 to the counter
        }
    }
    return gradeCounter;
}
```

```
Number of grades to enter: 12
Enter grade: 76
Enter grade: 87
Enter grade: 68
Enter grade: 89
Enter grade: 56
Enter grade: 75
Enter grade: 45
Enter grade: 45
Enter grade: 87
Enter grade: 66
Enter grade: 58
Enter grade: 58

Maximum Grade: 89.00
Minimum Grade: 45.00
Average Grade: 67.50
Median Grade: 67.00
Number of A Grade: 3
Number of B Grade: 2
Number of C Grade: 5
Number of D Grade: 2
Number of F Grade: 0

E:\vsCode\Project Location\COE
ode 0.
To automatically close the cor
le when debugging stops.
Press any key to close this wi
```

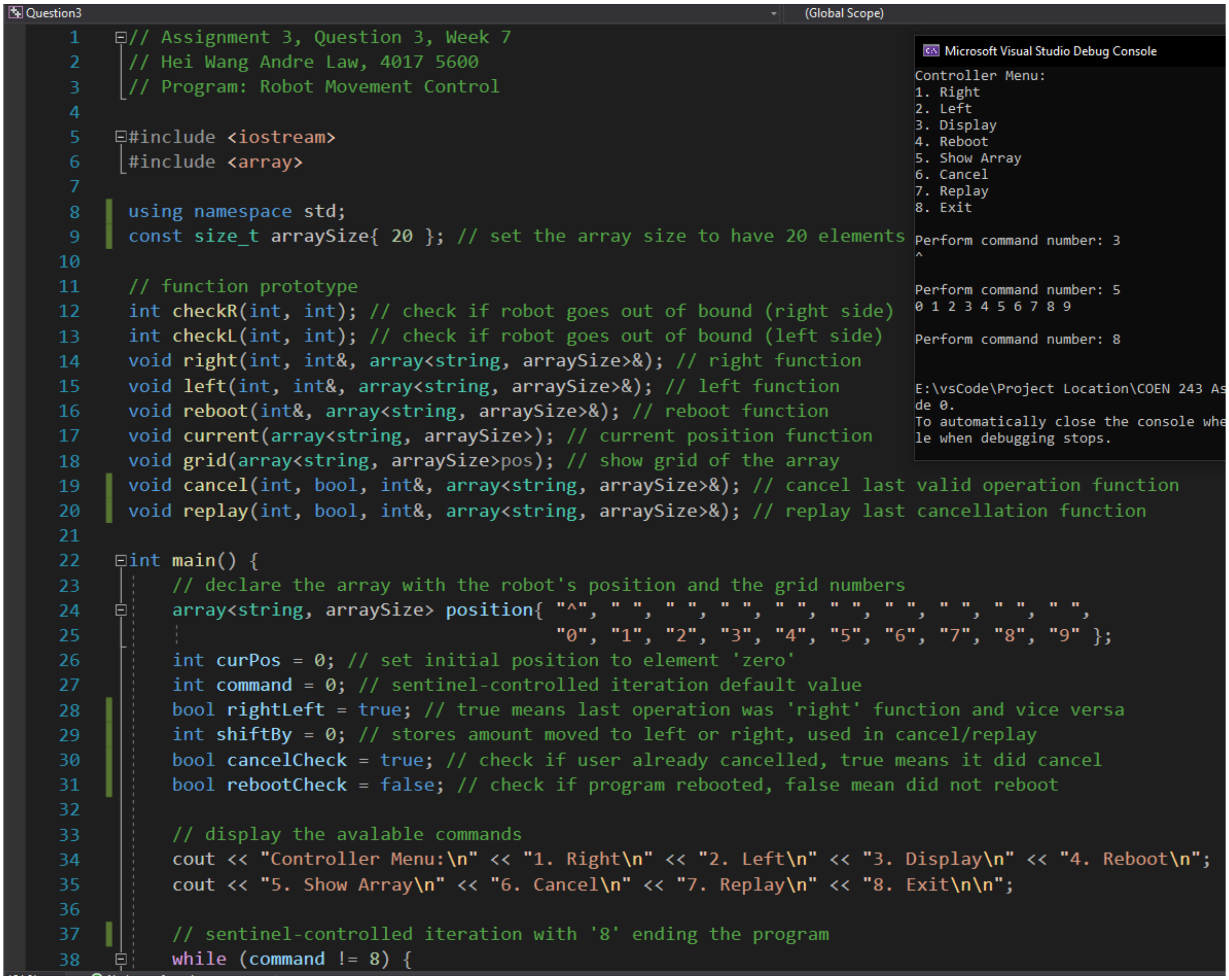
Question 2:



```cpp
// Assignment 3, Question 2, Week 7
// Hei Wang Andre Law, 4017 5600
// Program: Reverse order of inputed number

#include<iostream>
using namespace std;

void reverse(); // function prototype of the reverse function

int main() {
    reverse(); // call the reverse function
    cout << endl;
    reverse(); // call the reverse function again
    return 0;
}

// reverse function
void reverse() {
    int num; // user input of integer
    int reverse = 0; // set initial value of the reversed number to zero
    int rest; // the remain of a division

    // prompt user to enter a number
    cout << "Type an integer number: ";
    cin >> num;

    // while loop as long as it isn't zero (meaning still divisable by 10 to get a 'rest')
    while (num != 0) {
        rest = num % 10; // the remain corresponding the right-most digit
        reverse = reverse * 10 + rest; // set the second left-most digit as the rest
        num = num / 10; // division original number by 10 for the while loop check
    }
    cout << "Output: " << reverse << endl; // output the final reversed number
}
```

```
Type an integer number: 345
Output: 543

Type an integer number: 3097
Output: 7903

E:\vsCode\Project Location\COEN 243 Assignments\Assi
ode 0.
To automatically close the console when debugging st
le when debugging stops.
Press any key to close this window . . .
```

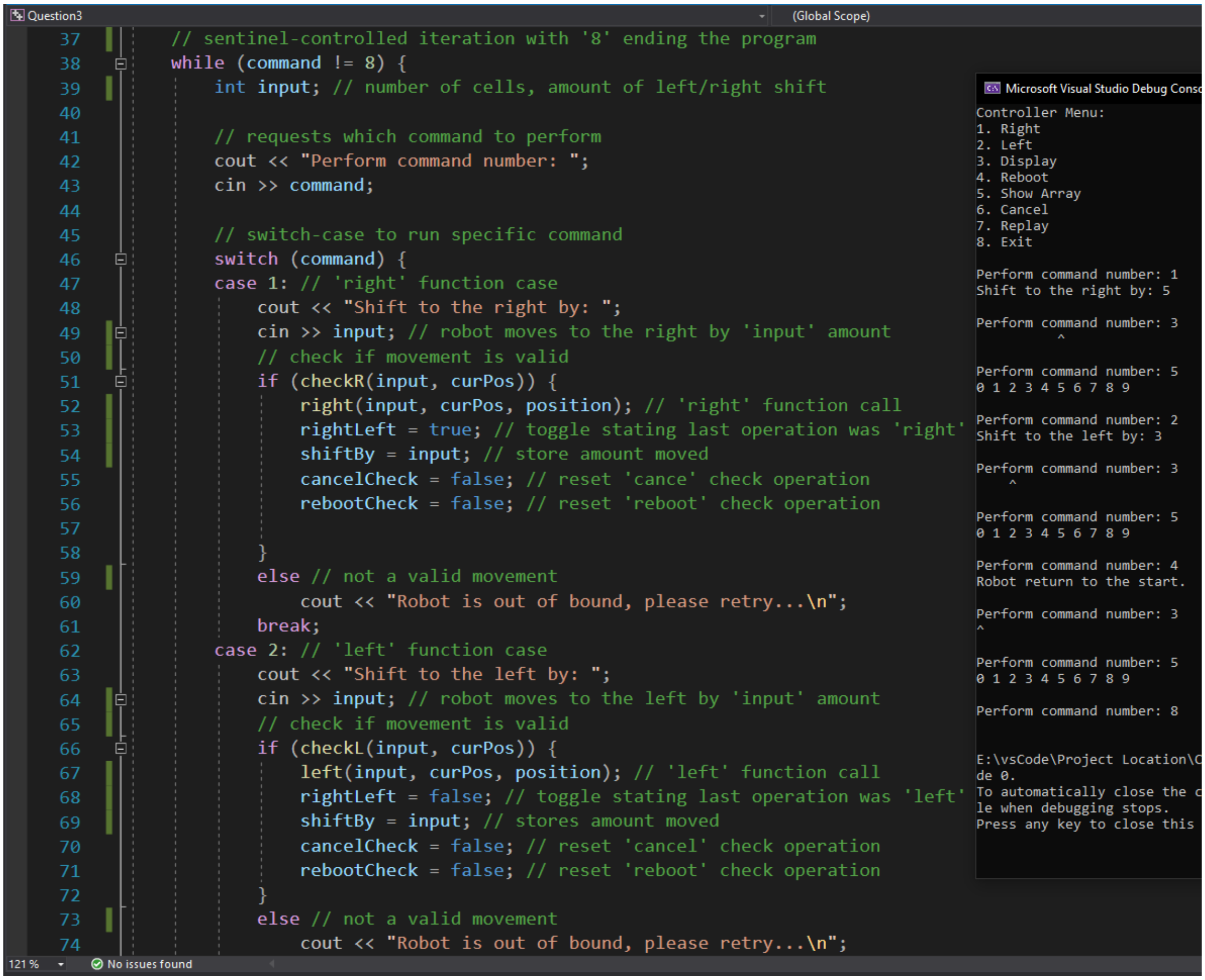
Question 3: (Part 1)



```cpp
// Assignment 3, Question 3, Week 7
// Hei Wang Andre Law, 4017 5600
// Program: Robot Movement Control

#include <iostream>
#include <array>

using namespace std;
const size_t arraySize{ 20 }; // set the array size to have 20 elements

// function prototype
int checkR(int, int); // check if robot goes out of bound (right side)
int checkL(int, int); // check if robot goes out of bound (left side)
void right(int, int&, array<string, arraySize>&); // right function
void left(int, int&, array<string, arraySize>&); // left function
void reboot(int&, array<string, arraySize>&); // reboot function
void current(array<string, arraySize>); // current position function
void grid(array<string, arraySize>pos); // show grid of the array
void cancel(int, bool, int&, array<string, arraySize>&); // cancel last valid operation function
void replay(int, bool, int&, array<string, arraySize>&); // replay last cancellation function

int main() {
    // declare the array with the robot's position and the grid numbers
    array<string, arraySize> position{ "^", " ", " ", " ", " ", " ", " ", " ", " ", " ",
                                       "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" };
    int curPos = 0; // set initial position to element 'zero'
    int command = 0; // sentinel-controlled iteration default value
    bool rightLeft = true; // true means last operation was 'right' function and vice versa
    int shiftBy = 0; // stores amount moved to left or right, used in cancel/replay
    bool cancelCheck = true; // check if user already cancelled, true means it did cancel
    bool rebootCheck = false; // check if program rebooted, false mean did not reboot

    // display the avalable commands
    cout << "Controller Menu:\n" << "1. Right\n" << "2. Left\n" << "3. Display\n" << "4. Reboot\n";
    cout << "5. Show Array\n" << "6. Cancel\n" << "7. Replay\n" << "8. Exit\n\n";

    // sentinel-controlled iteration with '8' ending the program
    while (command != 8) {
```

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 3
^


Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\COEN 243 As
de 0.
To automatically close the console whe
le when debugging stops.
```

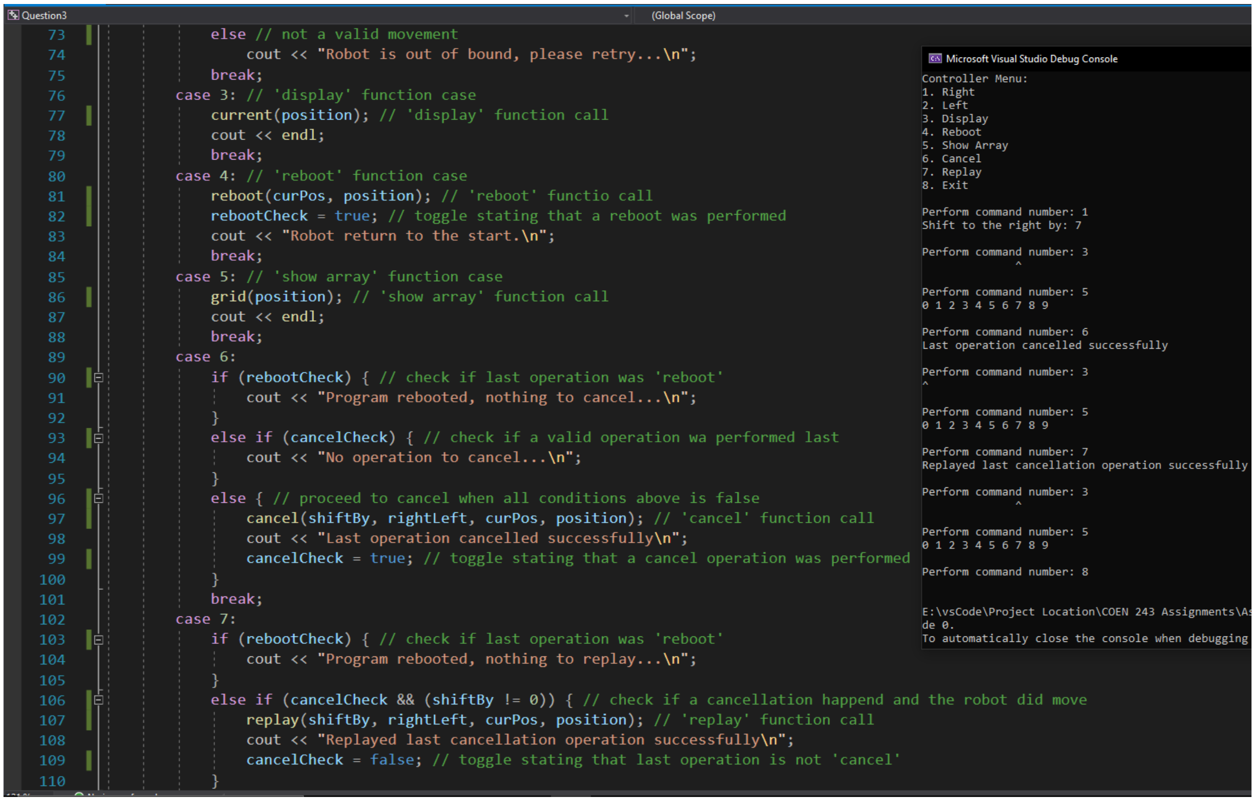
Question 3: (Part 2)



```cpp
    // sentinel-controlled iteration with '8' ending the program
    while (command != 8) {
        int input; // number of cells, amount of left/right shift

        // requests which command to perform
        cout << "Perform command number: ";
        cin >> command;

        // switch-case to run specific command
        switch (command) {
        case 1: // 'right' function case
            cout << "Shift to the right by: ";
            cin >> input; // robot moves to the right by 'input' amount
            // check if movement is valid
            if (checkR(input, curPos)) {
                right(input, curPos, position); // 'right' function call
                rightLeft = true; // toggle stating last operation was 'right'
                shiftBy = input; // store amount moved
                cancelCheck = false; // reset 'cance' check operation
                rebootCheck = false; // reset 'reboot' check operation

            }
            else // not a valid movement
                cout << "Robot is out of bound, please retry...\n";
            break;
        case 2: // 'left' function case
            cout << "Shift to the left by: ";
            cin >> input; // robot moves to the left by 'input' amount
            // check if movement is valid
            if (checkL(input, curPos)) {
                left(input, curPos, position); // 'left' function call
                rightLeft = false; // toggle stating last operation was 'left'
                shiftBy = input; // stores amount moved
                cancelCheck = false; // reset 'cancel' check operation
                rebootCheck = false; // reset 'reboot' check operation
            }
            else // not a valid movement
                cout << "Robot is out of bound, please retry...\n";
```

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 1
Shift to the right by: 5

Perform command number: 3
          ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 2
Shift to the left by: 3

Perform command number: 3
    ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 4
Robot return to the start.

Perform command number: 3
^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\C
de 0.
To automatically close the c
le when debugging stops.
Press any key to close this
```

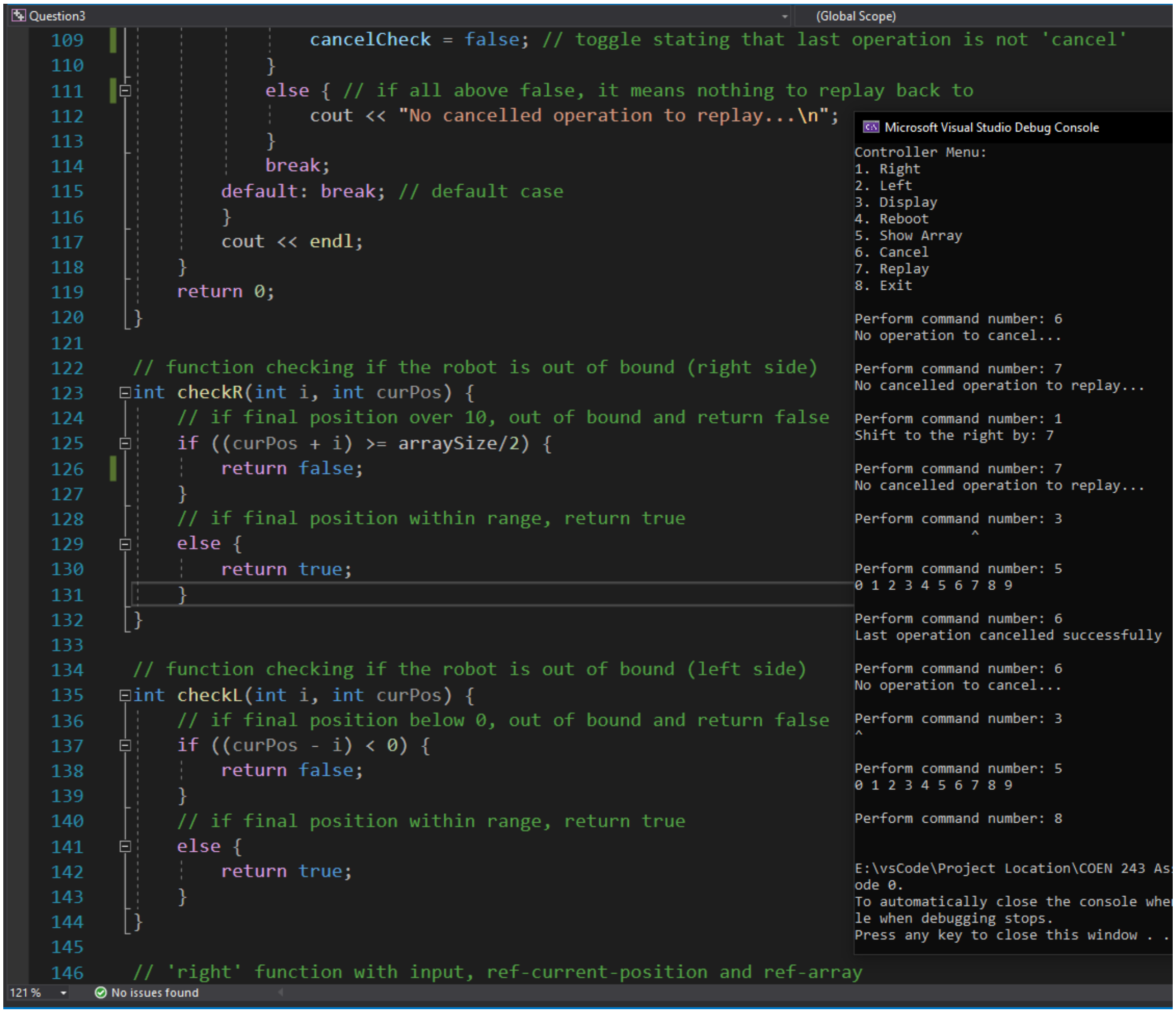
Question 3: (Part 3)



```cpp
            else // not a valid movement
                cout << "Robot is out of bound, please retry...\n";
            break;
        case 3: // 'display' function case
            current(position); // 'display' function call
            cout << endl;
            break;
        case 4: // 'reboot' function case
            reboot(curPos, position); // 'reboot' functio call
            rebootCheck = true; // toggle stating that a reboot was performed
            cout << "Robot return to the start.\n";
            break;
        case 5: // 'show array' function case
            grid(position); // 'show array' function call
            cout << endl;
            break;
        case 6:
            if (rebootCheck) { // check if last operation was 'reboot'
                cout << "Program rebooted, nothing to cancel...\n";
            }
            else if (cancelCheck) { // check if a valid operation wa performed last
                cout << "No operation to cancel...\n";
            }
            else { // proceed to cancel when all conditions above is false
                cancel(shiftBy, rightLeft, curPos, position); // 'cancel' function call
                cout << "Last operation cancelled successfully\n";
                cancelCheck = true; // toggle stating that a cancel operation was performed
            }
            break;
        case 7:
            if (rebootCheck) { // check if last operation was 'reboot'
                cout << "Program rebooted, nothing to replay...\n";
            }
            else if (cancelCheck && (shiftBy != 0)) { // check if a cancellation happend and the robot did move
                replay(shiftBy, rightLeft, curPos, position); // 'replay' function call
                cout << "Replayed last cancellation operation successfully\n";
                cancelCheck = false; // toggle stating that last operation is not 'cancel'
            }
```

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 1
Shift to the right by: 7

Perform command number: 3
              ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 6
Last operation cancelled successfully

Perform command number: 3
^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 7
Replayed last cancellation operation successfully

Perform command number: 3
              ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\COEN 243 Assignments\As
de 0.
To automatically close the console when debugging
```

Question 3: (Part 4)

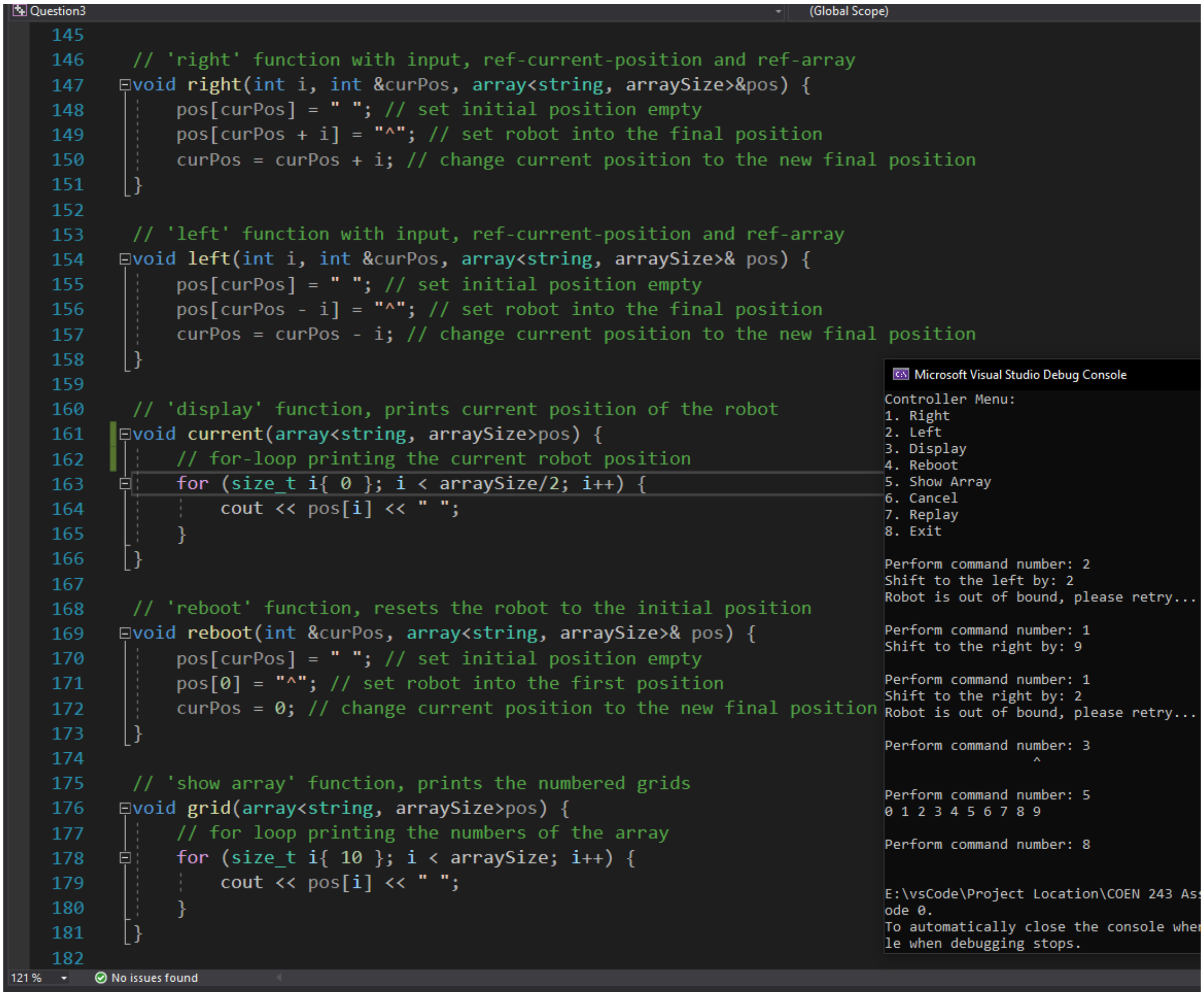


```
                cancelCheck = false; // toggle stating that last operation is not 'cancel'
            }
            else { // if all above false, it means nothing to replay back to
                cout << "No cancelled operation to replay...\n";
            }
            break;
        default: break; // default case
        }
        cout << endl;
    }
    return 0;
}

// function checking if the robot is out of bound (right side)
int checkR(int i, int curPos) {
    // if final position over 10, out of bound and return false
    if ((curPos + i) >= arraySize/2) {
        return false;
    }
    // if final position within range, return true
    else {
        return true;
    }
}

// function checking if the robot is out of bound (left side)
int checkL(int i, int curPos) {
    // if final position below 0, out of bound and return false
    if ((curPos - i) < 0) {
        return false;
    }
    // if final position within range, return true
    else {
        return true;
    }
}

// 'right' function with input, ref-current-position and ref-array
```

Microsoft Visual Studio Debug Console

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 6
No operation to cancel...

Perform command number: 7
No cancelled operation to replay...

Perform command number: 1
Shift to the right by: 7

Perform command number: 7
No cancelled operation to replay...

Perform command number: 3
              ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 6
Last operation cancelled successfully

Perform command number: 6
No operation to cancel...

Perform command number: 3
^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\COEN 243 As
ode 0.
To automatically close the console whe
le when debugging stops.
Press any key to close this window . .
```

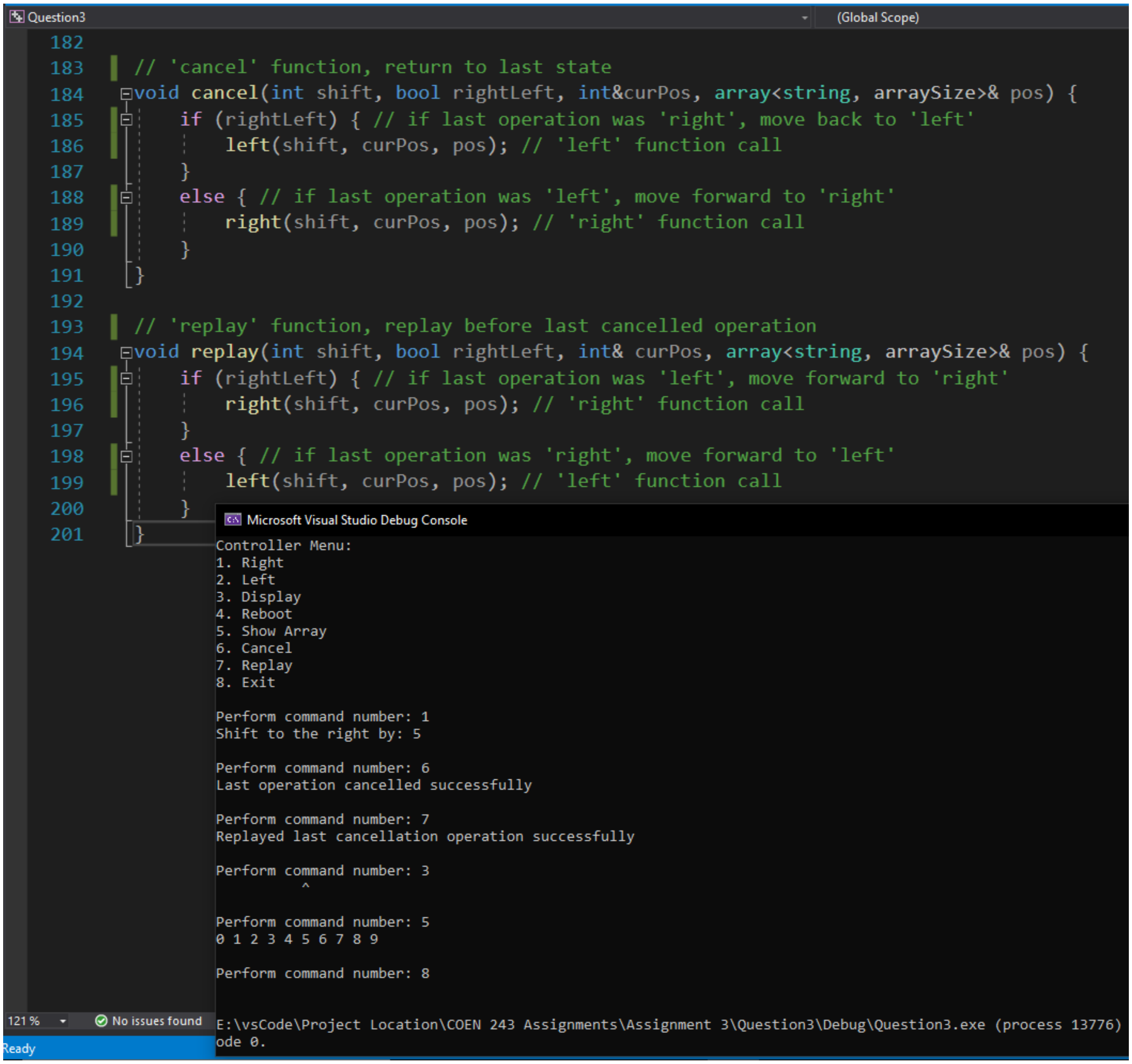
Question 3: (Part 5)



```cpp

// 'right' function with input, ref-current-position and ref-array
void right(int i, int &curPos, array<string, arraySize>&pos) {
    pos[curPos] = " "; // set initial position empty
    pos[curPos + i] = "^"; // set robot into the final position
    curPos = curPos + i; // change current position to the new final position
}

// 'left' function with input, ref-current-position and ref-array
void left(int i, int &curPos, array<string, arraySize>& pos) {
    pos[curPos] = " "; // set initial position empty
    pos[curPos - i] = "^"; // set robot into the final position
    curPos = curPos - i; // change current position to the new final position
}

// 'display' function, prints current position of the robot
void current(array<string, arraySize>pos) {
    // for-loop printing the current robot position
    for (size_t i{ 0 }; i < arraySize/2; i++) {
        cout << pos[i] << " ";
    }
}

// 'reboot' function, resets the robot to the initial position
void reboot(int &curPos, array<string, arraySize>& pos) {
    pos[curPos] = " "; // set initial position empty
    pos[0] = "^"; // set robot into the first position
    curPos = 0; // change current position to the new final position
}

// 'show array' function, prints the numbered grids
void grid(array<string, arraySize>pos) {
    // for loop printing the numbers of the array
    for (size_t i{ 10 }; i < arraySize; i++) {
        cout << pos[i] << " ";
    }
}

```

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 2
Shift to the left by: 2
Robot is out of bound, please retry...

Perform command number: 1
Shift to the right by: 9

Perform command number: 1
Shift to the right by: 2
Robot is out of bound, please retry...

Perform command number: 3
                  ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\COEN 243 As
ode 0.
To automatically close the console whe
le when debugging stops.
```

Question 3: (Part 6)



```cpp

// 'cancel' function, return to last state
void cancel(int shift, bool rightLeft, int&curPos, array<string, arraySize>& pos) {
    if (rightLeft) { // if last operation was 'right', move back to 'left'
        left(shift, curPos, pos); // 'left' function call
    }
    else { // if last operation was 'left', move forward to 'right'
        right(shift, curPos, pos); // 'right' function call
    }
}

// 'replay' function, replay before last cancelled operation
void replay(int shift, bool rightLeft, int& curPos, array<string, arraySize>& pos) {
    if (rightLeft) { // if last operation was 'left', move forward to 'right'
        right(shift, curPos, pos); // 'right' function call
    }
    else { // if last operation was 'right', move forward to 'left'
        left(shift, curPos, pos); // 'left' function call
    }
}
```

```
Controller Menu:
1. Right
2. Left
3. Display
4. Reboot
5. Show Array
6. Cancel
7. Replay
8. Exit

Perform command number: 1
Shift to the right by: 5

Perform command number: 6
Last operation cancelled successfully

Perform command number: 7
Replayed last cancellation operation successfully

Perform command number: 3
          ^

Perform command number: 5
0 1 2 3 4 5 6 7 8 9

Perform command number: 8


E:\vsCode\Project Location\COEN 243 Assignments\Assignment 3\Question3\Debug\Question3.exe (process 13776)
ode 0.
```