

Name :-Jeevan Rajpurohit

(Assignment-2)

Student ID:-202312090

(IT603)-Programming

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";

}

```

The screenshot shows a Visual Studio Code editor with a C++ file named 2.cpp. The code implements several bitset operations as shown in the previous block. The output window on the right displays the results of these operations.

```

PS C:\Users\ADMIN> cd "C:\Users\ADMIN\Down
loads\"; if ($?) { g++ 2.cpp -o 2 }; if
($?) { . 2 }
Decimal = 144
Binary = 10010000
Decimal = 196
Binary = 11000100
Decimal = 255
Binary = 11111111
Decimal = 222
Binary = 11011110
Decimal = 79
Binary = 01001111
Decimal = 199
Binary = 11000111
Binary = 010001100101001
Binary = 00000001
10000000
00001000
PS C:\Users\ADMIN\Downloads>

```

2).

//2nd question

```
#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;

}

```

The screenshot shows the Visual Studio Code editor with a C++ file named 2.cpp. The code implements a program that takes a 32-bit integer 'bit' (0b10101110101010101010101010101011) and processes it. It sets the 16th and 28th bits, then iterates through the 32 bits, toggling every 2nd bit (odd-indexed bits). The final output shows the original bit string, the modified bit string, and the 4th nibble (bits 12-15) as 1010.

```

1 //2nd question
2 #include <bits/stdc++.h>
3 using namespace std;
4 typedef long long ll;
5 int main()
6 {
7     ll bit = 0b10101110101010101010101010101011;
8     cout << "1)\n";
9     // set 16th bit
10    bitset<32> result(bit | (1 << 15));
11    cout << result << "\n";
12    cout << "2)\n";
13    // set 28th bit
14    result = bitset<32>(bit & ~(1 << 27));
15
16    cout << result << "\n";
17    bitset<16> oddBit;
18    bitset<16> evenBit;
19    ll temp(bit);
20    int i = 0;
21    while (temp << 1 != 0)
22    {
23        oddBit[i++ / 2] = (temp & 1);
24        temp = temp >> 1;
25        evenBit[i++ / 2] = (temp & 1);
26        temp = temp >> 1;
27    }
28    cout << "3)\n";
29    cout << "Odd Bit : " << oddBit << "\n";
30    cout << "Even Bit : " << evenBit << "\n";
31    cout << "4)\n";
32    bitset<32> tempBit(bit);
33    int n = 3; // 3rd nibble

```

Terminal Output:

```

PS C:\Users\ADMIN> cd "C:\Users\ADMIN\Downloads\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }
1)
101011101010101011101110100111011
2)
101001101010101010101110100111011
3)
Odd Bit : 0010000111110101
Even Bit : 1111111100100111
4)
101011101010101010101001000111011
PS C:\Users\ADMIN\Downloads>

```

3).

//4thrd question

```
#include <iostream>
```

```
using namespace std;
```

```

int main(){

    int num;

    cin>>num;

    int count{0};

    while(num!=0){

        if(num & 1)

            count++;

            num=num>>1;

    }

    cout<<"Number of bit set : " <<count;

}

```

The screenshot shows the Visual Studio Code editor with a C++ file named 2.cpp. The code is as follows:

```

1 //4thrd question
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6     int num;
7     cin>>num;
8     int count{0};
9
10    while(num!=0){
11        if(num & 1)
12            count++;
13            num=num>>1;
14    }
15    cout<<"Number of bit set : " <<count;
16 }
17
18
19

```

The output window on the right shows the execution results:

```

PS C:\Users\ADMIN\Downloads> cd "C:\Users\ADMIN\Downloads\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }
30
Number of bit set : 4
PS C:\Users\ADMIN\Downloads>

```

The status bar at the bottom indicates the current line and column (Ln 11, Col 20), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the language (C++), the architecture (Win32), and the system status (31°C, Haze, 7:28 PM, 17-Aug-23).

6).

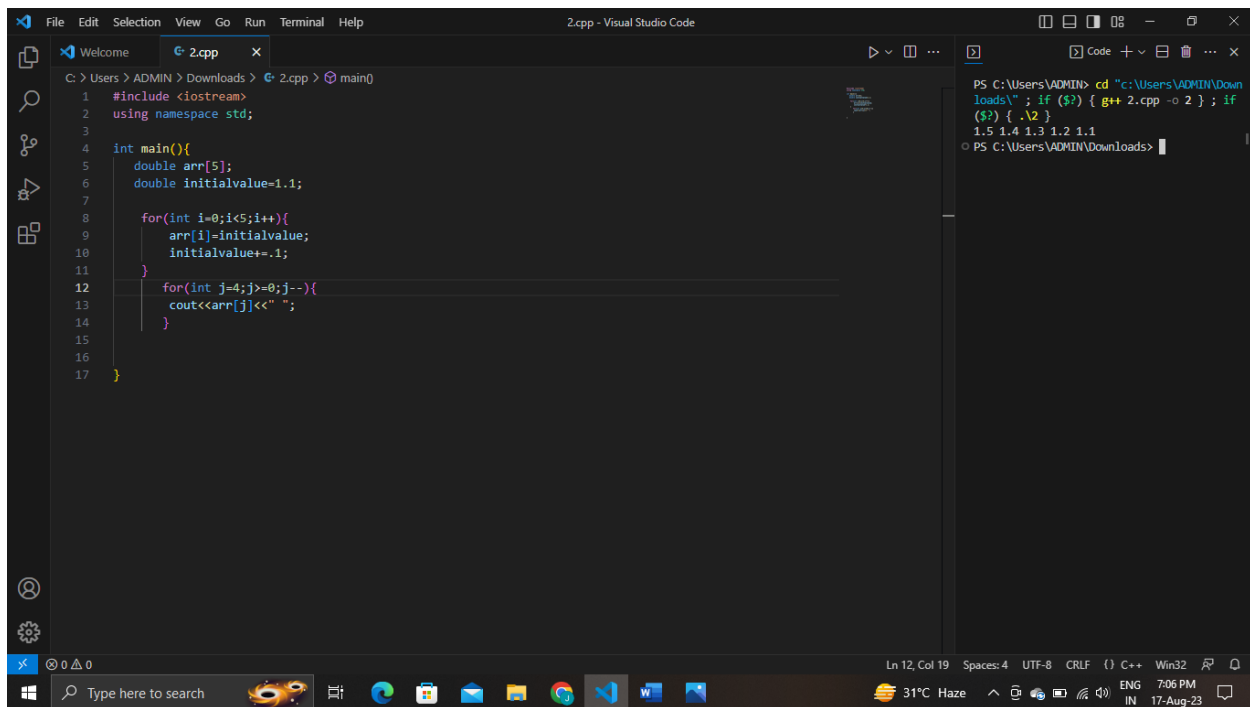
```
#include <iostream>

using namespace std;

int main(){
    double arr[5];
    double initialvalue=1.1;

    for(int i=0;i<5;i++){
        arr[i]=initialvalue;
        initialvalue+=.1;
    }
    for(int j=4;j>=0;j--){
        cout<<arr[j]<<" ";
    }

}
```



4).

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
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);

```



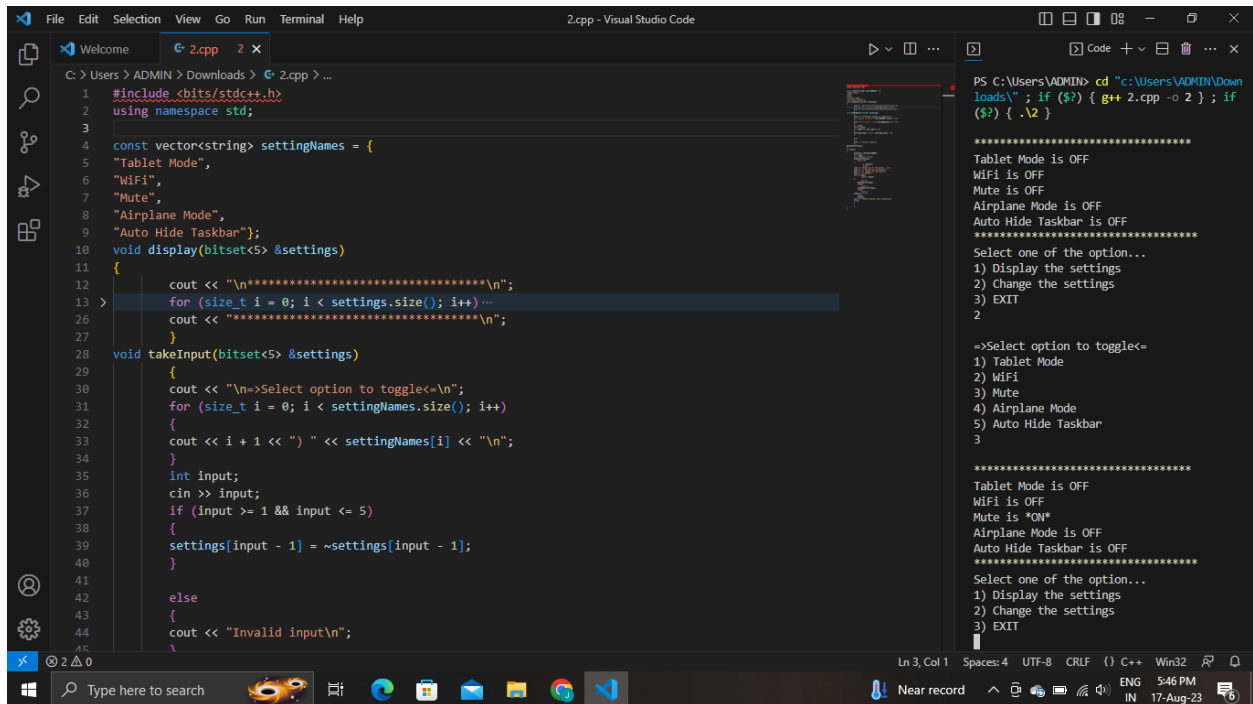
```
}  
  
int main()  
{  
    bitset<5> settings(00000);  
    int input;  
    bool isExist = false;  
    display(settings);  
    while (1)  
    {  
        if (isExist)  
            break;  
        cout << "Select one of the option...\n";  
        cout << "1) Display the settings\n";  
        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;
```

```
}
```

```
}
```

```
}
```



```
File Edit Selection View Go Run Terminal Help
2.cpp - Visual Studio Code

C:\Users\ADMIN\Downloads> 2.cpp > ...

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const vector<string> settingNames = {
5     "Tablet Mode",
6     "Wifi",
7     "Mute",
8     "Airplane Mode",
9     "Auto Hide Taskbar"};
10 void display(bitset<5> &settings)
11 {
12     cout << "\n*****\n";
13     for (size_t i = 0; i < settings.size(); i++) ...
14     cout << "*****\n";
15 }
16 void takeInput(bitset<5> &settings)
17 {
18     cout << "\n=>Select option to toggle<-\n";
19     for (size_t i = 0; i < settingNames.size(); i++)
20     {
21         cout << i + 1 << " " << settingNames[i] << "\n";
22     }
23     int input;
24     cin >> input;
25     if (input >= 1 && input <= 5)
26     {
27         settings[input - 1] = ~settings[input - 1];
28     }
29     else
30     {
31         cout << "Invalid input\n";
32     }
33 }
```

```
PS C:\Users\ADMIN> cd "C:\Users\ADMIN\Down
loads\" ; if ($?) { g++ 2.cpp -o 2 } ; if
($?) { .\2 }
```

```
*****
Tablet Mode is OFF
Wifi is OFF
Mute is OFF
Airplane Mode is OFF
Auto Hide Taskbar is OFF
*****
Select one of the option...
1) Display the settings
2) Change the settings
3) EXIT
2

=>Select option to toggle<-
1) Tablet Mode
2) Wifi
3) Mute
4) Airplane Mode
5) Auto Hide Taskbar
3

*****
Tablet Mode is OFF
Wifi is OFF
Mute is *ON*
Airplane Mode is OFF
Auto Hide Taskbar is OFF
*****
Select one of the option...
1) Display the settings
2) Change the settings
3) EXIT
```

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();
}

```

The screenshot shows the Visual Studio Code editor with a C++ file named 2.cpp. The code implements a car configuration system with options for Alloys, Colors, Steering Cover, and Body Cover. The output window on the right shows the program's execution, where the user has selected 'Dark' for Alloys, 'Pearl White' for Colors, '0' for Steering Cover, and '1' for Body Cover. The final output shows the selected values: 'Seat Covers : Dark', 'Alloys : Pearl White', and 'Body Cover'.

```

C:\Users\ADMIN> Downloads > 2.cpp > CarCompany()

111 break;
112 }
113 }
114 cout << "\nDo you want \"Alloys\" (0 for no / 1 for yes): ";
115 cin >> select;
116
117 if (select == 1)
118 {
119     num = num | bcv << 4;
120 }
121 cout << "\nDo you want to Select for \"Colors\" (0 for no / 1 for yes):";
122 cin >> select;
123 if (select == 1)
124 {
125     num = num | bcv << 5;
126     cout << "\nSelect for Colors : \n";
127     cout << " 1) Coffee Brown\n";
128     cout << " 2) Pearl White\n";
129     cout << " 3) Marine Blue\n";
130     cout << " 4) Ash Grey\n";
131     cout << "Enter Choice : ";
132 cin >> ch;
133 switch (ch)
134 {
135     case 1:
136         num = num | bcv << 6;
137         break;
138     case 2:
139         num = num | bcv << 7;
140         break;
141     case 3:
142         num = num | bcv << 8;

```

Output:

```

Select for "Seat Cover" (0 for no / 1 for yes): 1

Select for Seat cover :-
1) Beige
2) Dark
3) Dual Pattern
Enter Choice : 2

Do you want "Alloys" (0 for no / 1 for yes ): 1

Do you want to Select for "Colors" (0 for no / 1 for yes):1

Select for Colors :
1) Coffee Brown
2) Pearl White
3) Marine Blue
4) Ash Grey
Enter Choice : 2

Do you want "Steering Cover" (0 for no / 1 for yes): 0

Do you want "Body Cover" (0 for no / 1 for yes): 1

Values Selected :

Seat Covers : Dark
Alloys
Colors : Pearl White
Body Cover

```

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 l = row + 1; l > 0; l--)
{
    int i = l - 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;

}

```

The screenshot shows the Visual Studio Code editor with a C++ file named 2.cpp. The code implements a function to concatenate two strings into a new array. The output window shows the results of running the program.

```

C:\Users\ADMIN> cd "C:\Users\ADMIN\Downloads" & g++ 2.cpp -o 2 & if ($?) { .\2 }
jalhe jeevan
a)
Firstname:-  jalhe
Lastname :-  jeevan
b)
jalhe jeevan
i am super man
c)
i am super man
d)
nam repus ma
e)
man super am i

```

8).

```

#include <iostream>

using namespace std;

const size_t ROWS{ 16 };
const size_t COLS{ 10 };

const char symBooked{ 'X' };
const char symAvail{ '.' };

const char initRowName{ 'A' };

```

```
bool seats[ROWS][COLS];
```

```
char rowNames[ROWS];
```

```
char inRow{ NULL };
```

```
int inCol{ 0 };
```

```
void Init()
```

```
{
```

```
    for (auto i{ 0 }; i < ROWS; i++)
```

```
    {
```

```
        for (auto j{ 0 }; j < COLS; j++)
```

```
            seats[i][j] = false;
```

```
        rowNames[i] = initRowName + i;
```

```
    }
```

```
}
```

```
void Display()
```

```
{
```

```
    //system("cls");
```

```
    cout << "Current status:" << endl;
```

```
    for (auto i{ 0 }; i < ROWS; i++)
```

```
    {
```

```
        for (auto j{ 0 }; j < COLS; j++)
```

```
            cout << (seats[i][j] ? symBooked : symAvail) << ' ';
```

```
        cout << "--> " << rowNames[i];
```

```
        cout << endl;
```

```
    }
```

```
    cout << endl;
```

```

    cout << "vvvvvvvvvvvvvvvvvvvv ---- Screen this way" << endl;

}

void TakeInput()
{
    std::cin >> inRow >> inCol;
}

void ProcessInput(bool n)
{
    int rowIndex{ static_cast<int>(inRow - initRowName) };
    int colIndex{ inCol - 1 };
    if (n) {
        seats[rowIndex][colIndex] = true;
    }
    else {
        seats[rowIndex][colIndex] = false;
    }
}

int main()
{
    bool isExist = true;

    Init();
    Display();

    while (isExist) {

```

```
int ch;

cout << "\nSelect option :- \n";
cout << "1. Buy a ticket\n";
cout << "2. Cancel ticket\n";
cout << "3. Exit the app\n";
cout << "Enter Choice : ";
cin >> ch;
```

```
switch (ch)
{
case 1:
    TakeInput();
    if (inRow == 'Q' || inRow == 'q')
        break;
    ProcessInput(true);
    Display();
    break;
case 2:
    TakeInput();
    if (inRow == 'Q' || inRow == 'q')
        break;
    ProcessInput(false);
    Display();
    break;
case 3:
    isExist=false;
    cout << "Thank You For Visit...";
    break;
```

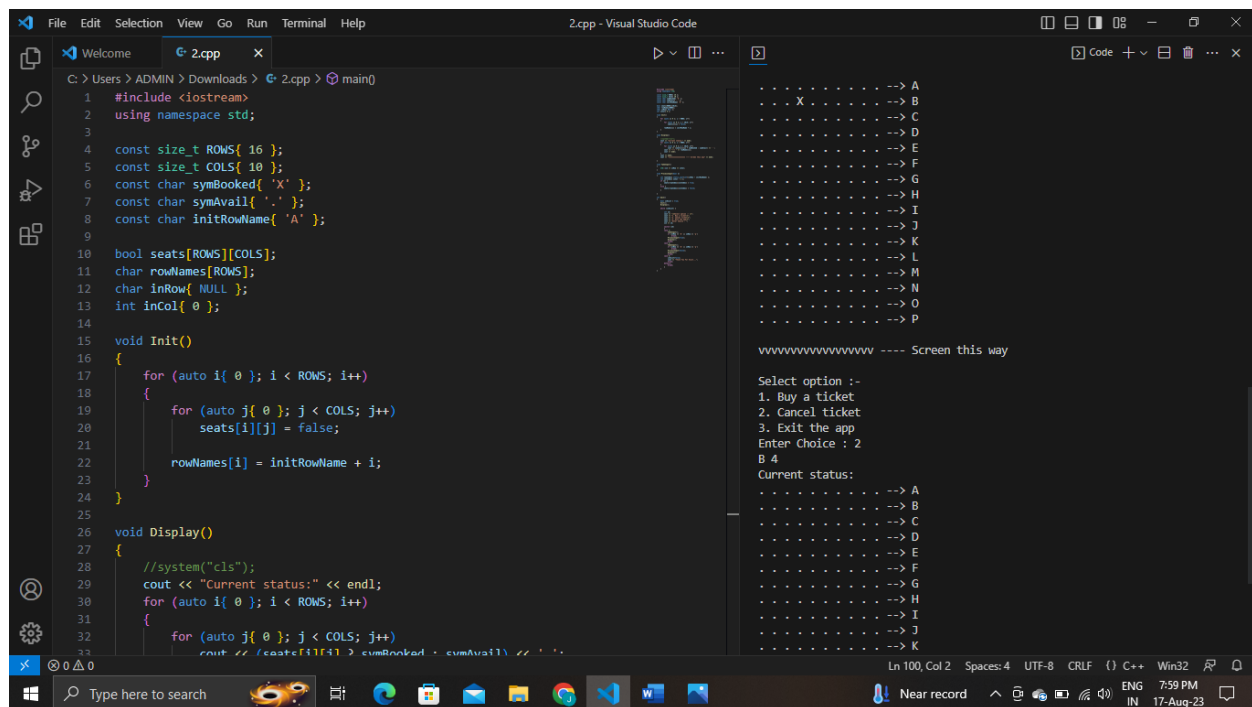
default:

break;

}

}

}



```
1 #include <iostream>
2 using namespace std;
3
4 const size_t ROWS{ 16 };
5 const size_t COLS{ 10 };
6 const char symBooked{ 'X' };
7 const char symAvail{ '.' };
8 const char initRowName{ 'A' };
9
10 bool seats[ROWS][COLS];
11 char rowNames[ROWS];
12 char inRow{ NULL };
13 int inCol{ 0 };
14
15 void Init()
16 {
17     for (auto i{ 0 }; i < ROWS; i++)
18     {
19         for (auto j{ 0 }; j < COLS; j++)
20             seats[i][j] = false;
21
22         rowNames[i] = initRowName + i;
23     }
24 }
25
26 void Display()
27 {
28     //system("cls");
29     cout << "Current status:" << endl;
30     for (auto i{ 0 }; i < ROWS; i++)
31     {
32         for (auto j{ 0 }; j < COLS; j++)
33             cout << (seats[i][j] ? symBooked : symAvail) << " ";
34     }
35 }
```

.....--> A
..X.....--> B
.....--> C
.....--> D
.....--> E
.....--> F
.....--> G
.....--> H
.....--> I
.....--> J
.....--> K
.....--> L
.....--> M
.....--> N
.....--> O
.....--> P

xxxxxxxxxx ---- Screen this way

Select option :-
1. Buy a ticket
2. Cancel ticket
3. Exit the app
Enter Choice : 2
B 4
Current status:
.....--> A
.....--> B
.....--> C
.....--> D
.....--> E
.....--> F
.....--> G
.....--> H
.....--> I
.....--> J
.....--> K

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;

}

The screenshot shows a Visual Studio Code editor with a C++ file named 2.cpp. The code implements a menu-driven program with options to place, pending, serve, or exit orders. It uses a vector to store pending orders and a loop to repeatedly show the menu until the user exits. The terminal output shows the program running, displaying the menu, accepting input, and showing the state of pending orders.

```
File Edit Selection View Go Run Terminal Help
2.cpp - Visual Studio Code

C:\Users\ADMIN> cd "C:\Users\ADMIN\Downloads\" ; if ($?) { g++ 2.cpp -o 2
} ; if ($?) { .\2 }
1) Place order
2) Pending orders
3) Serve order
4) Exit
Enter your pick: 2
=====
Pending orders:
Total pending orders:- 0
=====
1) Place order
2) Pending orders
3) Serve order
4) Exit
Enter your pick: 1
=====
Your order is placed. Order Number:- 1
=====
1) Place order
2) Pending orders
3) Serve order
4) Exit
Enter your pick: 3
=====
Pending orders:
=> Order Number: 1
=====
Pick order: 2
Invalid order number.
=====
1) Place order
2) Pending orders
3) Serve order
4) Exit
Enter your pick: 4
PS C:\Users\ADMIN\Downloads>
```

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;
    }
}

```

[illegible]