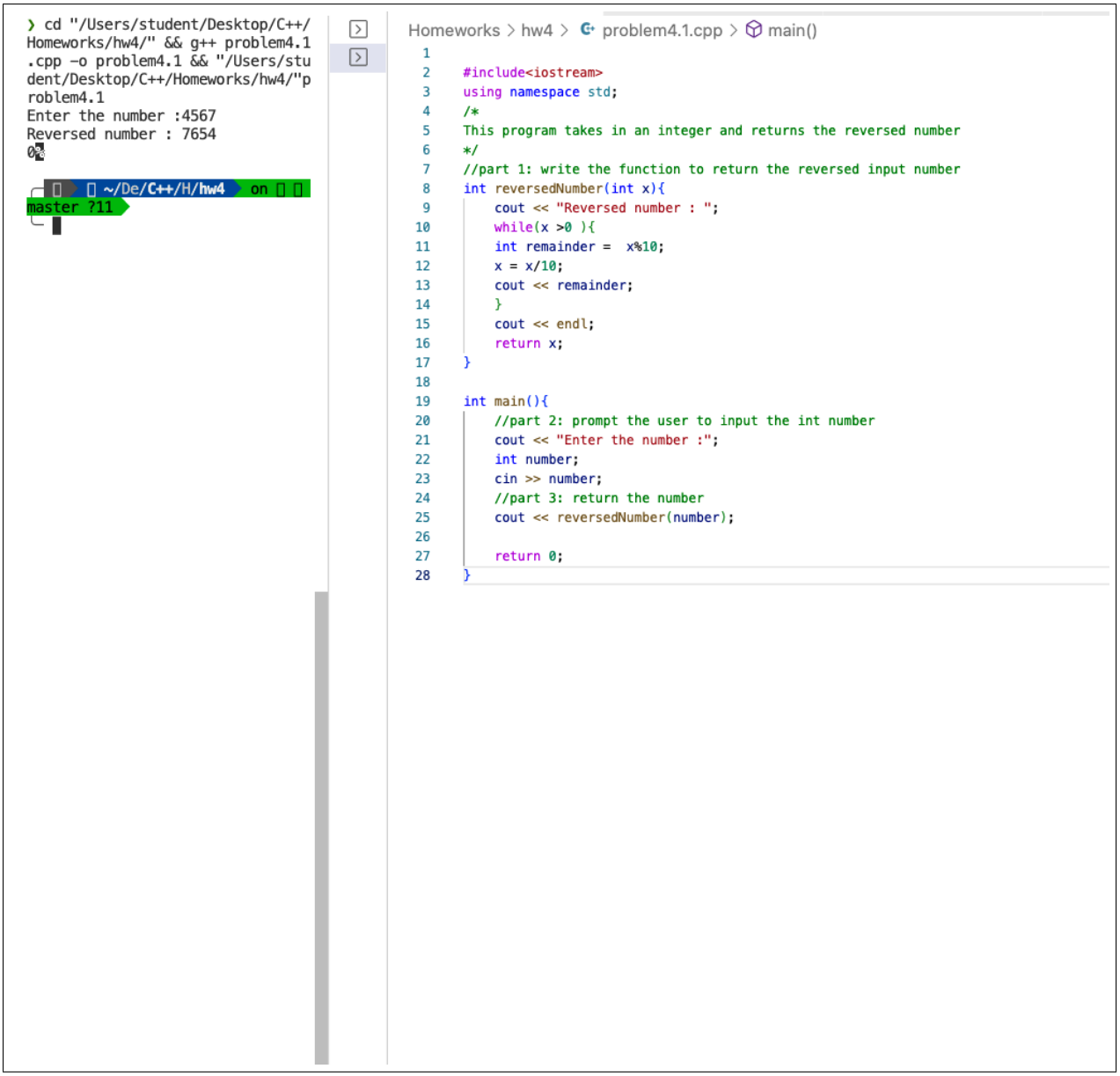
Name: Md Jubair Pantho

UTC ID : WZS444

Course : CPSC 5010

Homework: 4

Problem 4.1:
Source code and Results:



```
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.1.cpp -o problem4.1 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.1
Enter the number :4567
Reversed number : 7654
0
```

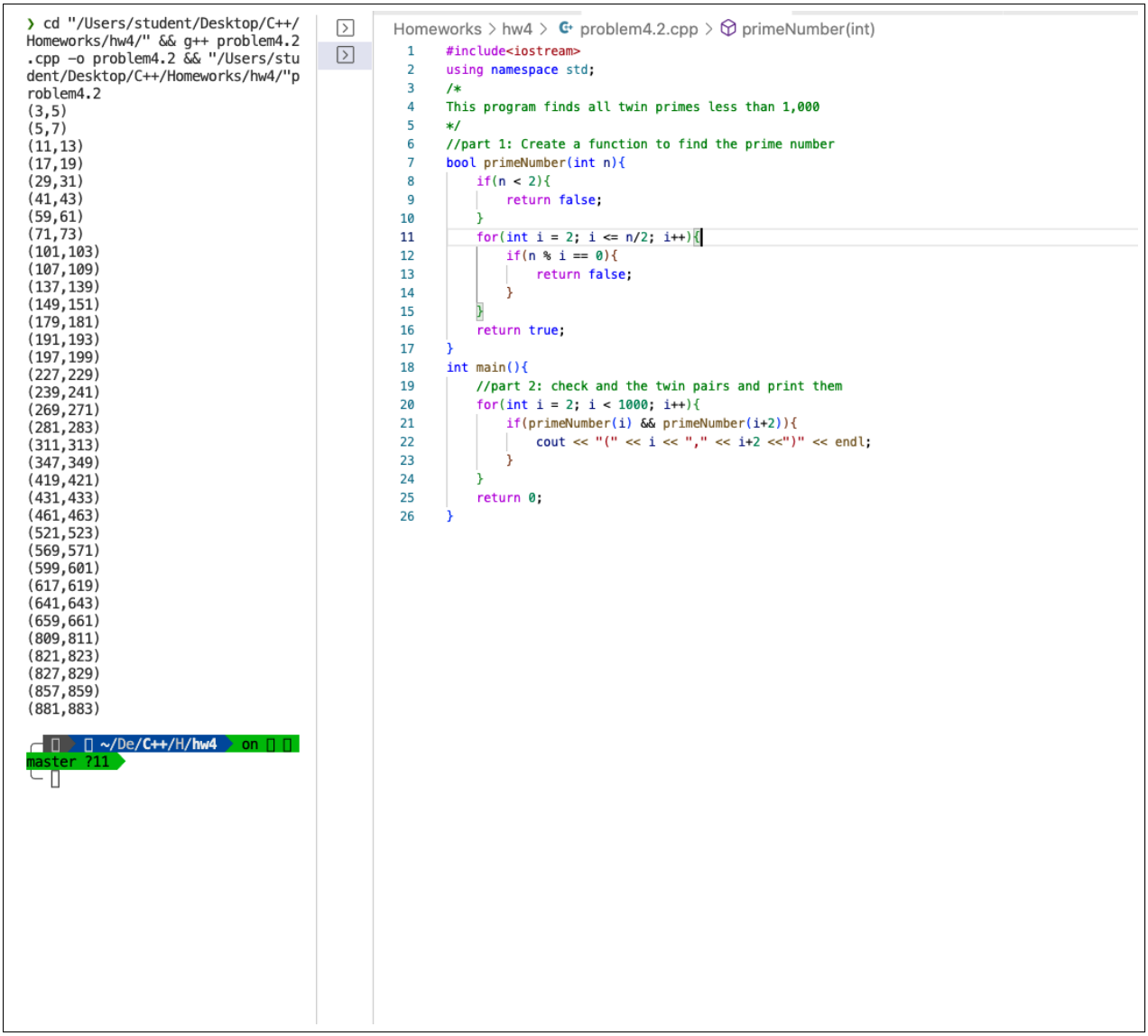
```
 ~/De/C++/H/hw4 on  master ?11
```

Homeworks > hw4 > problem4.1.cpp > main()

```cpp

#include<iostream>
using namespace std;
/*
This program takes in an integer and returns the reversed number
*/
//part 1: write the function to return the reversed input number
int reversedNumber(int x){
    cout << "Reversed number : ";
    while(x >0 ){
    int remainder =  x%10;
    x = x/10;
    cout << remainder;
    }
    cout << endl;
    return x;
}

int main(){
    //part 2: prompt the user to input the int number
    cout << "Enter the number :";
    int number;
    cin >> number;
    //part 3: return the number
    cout << reversedNumber(number);

    return 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/hw4/" && g++ problem4.2.cpp -o problem4.2 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.2
(3,5)
(5,7)
(11,13)
(17,19)
(29,31)
(41,43)
(59,61)
(71,73)
(101,103)
(107,109)
(137,139)
(149,151)
(179,181)
(191,193)
(197,199)
(227,229)
(239,241)
(269,271)
(281,283)
(311,313)
(347,349)
(419,421)
(431,433)
(461,463)
(521,523)
(569,571)
(599,601)
(617,619)
(641,643)
(659,661)
(809,811)
(821,823)
(827,829)
(857,859)
(881,883)
```

```
 ~/De/C++/H/hw4 on  master ?11
```

Homeworks > hw4 > problem4.2.cpp > primeNumber(int)

```cpp
#include<iostream>
using namespace std;
/*
This program finds all twin primes less than 1,000
*/
//part 1: Create a function to find the prime number
bool primeNumber(int n){
    if(n < 2){
        return false;
    }
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            return false;
        }
    }
    return true;
}
int main(){
    //part 2: check and the twin pairs and print them
    for(int i = 2; i < 1000; i++){
        if(primeNumber(i) && primeNumber(i+2)){
            cout << "(" << i << "," << i+2 <<")" << endl;
        }
    }
    return 0;
}
```

Please click on the highlighted coding video link : Problem 4.2

Problem 4.3:
Source code and Results:

```
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.3.cpp -o problem4.3 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.3
Employee id     Total hours
2               42
3               42
4               39
5               28
6               28
7               28
1               26
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.3.cpp -o problem4.3 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.3
Employee id     Total hours
6               39
7               39
1               38
3               35
5               33
4               28
2               27
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.3.cpp -o problem4.3 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.3
Employee id     Total hours
7               40
1               33
3               32
4               32
6               30
5               28
2               21

 ~/De/C++/H/hw4  on  master ?11
```

Homeworks > hw4 > problem4.3.cpp > main()

```cpp
#include<iostream>
#include<stdlib.h>
using namespace std;
/*
This program displays employees and their total hours in decreasing order of the total hours.
*/
// Part 1: Define the data type for an employee
struct Employee
{
    int id;
    int total_hours;
};
int main()
{
    // part 2: Create a 2D array to store the weekly hours for each employee
    srand(time(NULL));
    int **weekly_hours = new int*[7];
    for (int i = 0; i < 7; i++) {
        weekly_hours[i] = new int[7];
            for(int j =0; j <7; j++){
                weekly_hours[i][j] = (rand() % 9) + 1;
            };
        }
    // Part 3: Calculate the total hours for each employee
    Employee employees[7];
    for (int i = 0; i < 7; i++) {
        employees[i].id = i + 1;
        employees[i].total_hours = 0;
        for (int j = 0; j < 7; j++) {
            employees[i].total_hours += weekly_hours[i][j];
        }
    }
    //Part 4: Sort the employees in decreasing order of their hours
    for (int i = 0; i < 7; i++) {
        for (int j = 0; j < 6 ; j++) {
            if (employees[j].total_hours < employees[j + 1].total_hours) {
                Employee temp = employees[j];
                employees[j] = employees[j + 1];
                employees[j + 1] = temp;
            }
        }
    }
    //Part 5: Display the output and release the dynamic array
    cout << "Employee id\tTotal hours" << endl;
    for (int i = 0; i < 7; i++) {
        cout << employees[i].id << "\t\t" << employees[i].total_hours << endl;
    }
    for(int i=0; i<7; i++){
        delete [] weekly_hours[i];
    }
    delete [] weekly_hours;
    return 0;
}
```

Please click on the highlighted coding video link : Problem 4.3

Problem 4.4 (a) :
Source code and Results:

```
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.4.cpp -o problem4.4 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.4
  1 2 3
1  | |
  -----
2  | |
  -----
3  | |
Player X enter your move (row and col): 1
1
  1 2 3
1 X| |
  -----
2  | |
  -----
3  | |
Player O enter your move (row and col): 2
1
  1 2 3
1 X| |
  -----
2 O| |
  -----
3  | |
Player X enter your move (row and col): 2
2
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3  | |
Player O enter your move (row and col): 3
1
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3 O| |
Player X enter your move (row and col): 3
3
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3 O| |X
Player O enter your move (row and
```

Homeworks > hw4 > problem4.4.cpp > checkDraw()

```cpp
#include<iostream>
#include<string>
/*
Tic-Tac-Toe game
*/
using namespace std;
//Part 1: Declare the variables
const int ROWS = 3;
const int COLUMNS = 3;
char gameBoard[ROWS][COLUMNS] = {{' ', ' ', ' '},
{' ',' ',' '},
{' ',' ',' '}};

char player = 'X';
bool gameOver = false;
//Part 2: function for displaying the game board
void boardDisplay() {
  cout << "  1 2 3" << endl;
  cout << "1 " << gameBoard[0][0] << "|" << gameBoard[0][1] << "|" << gameBoard[0][2] << endl;
  cout << "  -----" << endl;
  cout << "2 " << gameBoard[1][0] << "|" << gameBoard[1][1] << "|" << gameBoard[1][2] << endl;
  cout << "  -----" << endl;
  cout << "3 " << gameBoard[2][0] << "|" << gameBoard[2][1] << "|" << gameBoard[2][2] << endl;
}
//part 3: function for updating the board after each palyers input
void boardUpdate(int row, int col){
    gameBoard[row][col] = player;
    if(player == 'X'){
        player = 'O';
    }
    else{
        player = 'X';
    }
}
// part 4: function for checking win situations
bool checkWin(){
    //part 4.1 : check for win in the rows
    for(int i = 0; i < ROWS; i++){
        if(gameBoard[i][0] == player && gameBoard[i][1] == player && gameBoard[i][2] == player){
            return true;
        }
    }
    //part 4.2: check for win in the columns
    for(int j = 0; j < COLUMNS; j++){
        if(gameBoard[0][j] == player && gameBoard[1][j] == player && gameBoard[2][j] == player){
            return true;
        }
    }
    //part 4.3: check for win in the diagonals
    if(gameBoard[0][0] == player && gameBoard[1][1] == player && gameBoard[2][2] == player){
        return true;
    }
    if(gameBoard[0][2] == player && gameBoard[1][1] == player && gameBoard[2][0] == player){
```

Please click on the highlighted coding video link : Problem 4.4 (a)

Problem 4.4 (b) :
Source code and Results:

```
❯ cd "/Users/student/Desktop/C++/Homeworks/hw4/" && g++ problem4.4.cpp -o problem4.4 && "/Users/student/Desktop/C++/Homeworks/hw4/"problem4.4
  1 2 3
1  | |
  -----
2  | |
  -----
3  | |
Player X enter your move (row and col): 1
1
  1 2 3
1 X| |
  -----
2  | |
  -----
3  | |
Player O enter your move (row and col): 2
1
  1 2 3
1 X| |
  -----
2 O| |
  -----
3  | |
Player X enter your move (row and col): 2
2
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3  | |
Player O enter your move (row and col): 3
1
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3 O| |
Player X enter your move (row and col): 3
3
  1 2 3
1 X| |
  -----
2 O|X|
  -----
3 O| |X
Player O enter your move (row and
```

Homeworks > hw4 > problem4.4.cpp > checkDraw()

```cpp
    for(int i = 0; i < ROWS; i++){
        if(gameBoard[i][0] == player && gameBoard[i][1] == player && gameBoard[i][2] == player){
            return true;
        }
    }
    //part 4.2: check for win in the columns
    for(int j = 0; j < COLUMNS; j++){
        if(gameBoard[0][j] == player && gameBoard[1][j] == player && gameBoard[2][j] == player){
            return true;
        }
    }
    //part 4.3: check for win in the diagonals
    if(gameBoard[0][0] == player && gameBoard[1][1] == player && gameBoard[2][2] == player){
        return true;
    }
    if(gameBoard[0][2] == player && gameBoard[1][1] == player && gameBoard[2][0] == player){
        return true;
    }
    return false;
}
// part 5: function for checking a draw
bool checkDraw(){
    for(int i = 0; i < ROWS; i++){
        for(int j = 0; j < COLUMNS; j++){
            if(gameBoard[i][j] == ' '){
                return false;
            }
        }
    }
    return true;
}
int main(){
    while(!gameOver){
        boardDisplay();
        cout << "Player " << player << " enter your move (row and col): ";
        int row, col;
        cin >> row >> col;
        boardUpdate(row - 1, col - 1);
        if(checkWin()){
            cout << "Player " << player << " won" << endl;
            gameOver = true;
        }
        else if(checkDraw()){
            cout << "Draw!" << endl;
            gameOver = true;
        }
    }
    return 0;
}
```

Please click on the highlighted coding video link : Problem 4.4 (b)