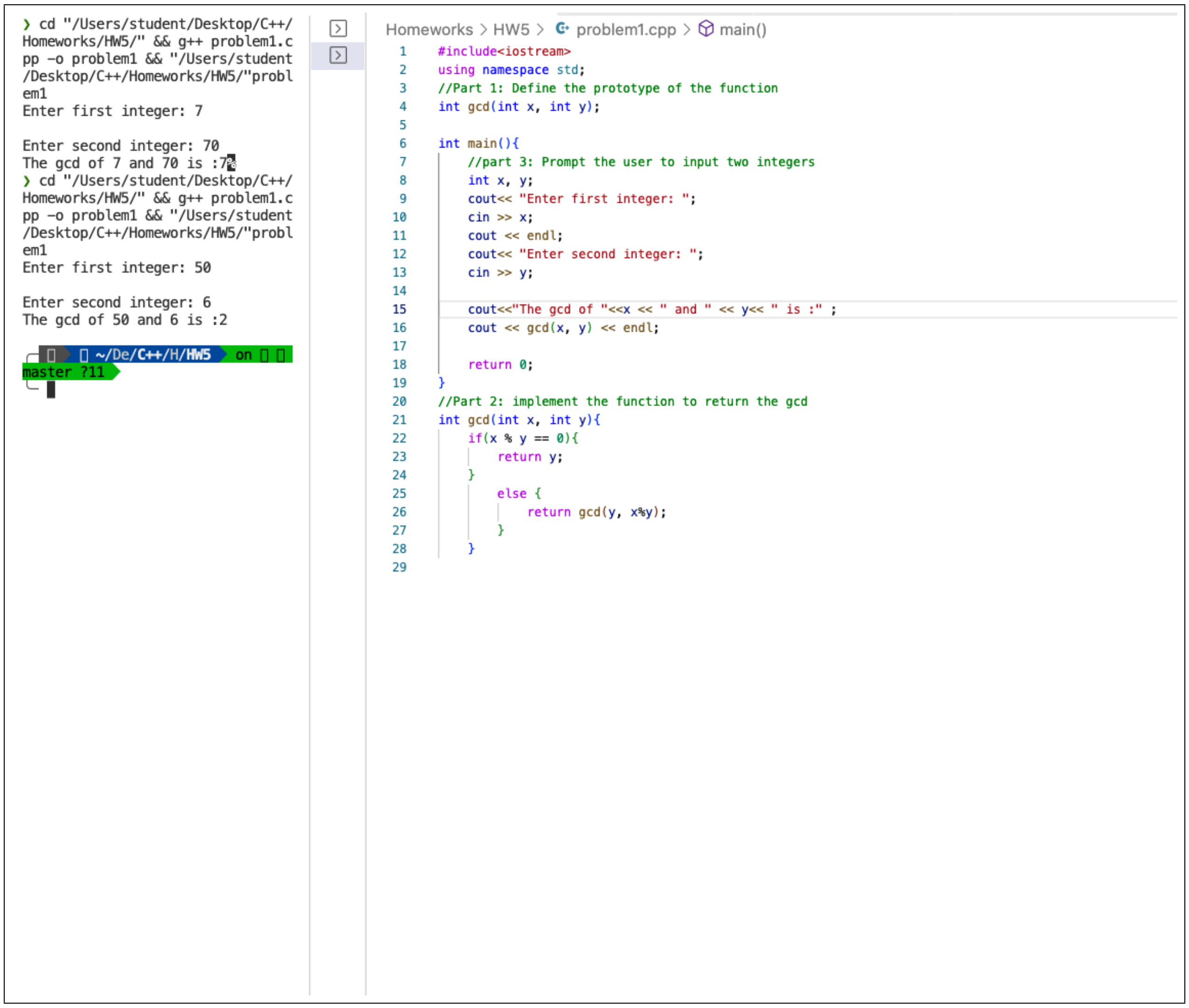
Name: Md Jubair Pantho

UTC ID : WZS444

Course : CPSC 5010

Homework: 5

Problem 1:
Source code and Results:



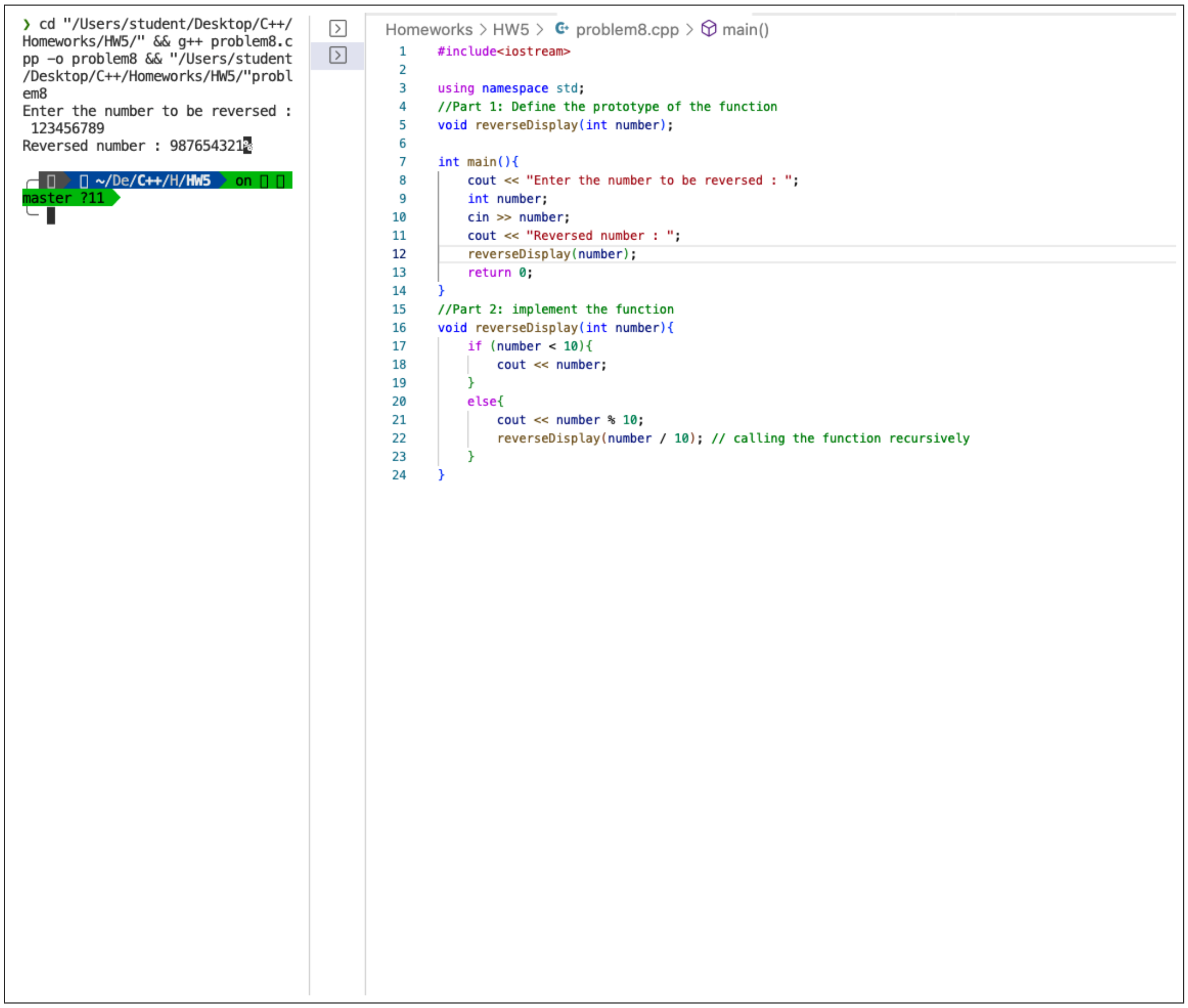
```
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem1.c
pp -o problem1 && "/Users/student
/Desktop/C++/Homeworks/HW5/"probl
em1
Enter first integer: 7

Enter second integer: 70
The gcd of 7 and 70 is :7
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem1.c
pp -o problem1 && "/Users/student
/Desktop/C++/Homeworks/HW5/"probl
em1
Enter first integer: 50

Enter second integer: 6
The gcd of 50 and 6 is :2

 ~/De/C++/H/HW5  on
master ?11
```

Homeworks > HW5 > problem1.cpp > main()

```cpp
#include<iostream>
using namespace std;
//Part 1: Define the prototype of the function
int gcd(int x, int y);

int main(){
    //part 3: Prompt the user to input two integers
    int x, y;
    cout<< "Enter first integer: ";
    cin >> x;
    cout << endl;
    cout<< "Enter second integer: ";
    cin >> y;

    cout<<"The gcd of "<<x << " and " << y<< " is :" ;
    cout << gcd(x, y) << endl;

    return 0;
}
//Part 2: implement the function to return the gcd
int gcd(int x, int y){
    if(x % y == 0){
        return y;
    }
        else {
            return gcd(y, x%y);
        }
    }
```

Please click on the highlighted coding video link : Problem 1

Problem 2:
Source code and Results:



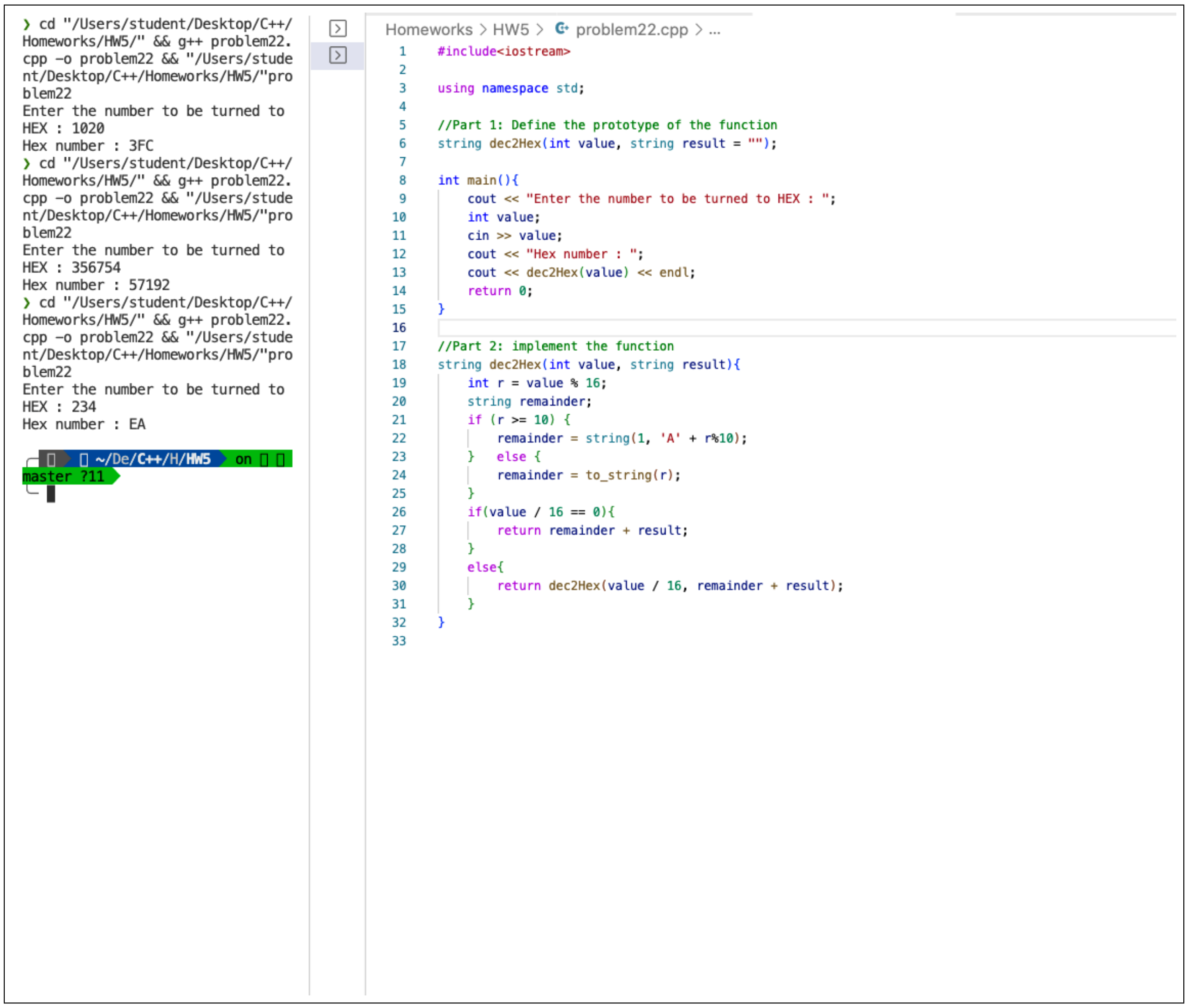
```
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem8.c
pp -o problem8 && "/Users/student
/Desktop/C++/Homeworks/HW5/"probl
em8
Enter the number to be reversed :
 123456789
Reversed number : 987654321%

  ~/De/C++/H/HW5  on
master ?11
```

Homeworks > HW5 > problem8.cpp > main()

```cpp
#include<iostream>

using namespace std;
//Part 1: Define the prototype of the function
void reverseDisplay(int number);

int main(){
    cout << "Enter the number to be reversed : ";
    int number;
    cin >> number;
    cout << "Reversed number : ";
    reverseDisplay(number);
    return 0;
}
//Part 2: implement the function
void reverseDisplay(int number){
    if (number < 10){
        cout << number;
    }
    else{
        cout << number % 10;
        reverseDisplay(number / 10); // calling the function recursively
    }
}
```

Please click on the highlighted coding video link : Problem 2

Problem 3:
Source code and Results:



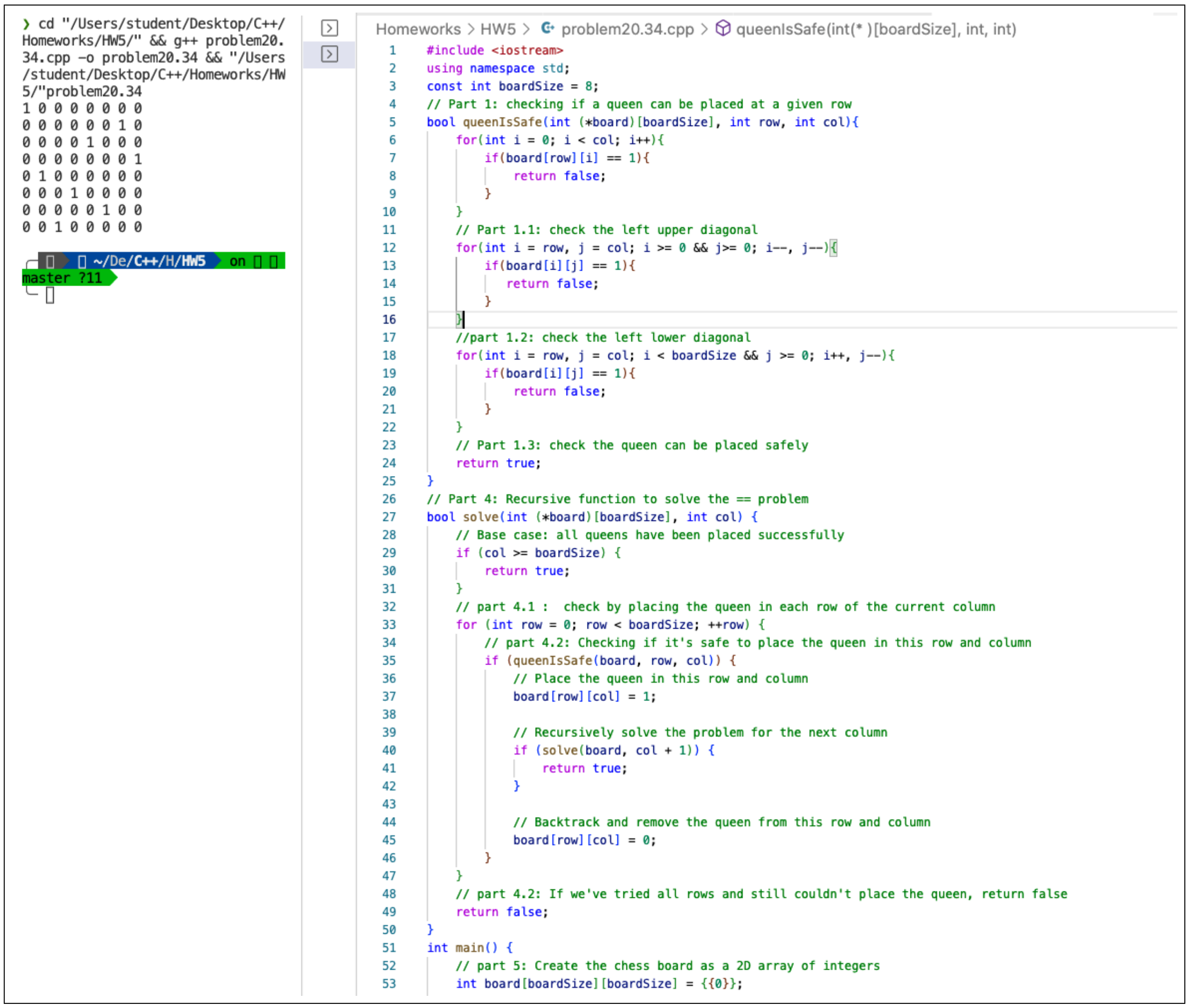
```
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem22.
cpp -o problem22 && "/Users/stude
nt/Desktop/C++/Homeworks/HW5/"pro
blem22
Enter the number to be turned to
HEX : 1020
Hex number : 3FC
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem22.
cpp -o problem22 && "/Users/stude
nt/Desktop/C++/Homeworks/HW5/"pro
blem22
Enter the number to be turned to
HEX : 356754
Hex number : 57192
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem22.
cpp -o problem22 && "/Users/stude
nt/Desktop/C++/Homeworks/HW5/"pro
blem22
Enter the number to be turned to
HEX : 234
Hex number : EA

~/De/C++/H/HW5  on
master ?11
```

Homeworks > HW5 > problem22.cpp > ...

```cpp
#include<iostream>

using namespace std;

//Part 1: Define the prototype of the function
string dec2Hex(int value, string result = "");

int main(){
    cout << "Enter the number to be turned to HEX : ";
    int value;
    cin >> value;
    cout << "Hex number : ";
    cout << dec2Hex(value) << endl;
    return 0;
}

//Part 2: implement the function
string dec2Hex(int value, string result){
    int r = value % 16;
    string remainder;
    if (r >= 10) {
        remainder = string(1, 'A' + r%10);
    }  else {
        remainder = to_string(r);
    }
    if(value / 16 == 0){
        return remainder + result;
    }
    else{
        return dec2Hex(value / 16, remainder + result);
    }
}
```

Please click on the highlighted coding video link : Problem 3

Problem 4.1 :
Source code and Results:



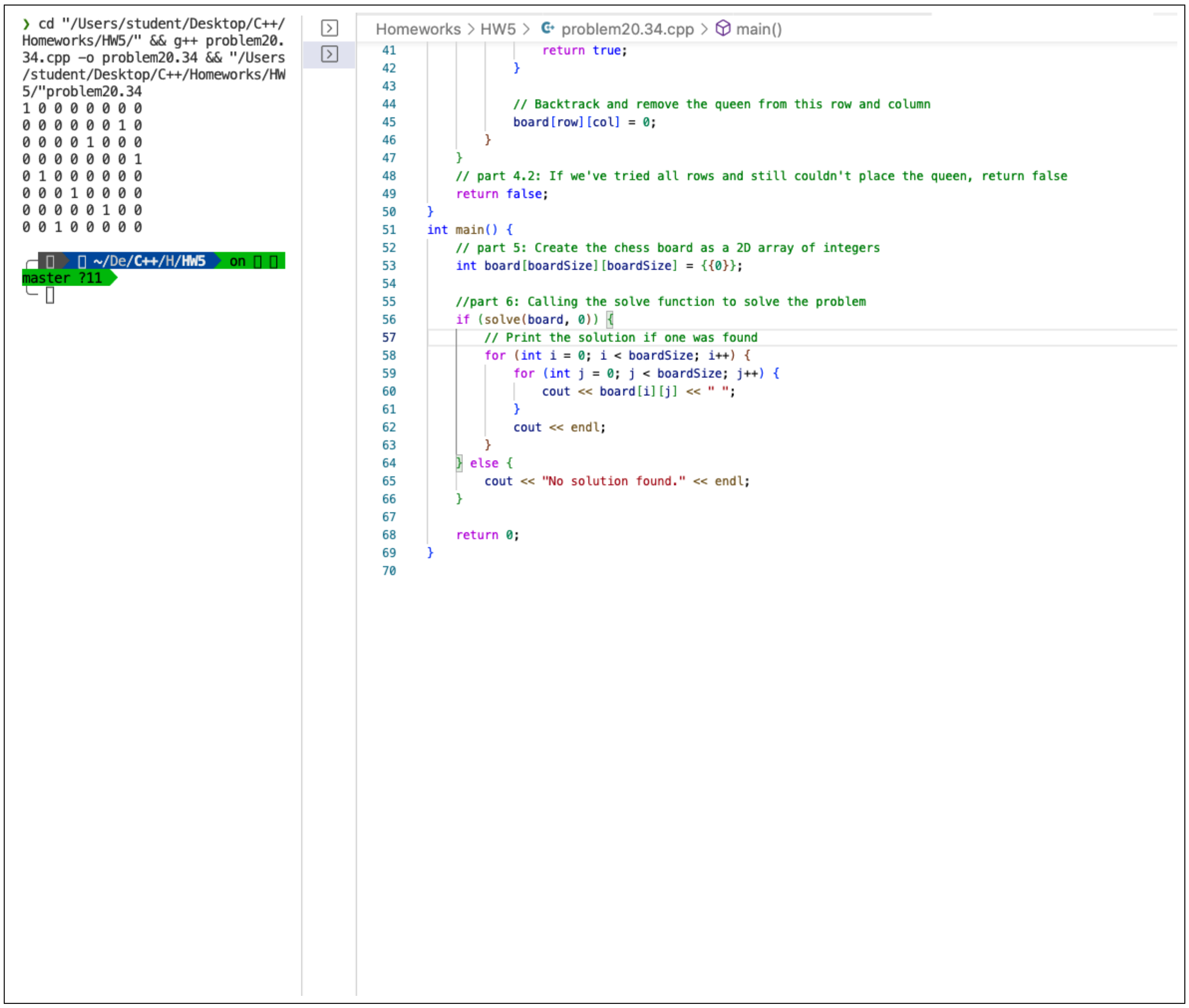
```
❯ cd "/Users/student/Desktop/C++/
Homeworks/HW5/" && g++ problem20.
34.cpp -o problem20.34 && "/Users
/student/Desktop/C++/Homeworks/HW
5/"problem20.34
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0

  ~/De/C++/H/HW5  on
master ?11
```

Homeworks > HW5 > problem20.34.cpp > queenIsSafe(int(*)[boardSize], int, int)

```cpp
#include <iostream>
using namespace std;
const int boardSize = 8;
// Part 1: checking if a queen can be placed at a given row
bool queenIsSafe(int (*board)[boardSize], int row, int col){
    for(int i = 0; i < col; i++){
        if(board[row][i] == 1){
            return false;
        }
    }
    // Part 1.1: check the left upper diagonal
    for(int i = row, j = col; i >= 0 && j>= 0; i--, j--){
        if(board[i][j] == 1){
           return false;
        }
    }
    //part 1.2: check the left lower diagonal
    for(int i = row, j = col; i < boardSize && j >= 0; i++, j--){
        if(board[i][j] == 1){
            return false;
        }
    }
    // Part 1.3: check the queen can be placed safely
    return true;
}
// Part 4: Recursive function to solve the == problem
bool solve(int (*board)[boardSize], int col) {
    // Base case: all queens have been placed successfully
    if (col >= boardSize) {
        return true;
    }
    // part 4.1 :  check by placing the queen in each row of the current column
    for (int row = 0; row < boardSize; ++row) {
        // part 4.2: Checking if it's safe to place the queen in this row and column
        if (queenIsSafe(board, row, col)) {
            // Place the queen in this row and column
            board[row][col] = 1;

            // Recursively solve the problem for the next column
            if (solve(board, col + 1)) {
                return true;
            }

            // Backtrack and remove the queen from this row and column
            board[row][col] = 0;
        }
    }
    // part 4.2: If we've tried all rows and still couldn't place the queen, return false
    return false;
}
int main() {
    // part 5: Create the chess board as a 2D array of integers
    int board[boardSize][boardSize] = {{0}};
```

Please click on the highlighted coding video link : Problem 4.1

Problem 4.2 :
Source code and Results:



```
❯ cd "/Users/student/Desktop/C++/Homeworks/HW5/" && g++ problem20.34.cpp -o problem20.34 && "/Users/student/Desktop/C++/Homeworks/HW5/"problem20.34
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0

 ~/De/C++/H/HW5  on
master ?11
```

Homeworks > HW5 > problem20.34.cpp > main()

```cpp
                return true;
            }

            // Backtrack and remove the queen from this row and column
            board[row][col] = 0;
        }
    }
    // part 4.2: If we've tried all rows and still couldn't place the queen, return false
    return false;
}
int main() {
    // part 5: Create the chess board as a 2D array of integers
    int board[boardSize][boardSize] = {{0}};

    //part 6: Calling the solve function to solve the problem
    if (solve(board, 0)) {
        // Print the solution if one was found
        for (int i = 0; i < boardSize; i++) {
            for (int j = 0; j < boardSize; j++) {
                cout << board[i][j] << " ";
            }
            cout << endl;
        }
    } else {
        cout << "No solution found." << endl;
    }

    return 0;
}
```

Please click on the highlighted coding video link : Problem 4.2