1)

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

int main()

{

    ll n1 = 0b10111001;

    ll n2 = 0b10010110;

    cout << "Decimal = " << (n1 & n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 & n2) << "\n";

    n1 = 0b11011110;

    n2 = 0b11000101;

    cout << "Decimal = " << (n1 & n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 & n2) << "\n";

    n1 = 0b01111101;

    n2 = 0b10111110;

    cout << "Decimal = " << (n1 | n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 | n2) << "\n";

    n1 = 0b11000110;

    n2 = 0b11011100;

    cout << "Decimal = " << (n1 | n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 | n2) << "\n";

    n1 = 0b10111001;

    n2 = 0b11110110;

    cout << "Decimal = " << (n1 ^ n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 ^ n2) << "\n";

    n1 = 0b11000010;

    n2 = 0b00000101;

    cout << "Decimal = " << (n1 ^ n2) << "\n";

    cout << "Binary  = " << bitset<8>(n1 ^ n2) << "\n";

    n1 = 0b1011100110010110;

    cout << "Binary  = " << bitset<16>(~n1) << "\n";

    n1 = 11011110;

    n2 = 11000101;

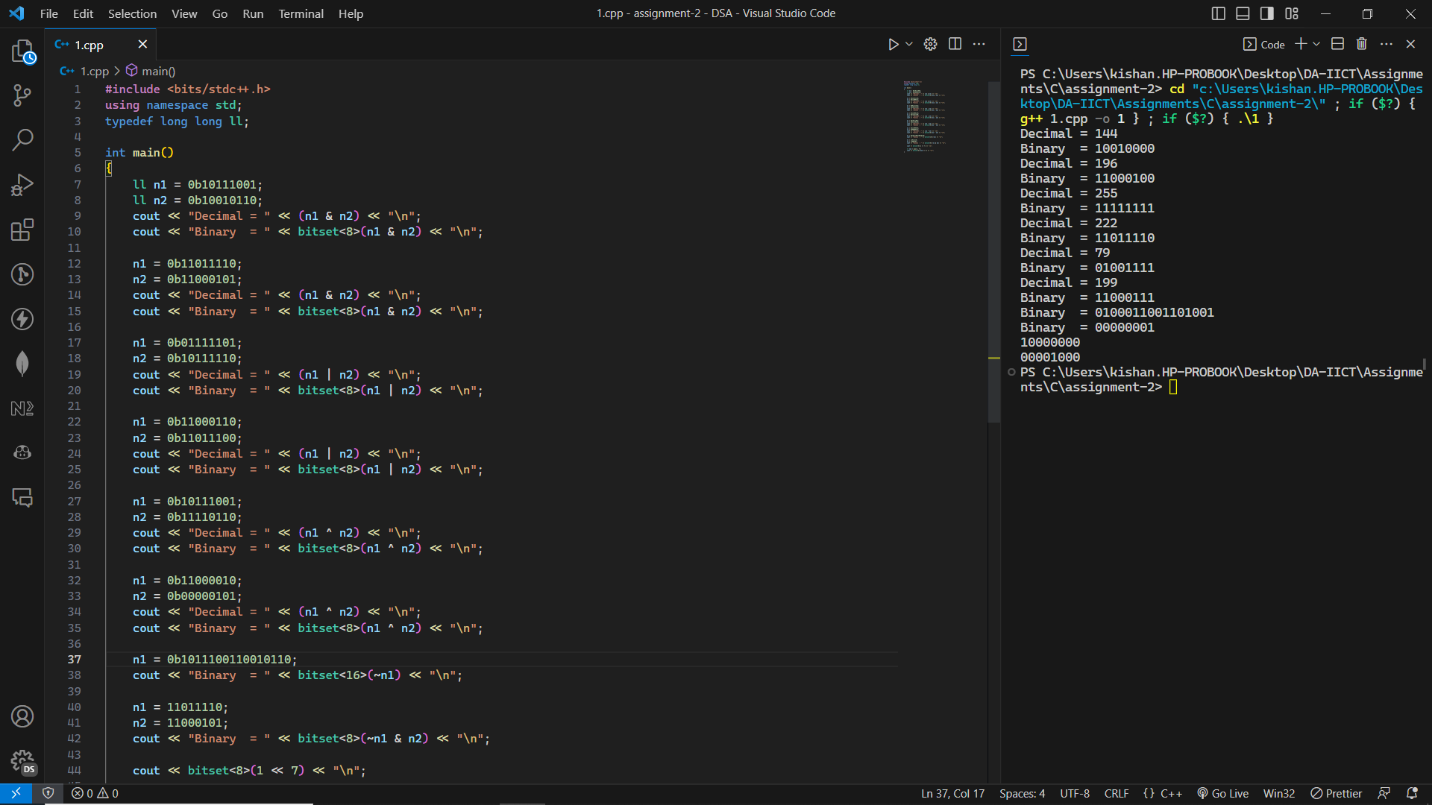
    cout << "Binary  = " << bitset<8>(~n1 & n2) << "\n";

    cout << bitset<8>(1 << 7) << "\n";

    ll num = pow(2, 7);

    cout << bitset<8>(num >> 4) << "\n";

}



2)

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

int main()

{

    ll bit = 0b10101110101010110101110100111011;

    cout << "1)\n";

    // set 16th bit

    bitset<32> result(bit | (1 << 15));

    cout << result << "\n";

    cout << "2)\n";

    // set 28th bit

    result = bitset<32>(bit & ~(1 << 27));

    cout << result << "\n";

    bitset<16> oddBit;

    bitset<16> evenBit;

    ll temp{bit};

    int i = 0;

    while (temp << 1 != 0)

    {

        oddBit[i++ / 2] = (temp & 1);

        temp = temp >> 1;

        evenBit[i++ / 2] = (temp & 1);

        temp = temp >> 1;

    }

    cout << "3)\n";

    cout << "Odd Bit  : " << oddBit << "\n";

    cout << "Even Bit : " << evenBit << "\n";

    cout << "4)\n";

    bitset<32> tempBit(bit);

    int n = 3; // 3rd nibble

    for (size\_t i = 4 \* (n - 1); i < 4 \* n; i++)

    {

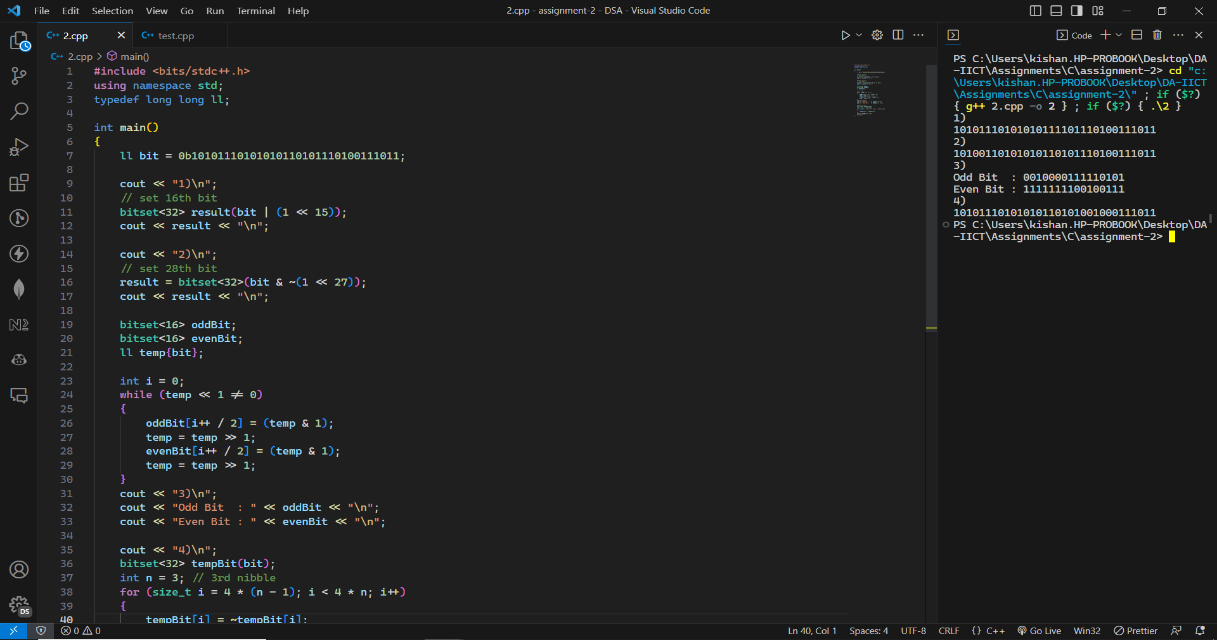
        tempBit[i] = ~tempBit[i];

    }

    cout << tempBit << "\n";

    return 0;

}



3)

#include <bits/stdc++.h>

using namespace std;

int main()

{

    int n, count{0};

    cin >> n;

    while (n != 0)

    {

        if (n & 1)

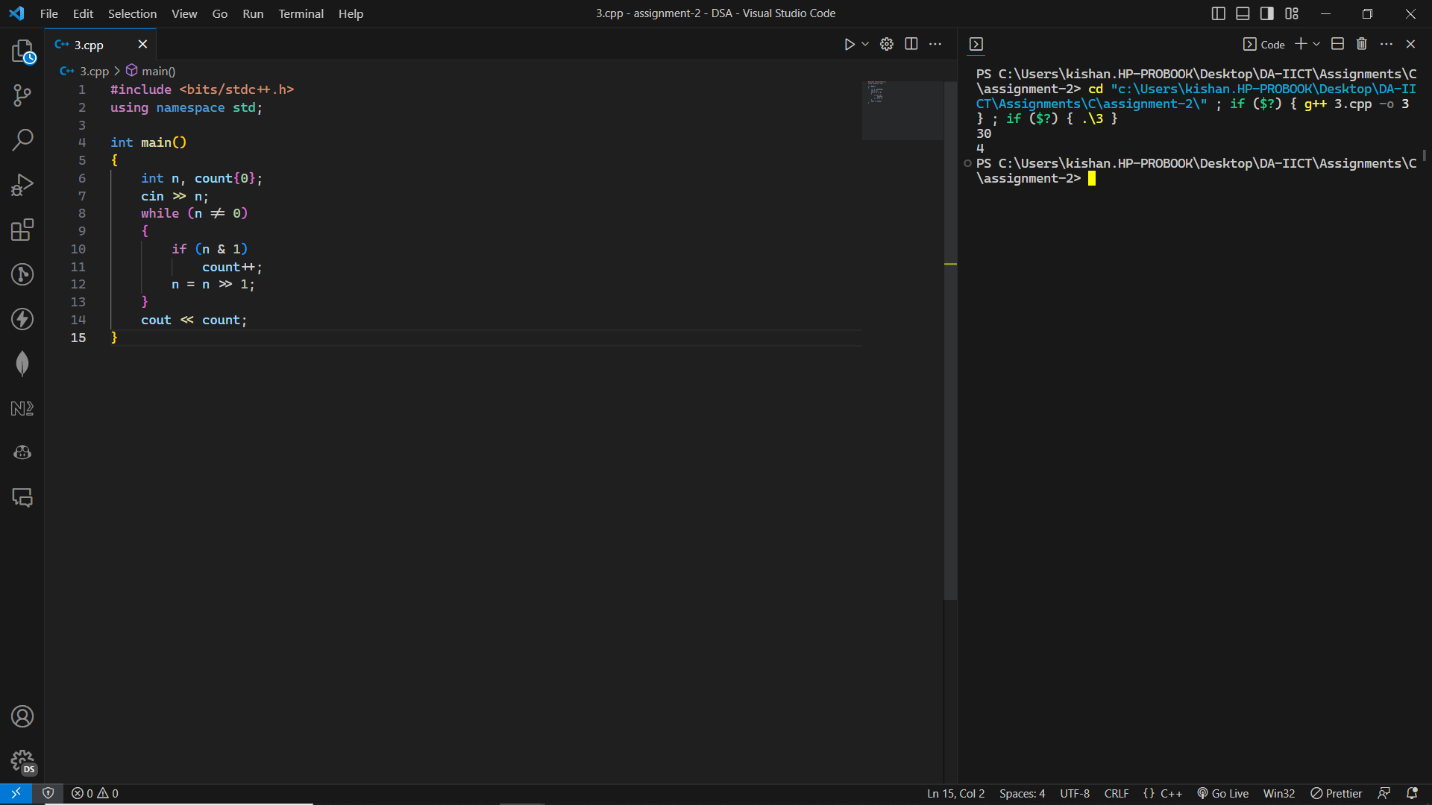
            count++;

        n = n >> 1;

    }

    cout << count;

}



4)

#include <bits/stdc++.h>

using namespace std;

// 0 - Tablet Mode

// 1 - WiFi

// 2 - Mute

// 3 - Airplane Mode

// 4 - Auto Hide Taskbar

const vector<string> settingNames = {

    "Tablet Mode",

    "WiFi",

    "Mute",

    "Airplane Mode",

    "Auto Hide Taskbar"};

void display(bitset<5> &settings)

{

    cout << "\n=================================\n";

    for (size\_t i = 0; i < settings.size(); i++)

    {

        if (settings[i])

        {

            cout << settingNames[i];

            cout << " is \*ON\*\n";

        }

        else

        {

            cout << settingNames[i];

            cout << " is OFF\n";

        }

    }

    cout << "=================================\n";

}

void takeInput(bitset<5> &settings)

{

    cout << "\n=>Select option to toggle<=\n";

    for (size\_t i = 0; i < settingNames.size(); i++)

    {

        cout << i + 1 << ") " << settingNames[i] << "\n";

    }

    int input;

    cin >> input;

    if (input >= 1 && input <= 5)

    {

        settings[input - 1] = ~settings[input - 1];

    }

    else

    {

        cout << "Invalid input\n";

    }

    display(settings);

}

int main()

{

    bitset<5> settings(00000);

    int input;

    bool isExist = false;

    display(settings);

    while (1)

    {

        if (isExist)

            break;

        cout << "Select one of the option...\n";

        cout << "1) Display the settings\n";

        cout << "2) Change the settings\n";

        cout << "3) EXIT\n";

        cin >> input;

        switch (input)

        {

        case 1:

            display(settings);

            break;

        case 2:

            takeInput(settings);

            break;

        case 3:

            isExist = true;

            break;

        default:

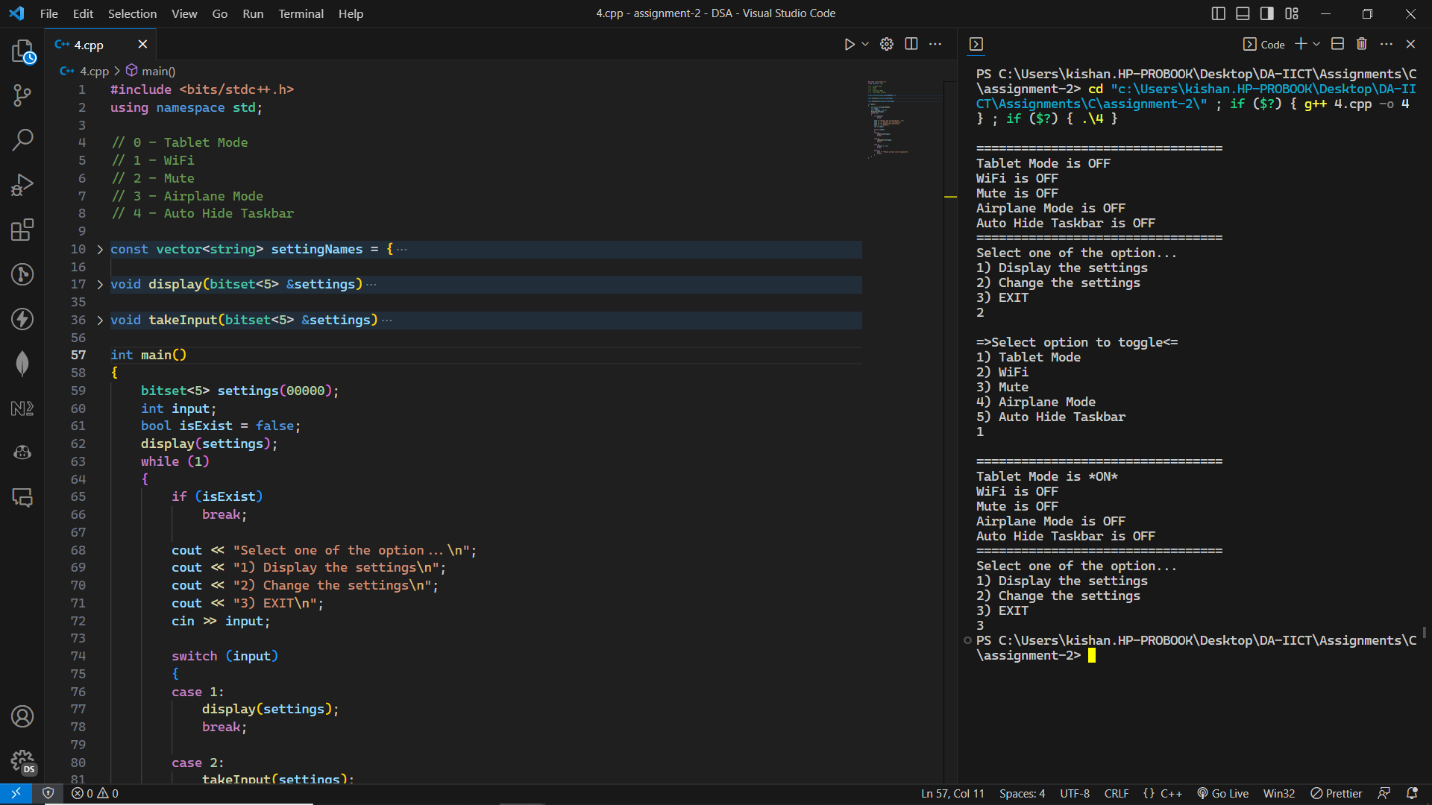
            cout << "Please provide valid response\n";

            break;

        }

    }

}



5)

#include <iostream>

using namespace std;

typedef unsigned int uint;

void print(int num)

{

    cout << "\nValues Selected : \n\n";

    int temp = 0;

    while (temp < 12)

    {

        if ((temp + 1) == 1)

        {

            if (num & 1)

                cout << "Seat Covers";

        }

        else if ((temp + 1) == 2)

        {

            if (num & 1)

                cout << " : Beige ";

        }

        else if ((temp + 1) == 3)

        {

            if (num & 1)

                cout << " : Dark ";

        }

        else if ((temp + 1) == 4)

        {

            if (num & 1)

                cout << " : Dual Pattern ";

        }

        else if ((temp + 1) == 5)

        {

            if (num & 1)

                cout << "\n"

                     << "Alloys";

        }

        else if ((temp + 1) == 6)

        {

            if (num & 1)

                cout << "\n"

                     << "Colors ";

        }

        else if ((temp + 1) == 7)

        {

            if (num & 1)

                cout << " : Coffee Brown ";

        }

        else if ((temp + 1) == 8)

        {

            if (num & 1)

                cout << " : Pearl White";

        }

        else if ((temp + 1) == 9)

        {

            if (num & 1)

                cout << " : Marine Blue ";

        }

        else if ((temp + 1) == 10)

        {

            if (num & 1)

                cout << " : Ash Grey ";

        }

        else if ((temp + 1) == 11)

        {

            if (num & 1)

                cout << "\n"

                     << "Steering Cover";

        }

        else if ((temp + 1) == 12)

        {

            if (num & 1)

                cout << "\n"

                     << "Body Cover"

                     << "\n";

        }

        num = num >> 1;

        temp++;

    }

    cout << "\n\n";

}

void CarCompany()

{

    unsigned int num{0b0}, bcv = {0b1};

    bool select = 0;

    int ch;

    cout << "\n--- Welcome To Car Company---\n\n";

    cout << "Select for \"Seat Cover\" (0 for no / 1 for yes): ";

    cin >> select;

    if (select == 1)

    {

        num = num | bcv;

        cout << "\nSelect for Seat cover :- \n";

        cout << " 1) Beige\n";

        cout << " 2) Dark\n";

        cout << " 3) Dual Pattern\n";

        cout << "Enter Choice : ";

        cin >> ch;

        switch (ch)

        {

        case 1:

            num = num | bcv << 1;

            break;

        case 2:

            num = num | bcv << 2;

            break;

        case 3:

            num = num | bcv << 3;

            break;

        default:

            break;

        }

    }

    cout << "\nDo you want \"Alloys\" (0 for no / 1 for yes): ";

    cin >> select;

    if (select == 1)

    {

        num = num | bcv << 4;

    }

    cout << "\nDo you want to Select for \"Colors\" (0 for no / 1 for yes): ";

    cin >> select;

    if (select == 1)

    {

        num = num | bcv << 5;

        cout << "\nSelect for Colors : \n";

        cout << " 1) Coffee Brown\n";

        cout << " 2) Pearl White\n";

        cout << " 3) Marine Blue\n";

        cout << " 4) Ash Grey\n";

        cout << "Enter Choice : ";

        cin >> ch;

        switch (ch)

        {

        case 1:

            num = num | bcv << 6;

            break;

        case 2:

            num = num | bcv << 7;

            break;

        case 3:

            num = num | bcv << 8;

            break;

        case 4:

            num = num | bcv << 9;

            break;

        default:

            break;

        }

    }

    cout << "\nDo you want \"Steering Cover\" (0 for no / 1 for yes): ";

    cin >> select;

    if (select == 1)

    {

        num = num | bcv << 10;

    }

    cout << "\nDo you want \"Body Cover\" (0 for no / 1 for yes): ";

    cin >> select;

    if (select == 1)

    {

        num = num | bcv << 11;

    }

    print(num);

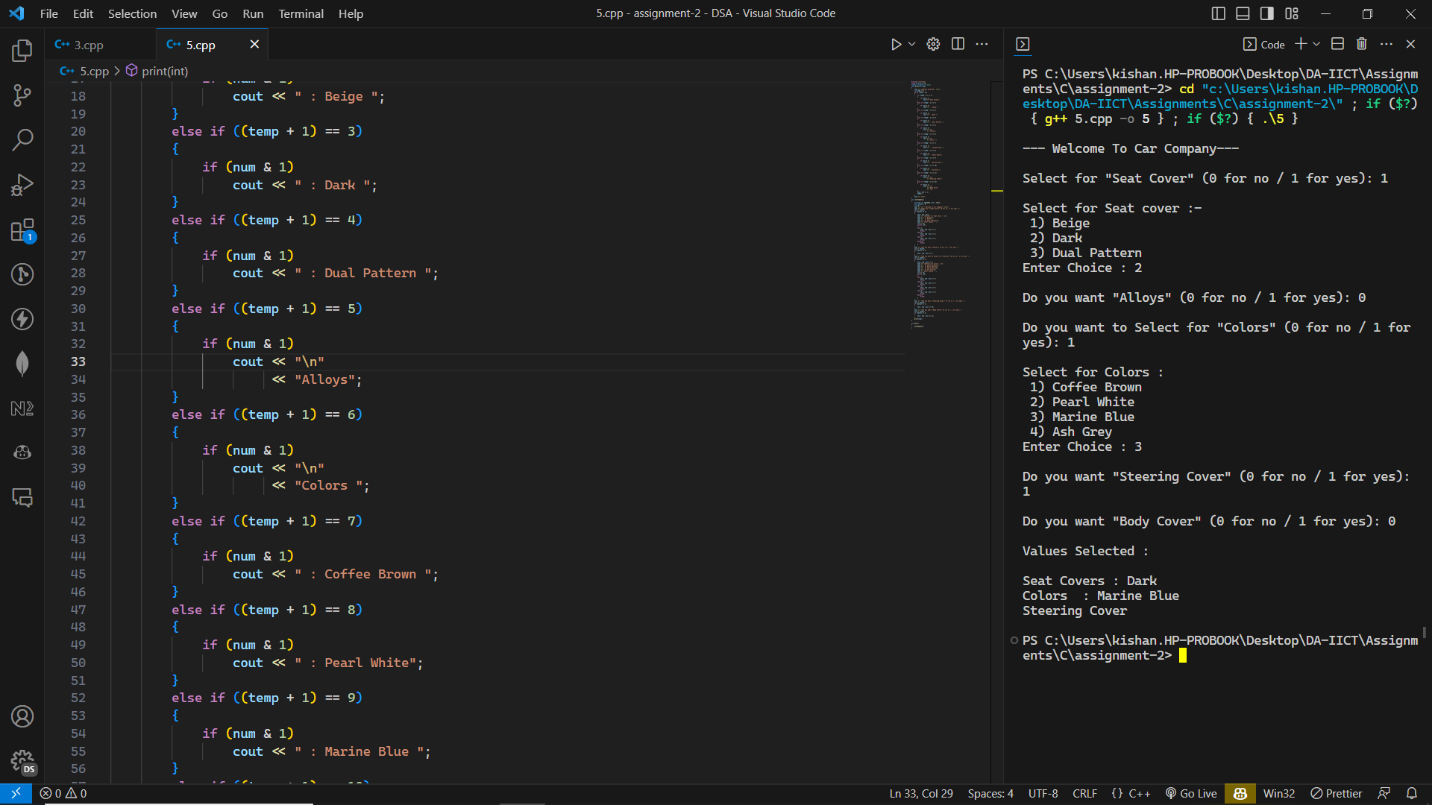
}

int main()

{

    CarCompany();

}



6)

#include <bits/stdc++.h>

using namespace std;

int main()

{

    double arr[5];

    double value = 1.1;

    for (size\_t i = 0; i < 5; i++)

    {

        arr[i] = value;

        // cout << value << "\n";

        value += 0.1;

    }

    for (size\_t i = 5; i > 0; i--)

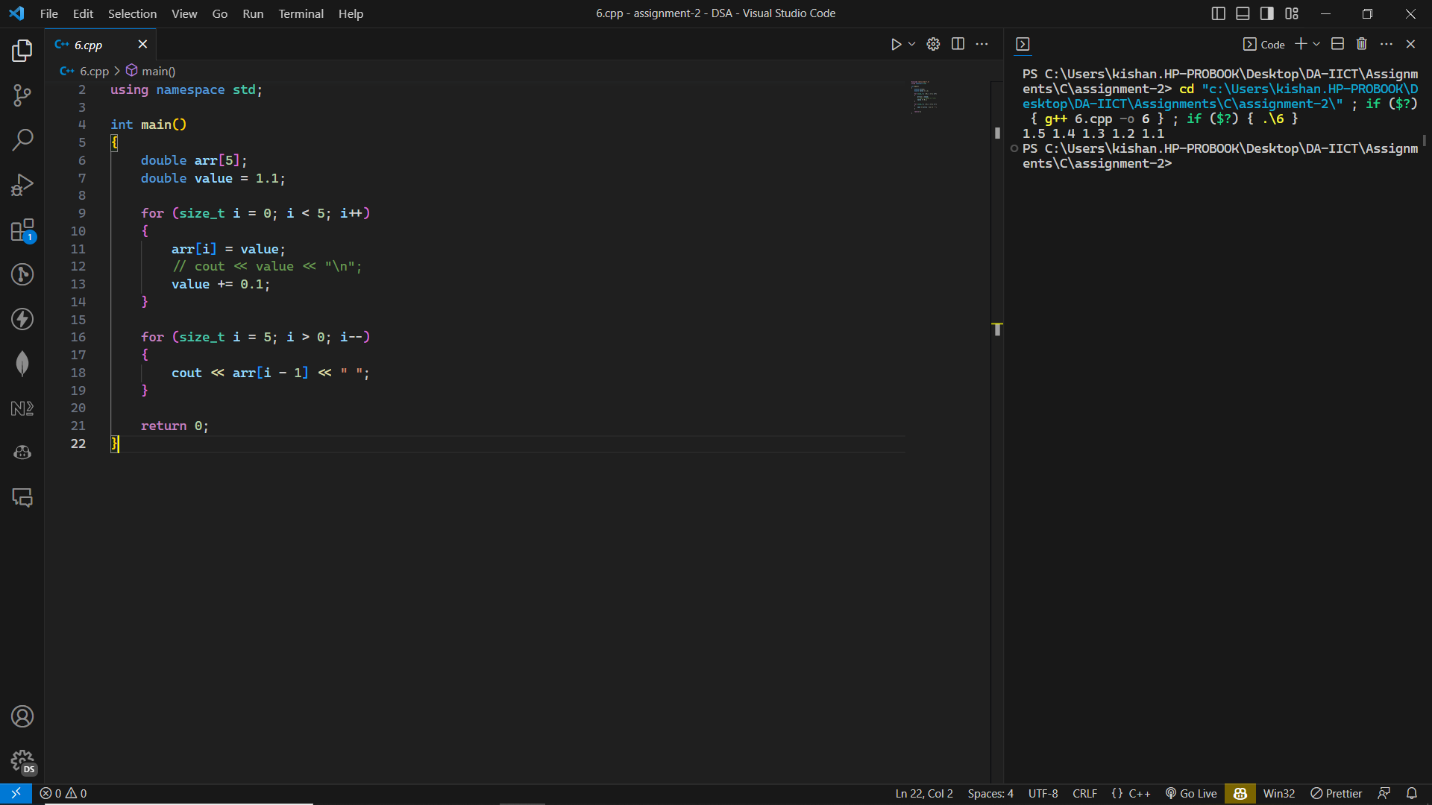
    {

        cout << arr[i - 1] << " ";

    }

    return 0;

}



7)

#include <iostream>

#include <cstring>

#include <string.h>

#include <cctype>

using namespace std;

int main()

{

    char chArrayA[100]{""}, chArrayB[100]{""};

    // a

    cin >> chArrayA >> chArrayB;

    cout << "a) \n";

    cout << "Firstname:- " << chArrayA << "\n"

         << "Lastname :- " << chArrayB << "\n";

    // b

    strcat(chArrayA, " ");

    strcat(chArrayA, chArrayB);

    cout << "b) \n";

    cout << chArrayA << "\n";

    // c

    cin.ignore();

    cin.getline(chArrayA, 90);

    cout << "c) \n";

    cout << chArrayA << "\n";

    // d

    int lastLocA = strlen(chArrayA) - 1;

    int lastLocB = 0;

    for (int i = lastLocA; i > 0; i--)

    {

        chArrayB[lastLocB++] = chArrayA[i];

    }

    cout << "d) \n";

    cout << chArrayB << "\n";

    // e

    char temp[100][100];

    int row{0}, col{0};

    temp[0][col++] = ' ';

    for (size\_t i = 0; i < strlen(chArrayA); i++)

    {

        if (chArrayA[i] == ' ')

        {

            row++;

            col = 0;

        }

        temp[row][col++] = chArrayA[i];

    }

    lastLocB = 0;

    for (size\_t I = row + 1; I > 0; I--)

    {

        int i = I - 1;

        for (size\_t j = 0; j < strlen(temp[i]); j++)

        {

            chArrayB[lastLocB++] = temp[i][j];

        }

    }

    cout << "e) \n";

    cout << chArrayB << "\n";

    // f

    cin.getline(chArrayA, 100);

    lastLocB = 0;

    for (int i = 0; i < strlen(chArrayA); ++i)

    {

        if (std::isalpha(chArrayA[i]))

        {

            chArrayB[lastLocB++] = chArrayA[i];

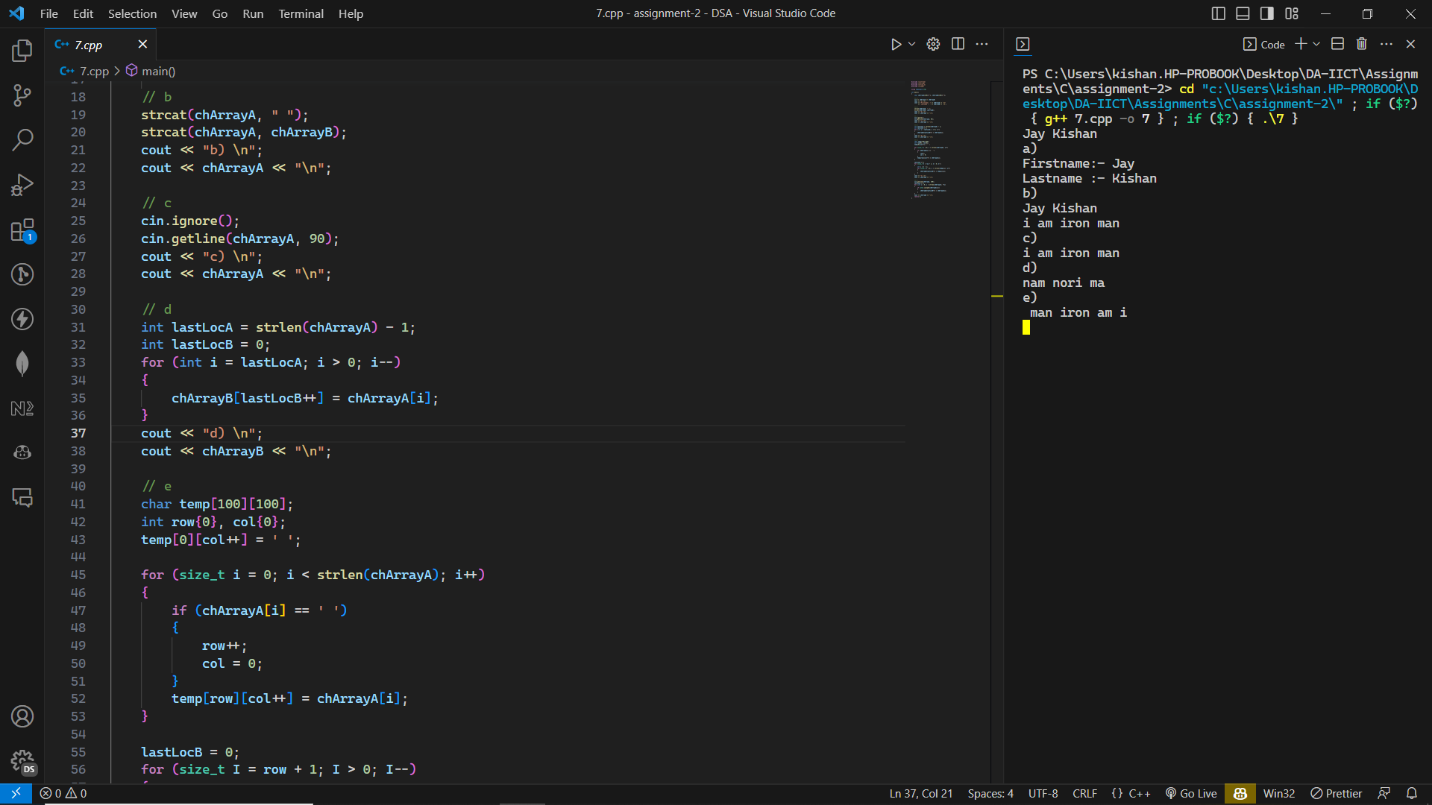
        }

    }

    cout << chArrayB << "\n";

    return 0;

}



8)

#include <iostream>

#include <bitset>

using namespace std;

int seats[16];

const int totalRow = 16;

const int totalSeatInRow = 20;

const char seatBooked{'X'};

const char seatAvailale{'.'};

void setInitialData()

{

    for (size\_t i = 0; i < totalRow; i++)

    {

        seats[i] = 0;

    }

}

void display()

{

    char rowName = 'A';

    for (size\_t i = 0; i < totalRow; i++)

    {

        bitset<20> row(seats[i]);

        for (size\_t i = 0; i < row.size(); i++)

        {

            if (row[i] == 1)

            {

                cout << seatBooked;

            }

            else

            {

                cout << seatAvailale;

            }

        }

        cout << "   <=== " << rowName++;

        cout << "\n";

    }

    cout << "VVVVVVVVVVVVVVVVVVVV   Screeen this way\n";

}

void buyTicket()

{

    char rowName{};

    int row{0}, col{0};

    cout << "Enter row:- ";

    cin >> rowName;

    cout << "Enter seat number:- ";

    cin >> col;

    row = (rowName >= 'A' && rowName <= 'P') ? (rowName - 'A') : ((rowName >= 'a' && rowName <= 'p') ? (rowName - 'a') : -1);

    if (row != -1)

    {

        if (col > 0 && col <= totalSeatInRow)

        {

            bitset<20> temp = bitset<20>(seats[row]);

            if (temp[col - 1] == 0)

            {

                temp[col - 1] = 1;

                seats[row] = static\_cast<int>(temp.to\_ulong());

                cout << rowName << "-" << col << " has been booked successfully.\n";

            }

            else

            {

                cout << "Sorry, this seat is already booked.\n";

            }

        }

        else

        {

            cout << "Seat number you entered is not valid.\n";

        }

    }

    else

    {

        cout << "Please enter vaild seat name.\n";

    }

}

void cancelTicket()

{

    char rowName{};

    int row{0}, col{0};

    cout << "Enter row:- ";

    cin >> rowName;

    cout << "Enter seat number:- ";

    cin >> col;

    row = (rowName >= 'A' && rowName <= 'P') ? (rowName - 'A') : ((rowName >= 'a' && rowName <= 'p') ? (rowName - 'a') : -1);

    if (row != -1)

    {

        if (col > 0 && col <= totalSeatInRow)

        {

            bitset<20> temp = bitset<20>(seats[row]);

            if (temp[col - 1] == 1)

            {

                temp[col - 1] = 0;

                seats[row] = static\_cast<int>(temp.to\_ulong());

                cout << "Your ticket " << rowName << "-" << col << " has been cancelled successfully.\n";

            }

            else

            {

                cout << "Please recheck again, this seat is not booked.\n";

            }

        }

        else

        {

            cout << "Seat number you entered is not valid.\n";

        }

    }

    else

    {

        cout << "Please enter vaild row name.\n";

    }

}

int main()

{

    int choice{0};

    display();

    bool exit = false;

    while (1)

    {

        if (exit)

            break;

        cout << "1) Buy a ticket \n2) Cancel a ticket \n3) Exit";

        cin >> choice;

        switch (choice)

        {

        case 1:

            cout << "===============================\n";

            buyTicket();

            cout << "===============================\n";

            display();

            break;

        case 2:

            cout << "===============================\n";

            cancelTicket();

            cout << "===============================\n";

            display();

            break;

        case 3:

            exit = true;

            break;

        default:

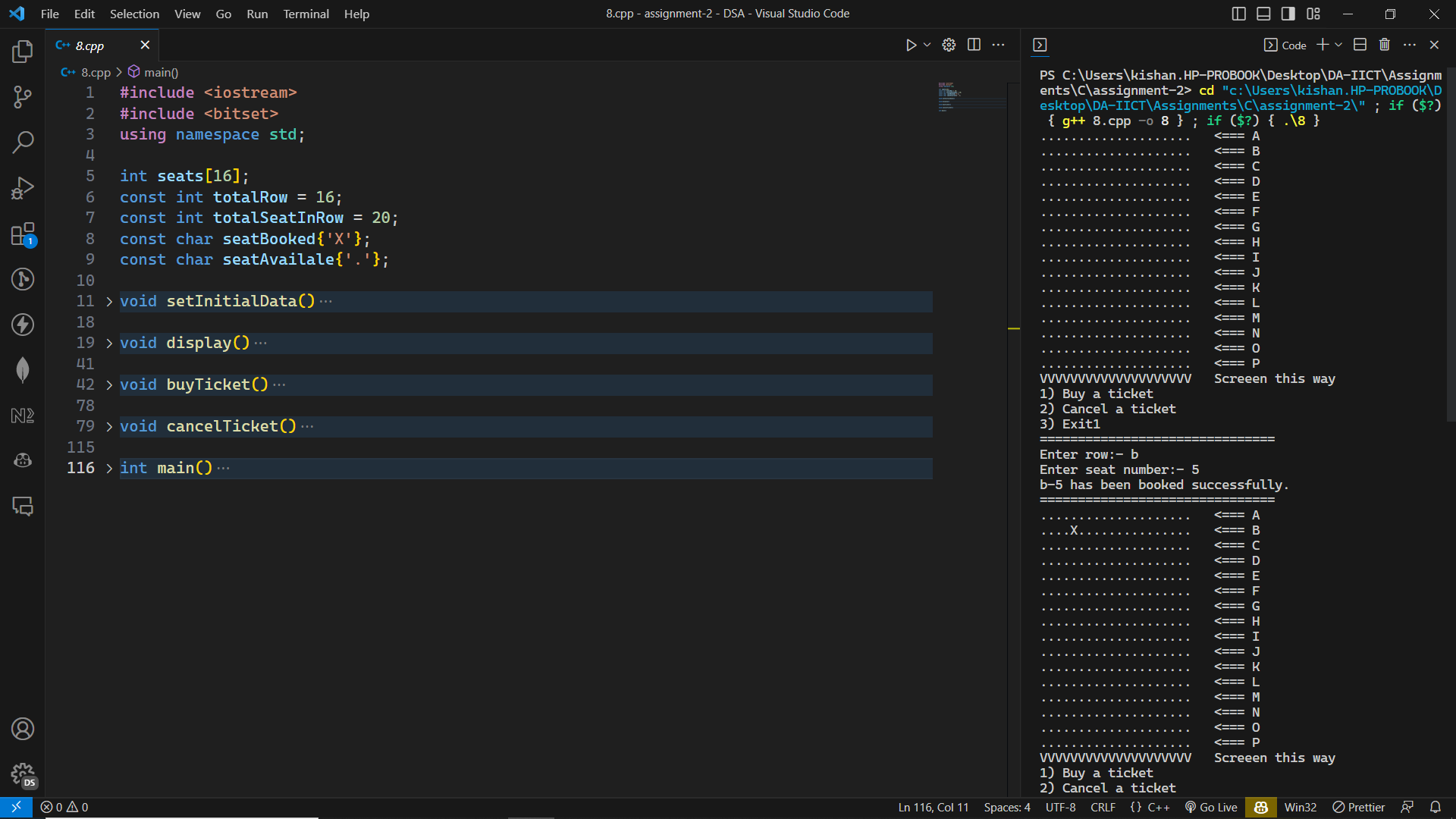
            cout << "Please enter valid choice\n";

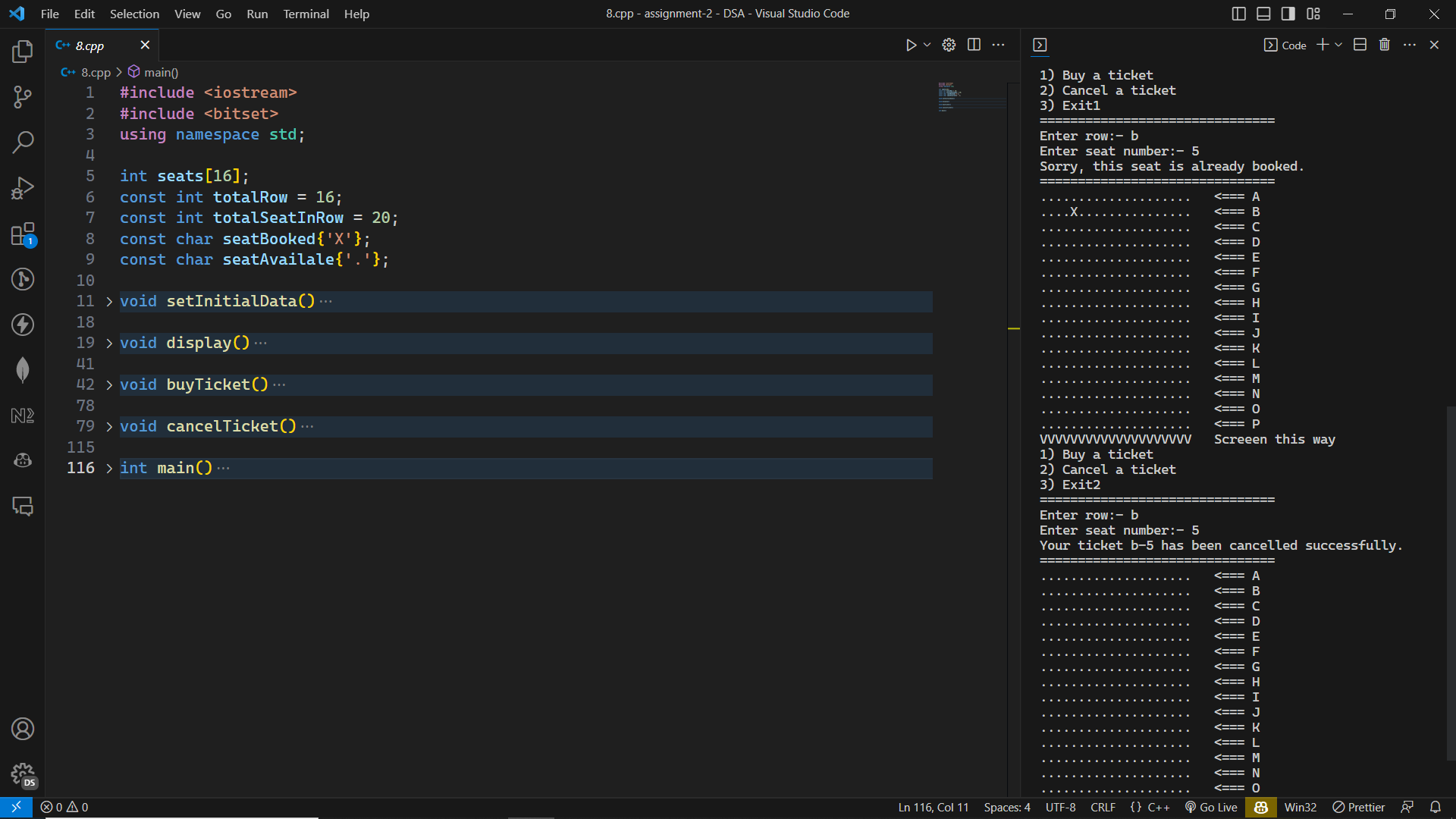
            break;

        }

    }

}





9)

#include <bits/stdc++.h>

using namespace std;

vector<int> pendingOrders;

void placeOrder()

{

    int orderNumber = 1 + pendingOrders.size();

    pendingOrders.push\_back(orderNumber);

    cout << "=================================\n";

    cout << "Your order is placed. Order Number:- " << orderNumber << "\n";

    cout << "=================================\n";

    cin.ignore();

}

void displayPendingOrders()

{

    cout << "=================================\n";

    cout << "Pending orders:- "

         << "\n";

    for (int orderNumber : pendingOrders)

    {

        cout << "=> Order Number: " << orderNumber << "\n";

    }

    cout << "Total pending orders:- " << pendingOrders.size() << "\n";

    cout << "=================================\n";

}

void serveOrder()

{

    cout << "=================================\n";

    cout << "Pending orders:\n";

    for (int orderNumber : pendingOrders)

    {

        cout << "=> Order Number: " << orderNumber << "\n";

    }

    cout << "\nPick order: ";

    int pickedOrder;

    cin >> pickedOrder;

    auto it = find(pendingOrders.begin(), pendingOrders.end(), pickedOrder);

    if (it != pendingOrders.end())

    {

        pendingOrders.erase(it);

        cout << "Order " << pickedOrder << " has been served.\n";

    }

    else

    {

        cout << "Invalid order number.\n";

    }

    cin.ignore();

    cout << "=================================\n";

}

int main()

{

    int choice{0};

    bool exit = false;

    while (1)

    {

        if (exit)

            break;

        cout << "1) Place order\n2) Pending orders\n3) Serve order\n4) Exit\nEnter your pick: ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            placeOrder();

            break;

        case 2:

            displayPendingOrders();

            break;

        case 3:

            serveOrder();

            break;

        case 4:

            exit = true;

            break;

        default:

            cout << "Please enter valid choice."

                 << "\n";

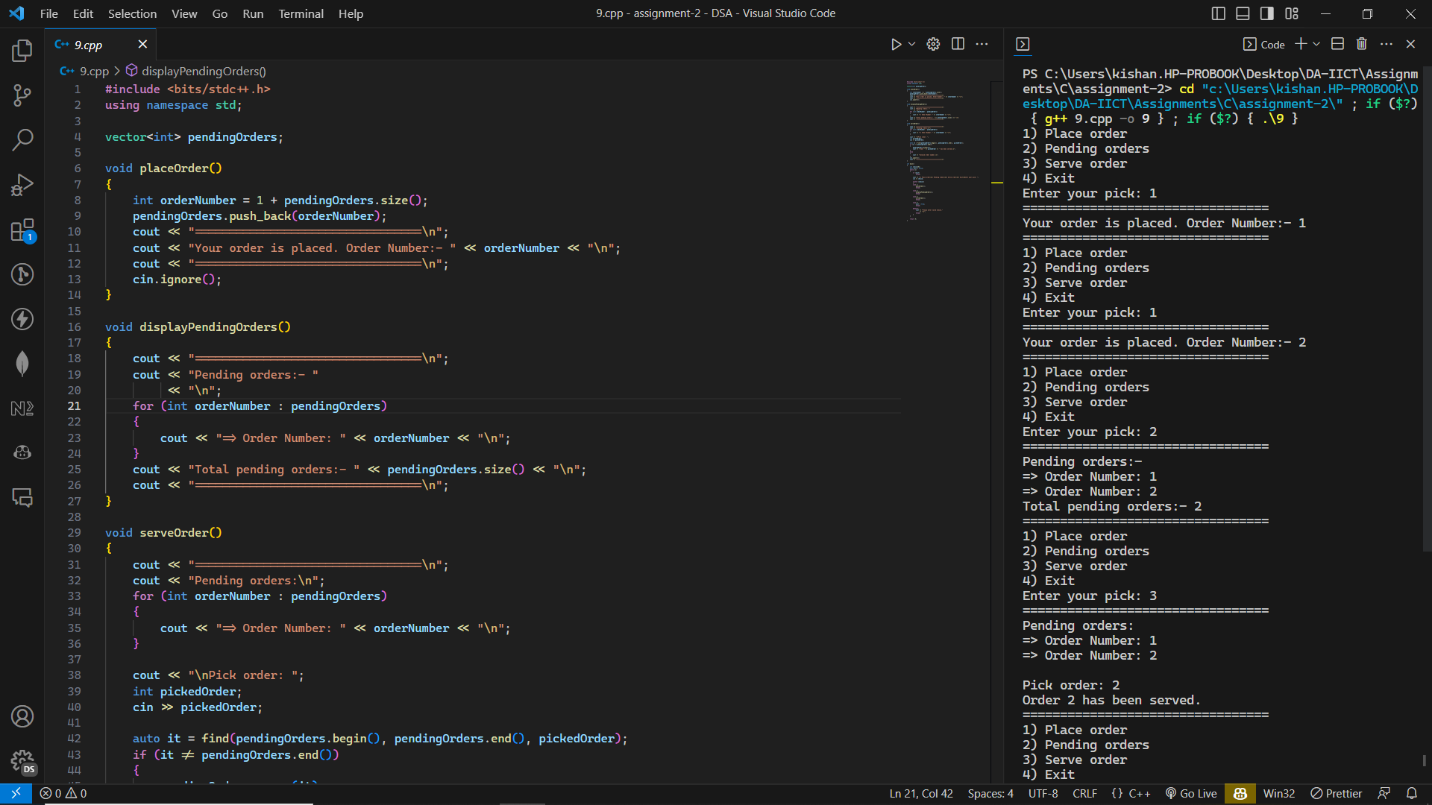
            break;

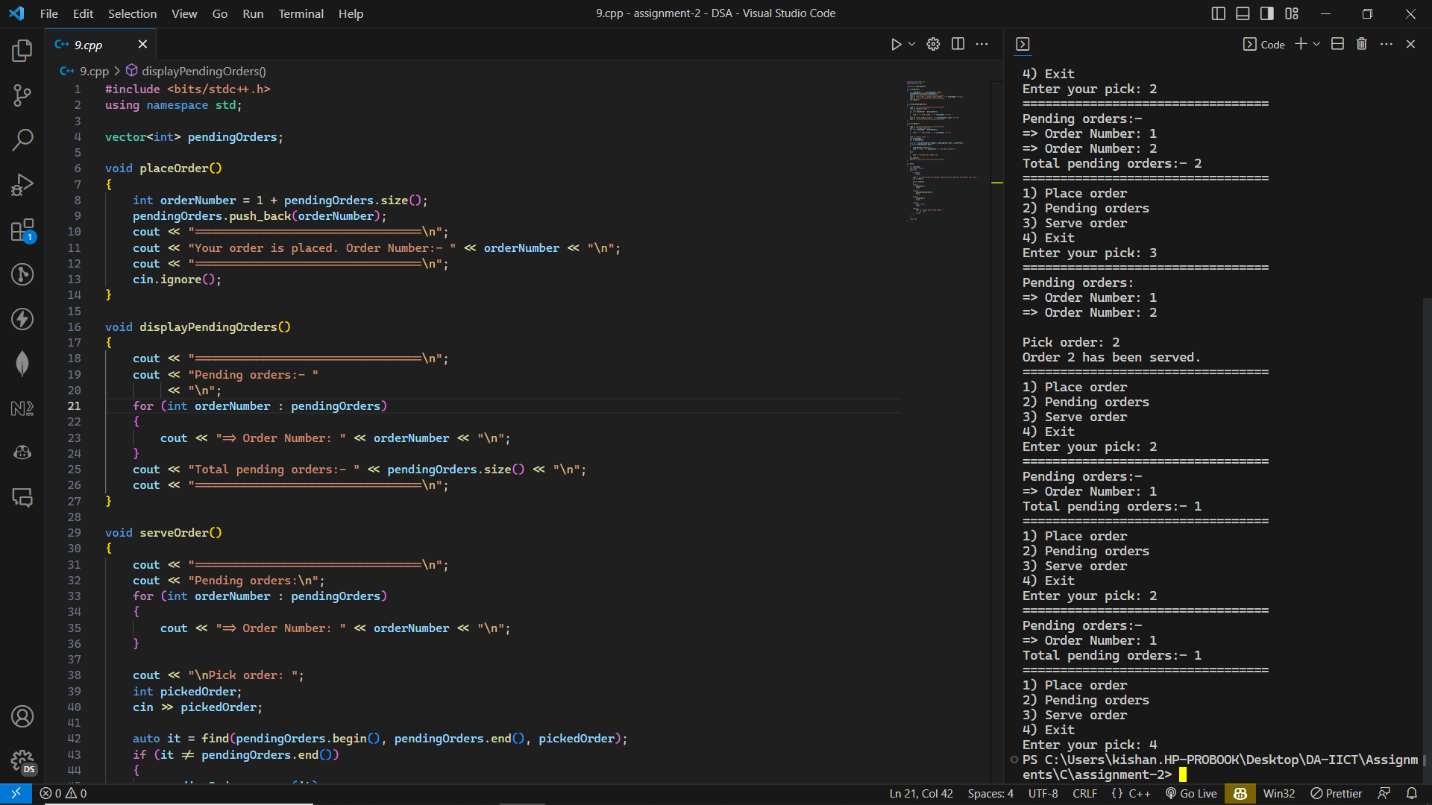
        }

    }

    return 0;

}





10)

#include <bits/stdc++.h>

using namespace std;

#define MAX 1000

int main()

{

    int row, col;

    cin >> row >> col;

    char arr[MAX][MAX] = {0};

    bool flg = 0;

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            arr[i][j] = '\*';

        }

    }

    int round = 0;

    row--, col--;

    while (1)

    {

        int col\_limit = col - (round + 1), row\_limit = row - (round + 1);

        int x1 = round, y1 = round, x2 = round, y2 = col - round, x3 = row - round, y3 = round, x4 = row - round, y4 = col - round;

        if (y1 > y2 || y3 > y4 || x1 > x3 || x2 > x4)

            break;

        while (y1 <= col\_limit)

            arr[x1][y1++] = arr[x4][y4--] = '-';

        while (x2 <= row\_limit)

            arr[x2++][y2] = arr[x3--][y3] = '-';

        round += 2;

    }

    if (row == col && (row % 4) == 0)

    {

        arr[row / 2][col / 2] = '-';

    }

    for (int i = 0; i < row + 1; i++)

    {

        for (int j = 0; j < col + 1; j++)

        {

            cout << arr[i][j] << " ";

        }

        cout << endl;

    }

}

