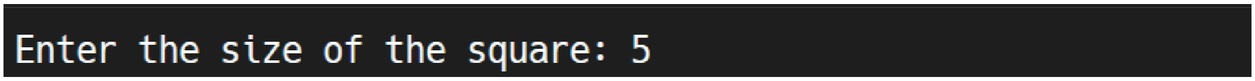
1- draw square

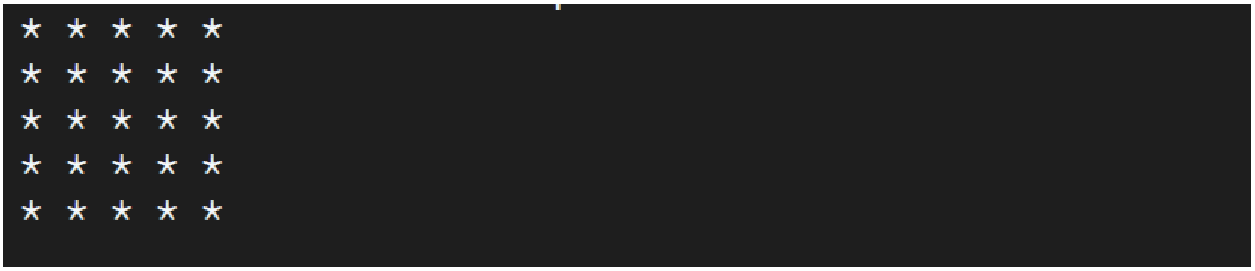
رسم مربع

Input



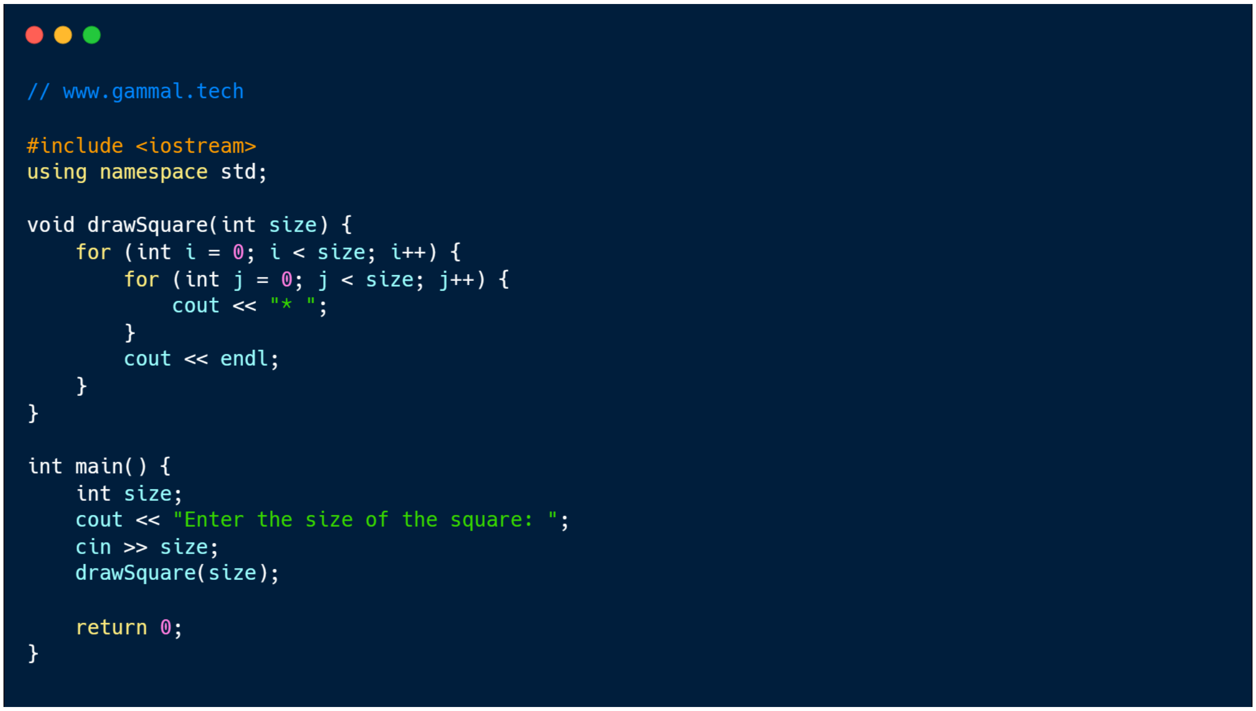
```
Enter the size of the square: 5
```

Output



```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Solution

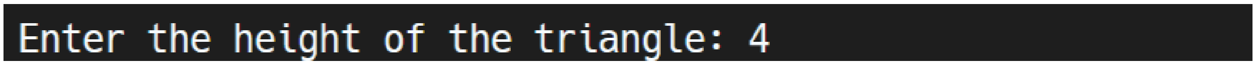


```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawSquare(int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            cout << "* ";
        }
        cout << endl;
    }
}

int main() {
    int size;
    cout << "Enter the size of the square: ";
    cin >> size;
    drawSquare(size);

    return 0;
}
```

2- Draw Triangle

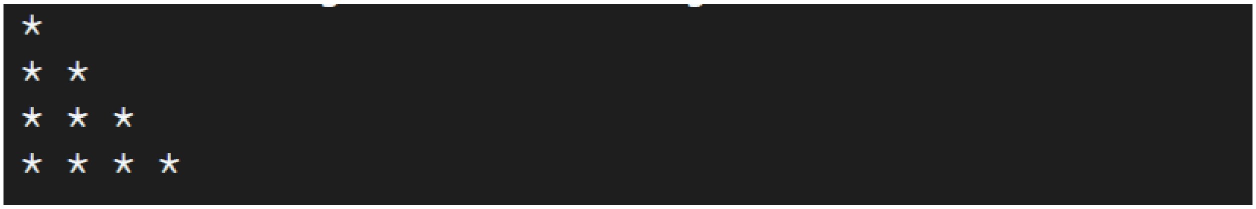
رسم المثلث

Input



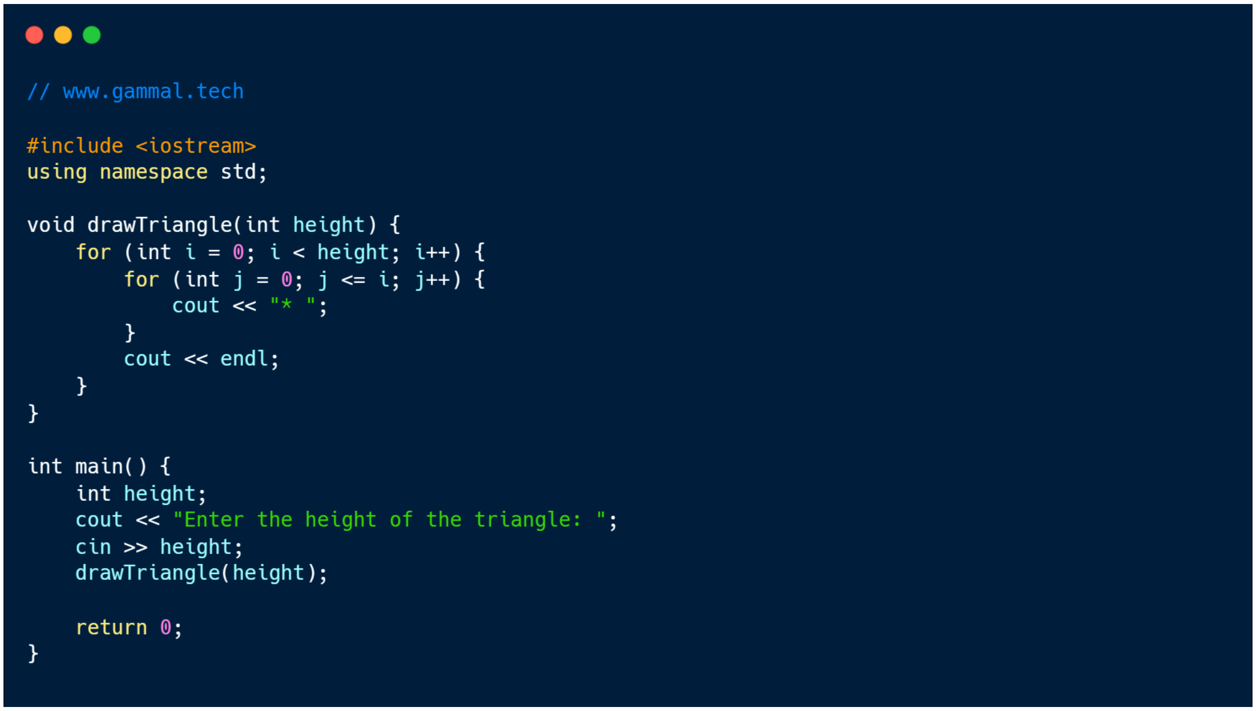
```
Enter the height of the triangle: 4
```

Output



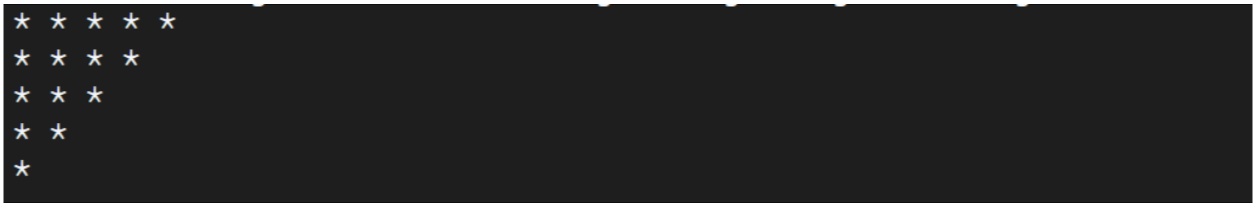
```
*
* *
* * *
* * * *
```

Solution



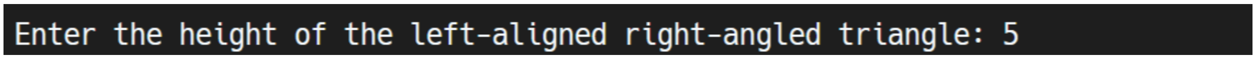
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawTriangle(int height) {
    for (int i = 0; i < height; i++) {
        for (int j = 0; j <= i; j++) {
            cout << "* ";
        }
        cout << endl;
    }
}

int main() {
    int height;
    cout << "Enter the height of the triangle: ";
    cin >> height;
    drawTriangle(height);

    return 0;
}
```

3- Draw Left-aligned Right-Angled Triangle



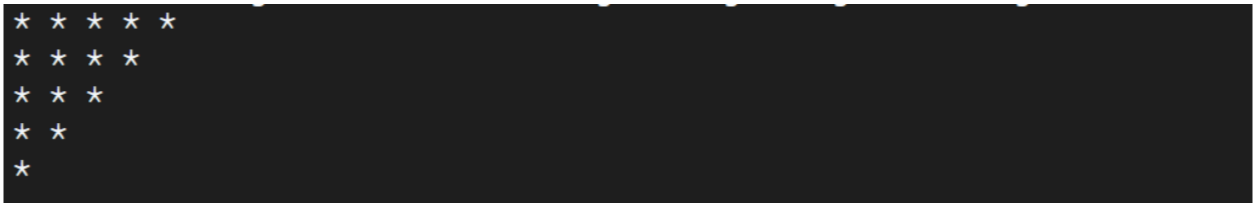
```
* * * * *
* * * *
* * *
* *
*
```

Input



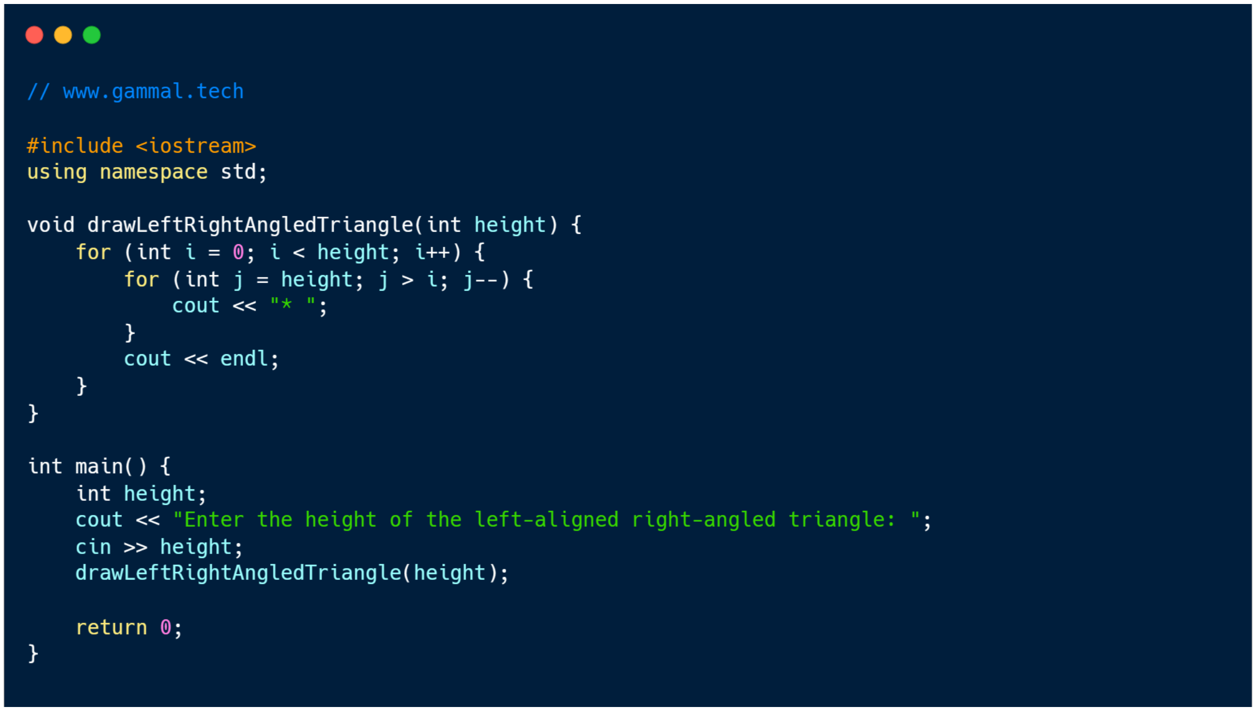
```
Enter the height of the left-aligned right-angled triangle: 5
```

Output



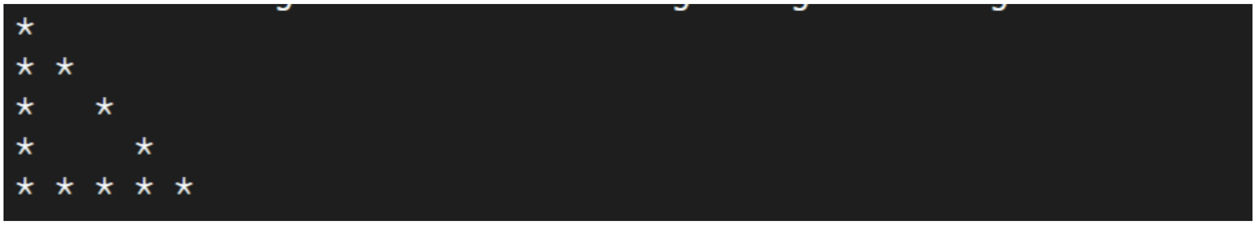
```
* * * * *
* * * *
* * *
* *
*
```

Solution



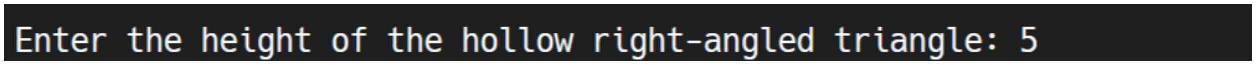
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawLeftRightAngledTriangle(int height) {
    for (int i = 0; i < height; i++) {
        for (int j = height; j > i; j--) {
            cout << "* ";
        }
        cout << endl;
    }
}

int main() {
    int height;
    cout << "Enter the height of the left-aligned right-angled triangle: ";
    cin >> height;
    drawLeftRightAngledTriangle(height);

    return 0;
}
```

4- Draw Hollow Right-Angled Triangle



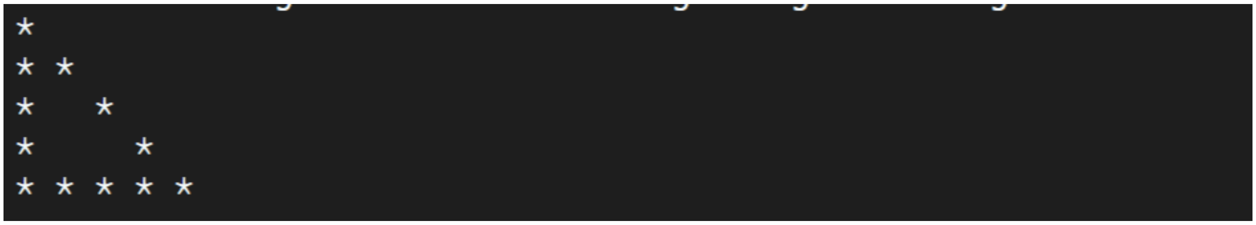
```
*
* *
*   *
*     *
* * * * *
```

Input



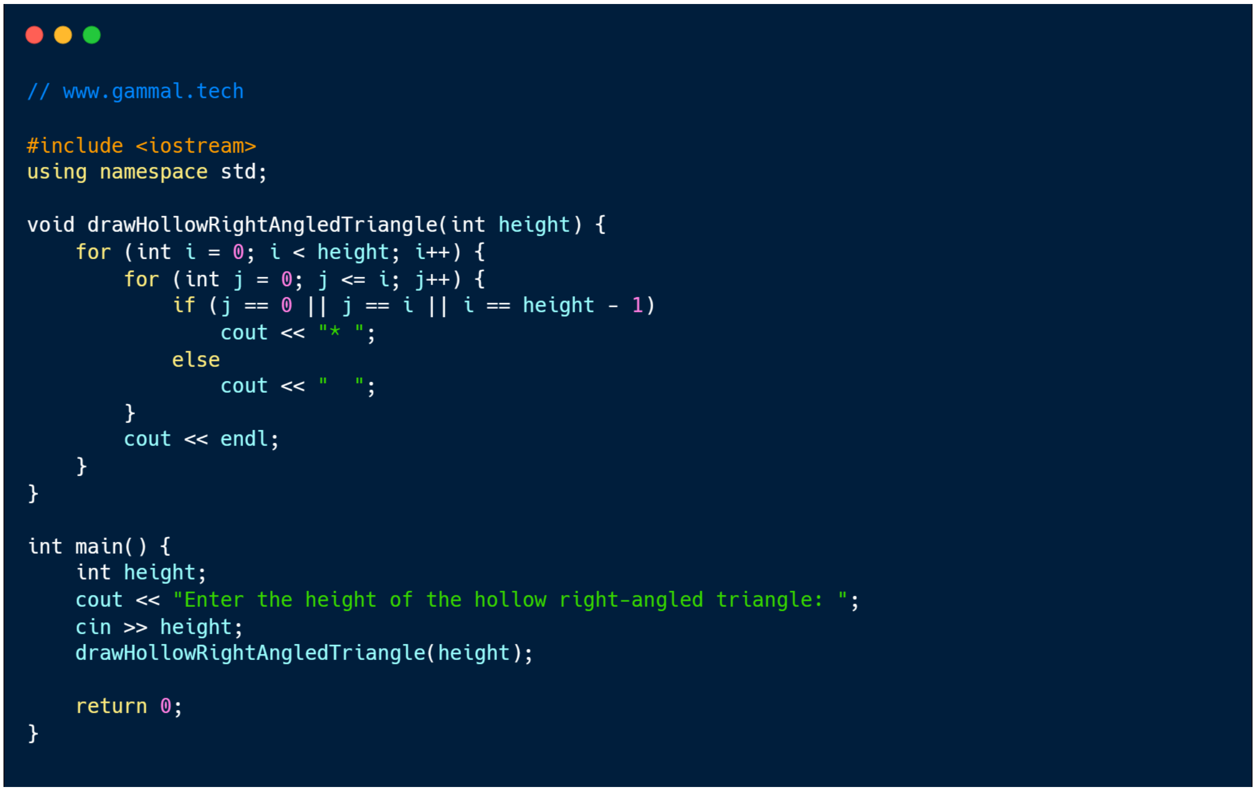
```
Enter the height of the hollow right-angled triangle: 5
```

Output



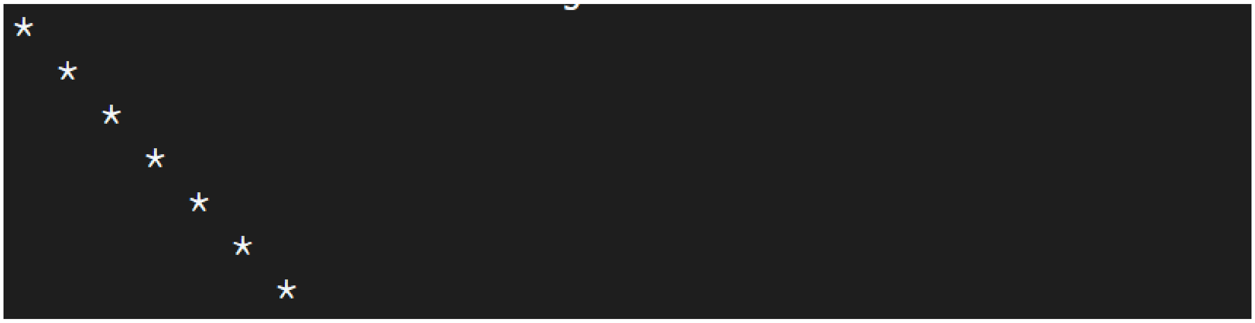
```
*
* *
*   *
*     *
* * * * *
```

Solution



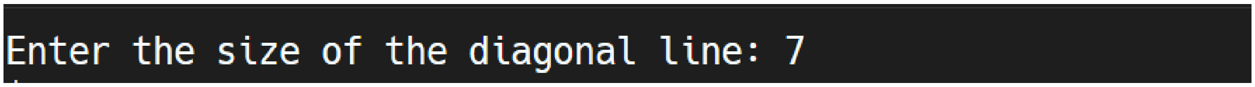
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawHollowRightAngledTriangle(int height) {
    for (int i = 0; i < height; i++) {
        for (int j = 0; j <= i; j++) {
            if (j == 0 || j == i || i == height - 1)
                cout << "* ";
            else
                cout << "  ";
        }
        cout << endl;
    }
}

int main() {
    int height;
    cout << "Enter the height of the hollow right-angled triangle: ";
    cin >> height;
    drawHollowRightAngledTriangle(height);

    return 0;
}
```

5- Draw Diagonal Line



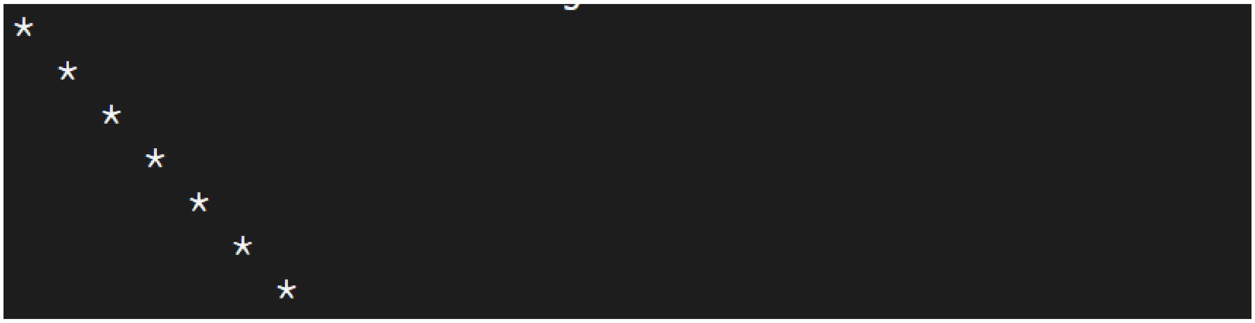
```
*
  *
    *
      *
        *
          *
            *
```

Input



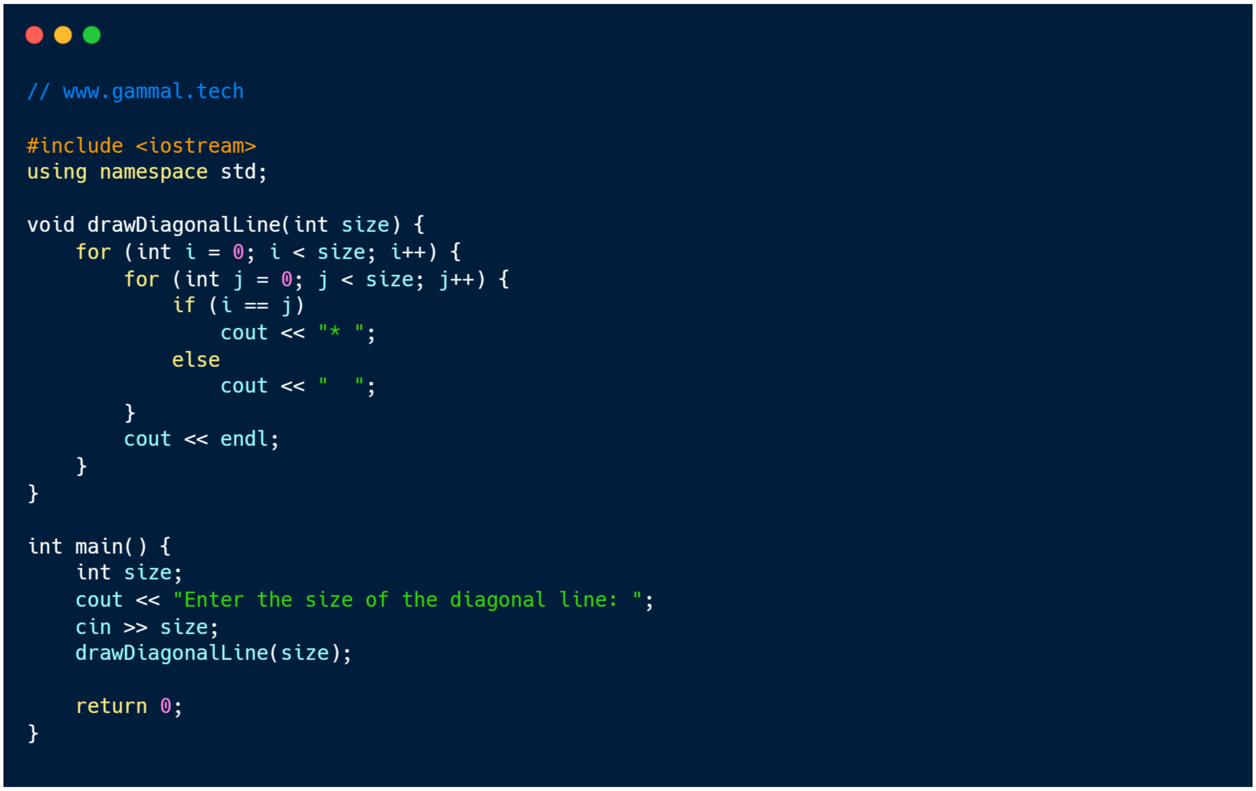
```
Enter the size of the diagonal line: 7
```

Output



```
*
  *
    *
      *
        *
          *
            *
```

Solution



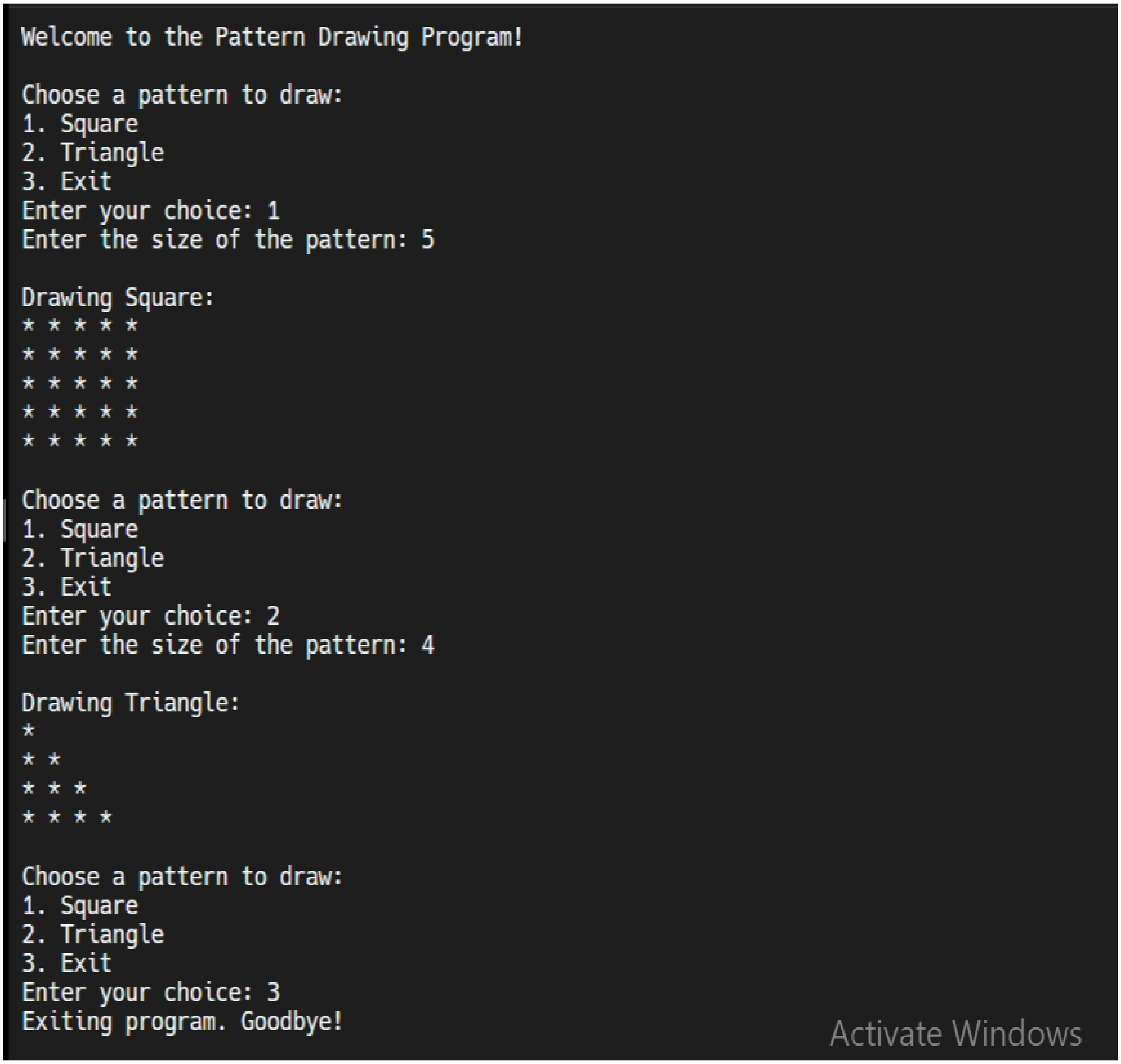
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawDiagonalLine(int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (i == j)
                cout << "* ";
            else
                cout << "  ";
        }
        cout << endl;
    }
}

int main() {
    int size;
    cout << "Enter the size of the diagonal line: ";
    cin >> size;
    drawDiagonalLine(size);

    return 0;
}
```

6- Develop an interactive and user-friendly program for drawing basic patterns. The program should allow users to choose between a square and a triangle, specifying the size of the pattern. Additionally, users should have the option to exit the program.

تطوير برنامج تفاعلي وسهل الاستخدام لرسم الأنماط الأساسية. يجب أن يسمح البرنامج للمستخدمين بالاختيار بين المربع والمثلث، مع تحديد حجم النموذج.

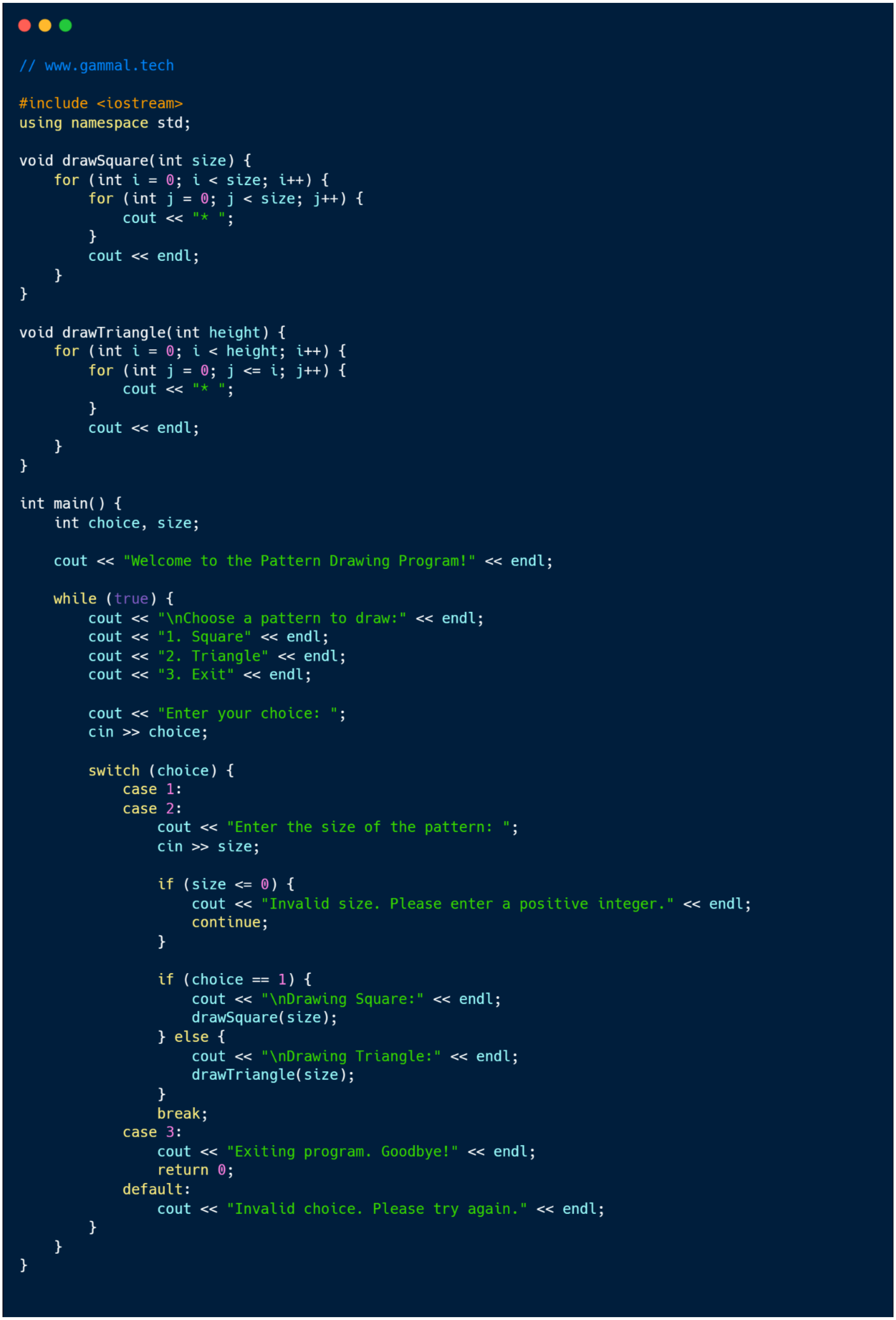
بالإضافة إلى ذلك، يجب أن يكون لدى المستخدمين خيار الخروج من البرنامج.

Input & Output



```
Welcome to the Pattern Drawing Program!

Choose a pattern to draw:
1. Square
2. Triangle
3. Exit
Enter your choice: 1
Enter the size of the pattern: 5

Drawing Square:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Choose a pattern to draw:
1. Square
2. Triangle
3. Exit
Enter your choice: 2
Enter the size of the pattern: 4

Drawing Triangle:
*
* *
* * *
* * * *

Choose a pattern to draw:
1. Square
2. Triangle
3. Exit
Enter your choice: 3
Exiting program. Goodbye!
```

Solution



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawSquare(int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            cout << "* ";
        }
        cout << endl;
    }
}

void drawTriangle(int height) {
    for (int i = 0; i < height; i++) {
        for (int j = 0; j <= i; j++) {
            cout << "* ";
        }
        cout << endl;
    }
}

int main() {
    int choice, size;

    cout << "Welcome to the Pattern Drawing Program!" << endl;

    while (true) {
        cout << "\nChoose a pattern to draw:" << endl;
        cout << "1. Square" << endl;
        cout << "2. Triangle" << endl;
        cout << "3. Exit" << endl;

        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
            case 2:
                cout << "Enter the size of the pattern: ";
                cin >> size;

                if (size <= 0) {
                    cout << "Invalid size. Please enter a positive integer." << endl;
                    continue;
                }

                if (choice == 1) {
                    cout << "\nDrawing Square:" << endl;
                    drawSquare(size);
                } else {
                    cout << "\nDrawing Triangle:" << endl;
                    drawTriangle(size);
                }
                break;
            case 3:
                cout << "Exiting program. Goodbye!" << endl;
                return 0;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    }
}
```

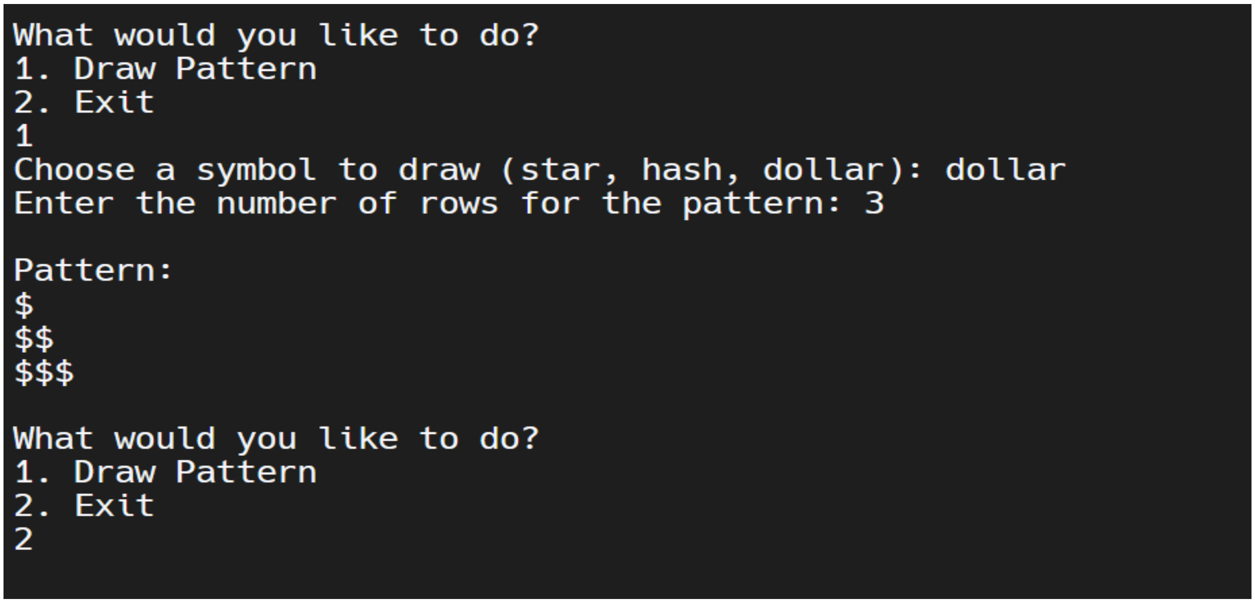
7- Write a program that allows users to draw a simple pattern using predefined symbols. The program should include the following features:

Users can choose from a list of symbols (e.g., "*", "#", "$") to create a pattern.
Users can specify the number of rows for the pattern.
Implement the necessary functions to achieve this functionality. Ensure the program handles invalid symbol selections gracefully.

اكتب برنامجًا يسمح للمستخدمين برسم نمط بسيط باستخدام رموز محددة مسبقًا.
يجب أن يتضمن البرنامج الميزات التالية:

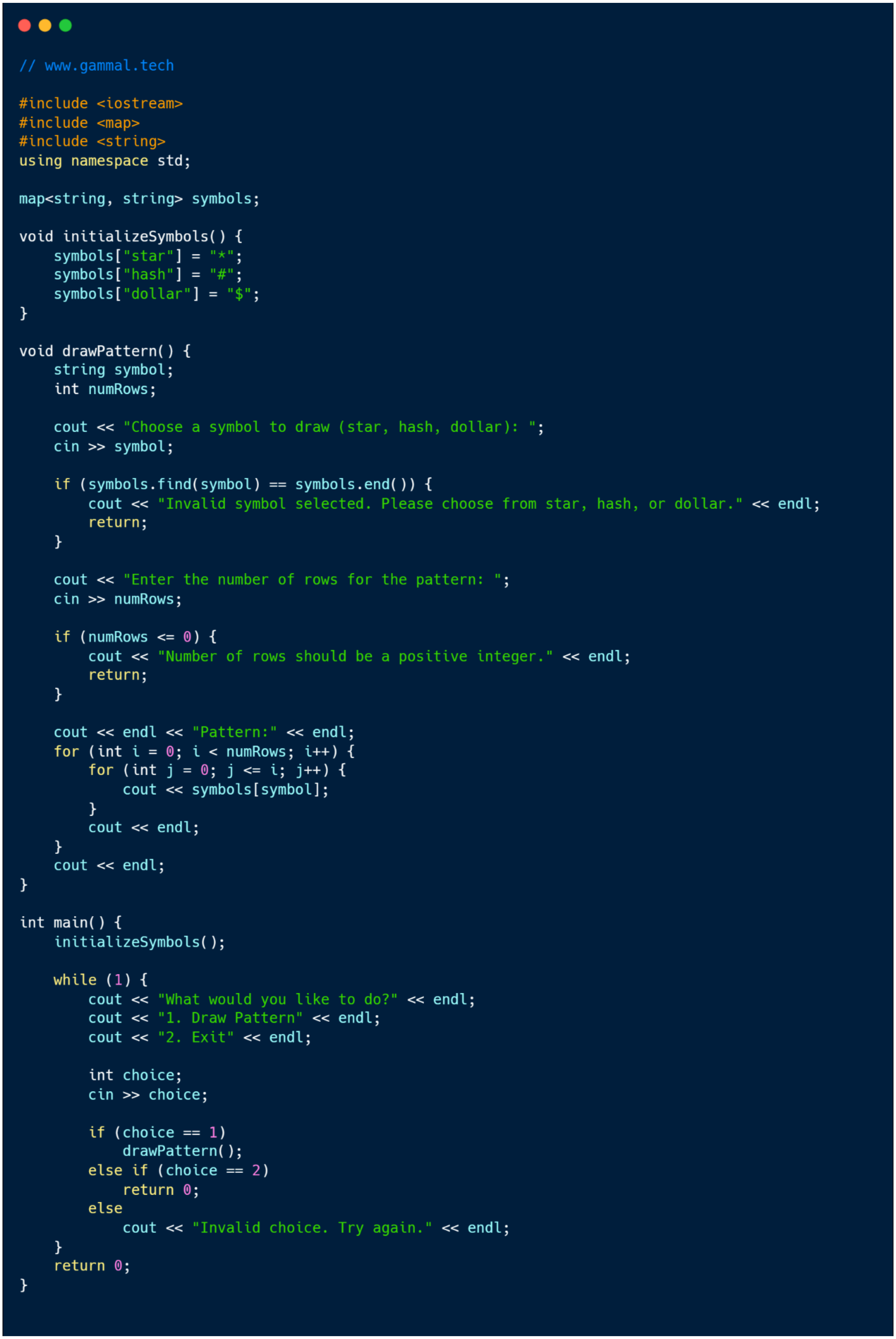
يمكن للمستخدمين الاختيار من قائمة الرموز (على سبيل المثال، "*"، "#"، "$")
لإنشاء نمط.
يمكن للمستخدمين تحديد عدد الصفوف للنمط.
تنفيذ الوظائف اللازمة لتحقيق هذه الوظيفة. تأكد من أن البرنامج يتعامل مع
تحديدات الرموز غير الصالحة بأمان.

Input & Output



```
What would you like to do?
1. Draw Pattern
2. Exit
1
Choose a symbol to draw (star, hash, dollar): dollar
Enter the number of rows for the pattern: 3

Pattern:
$
$$
$$$

What would you like to do?
1. Draw Pattern
2. Exit
2
```

Solution



```cpp
// www.gammal.tech

#include <iostream>
#include <map>
#include <string>
using namespace std;

map<string, string> symbols;

void initializeSymbols() {
    symbols["star"] = "*";
    symbols["hash"] = "#";
    symbols["dollar"] = "$";
}

void drawPattern() {
    string symbol;
    int numRows;

    cout << "Choose a symbol to draw (star, hash, dollar): ";
    cin >> symbol;

    if (symbols.find(symbol) == symbols.end()) {
        cout << "Invalid symbol selected. Please choose from star, hash, or dollar." << endl;
        return;
    }

    cout << "Enter the number of rows for the pattern: ";
    cin >> numRows;

    if (numRows <= 0) {
        cout << "Number of rows should be a positive integer." << endl;
        return;
    }

    cout << endl << "Pattern:" << endl;
    for (int i = 0; i < numRows; i++) {
        for (int j = 0; j <= i; j++) {
            cout << symbols[symbol];
        }
        cout << endl;
    }
    cout << endl;
}

int main() {
    initializeSymbols();

    while (1) {
        cout << "What would you like to do?" << endl;
        cout << "1. Draw Pattern" << endl;
        cout << "2. Exit" << endl;

        int choice;
        cin >> choice;

        if (choice == 1)
            drawPattern();
        else if (choice == 2)
            return 0;
        else
            cout << "Invalid choice. Try again." << endl;
    }
    return 0;
}
```

8- ًWrite a program that allows users to add custom symbols and draw patterns using those symbols. The program should offer the following functionalities:

Add Symbols: Allow users to add custom symbols by providing a name and the corresponding symbol.

Draw Pattern: Enable users to draw patterns using the previously defined symbols. Users can specify the repetition count for each symbol, and the program will display the resulting pattern.

Exit: Provide an option to exit the program.

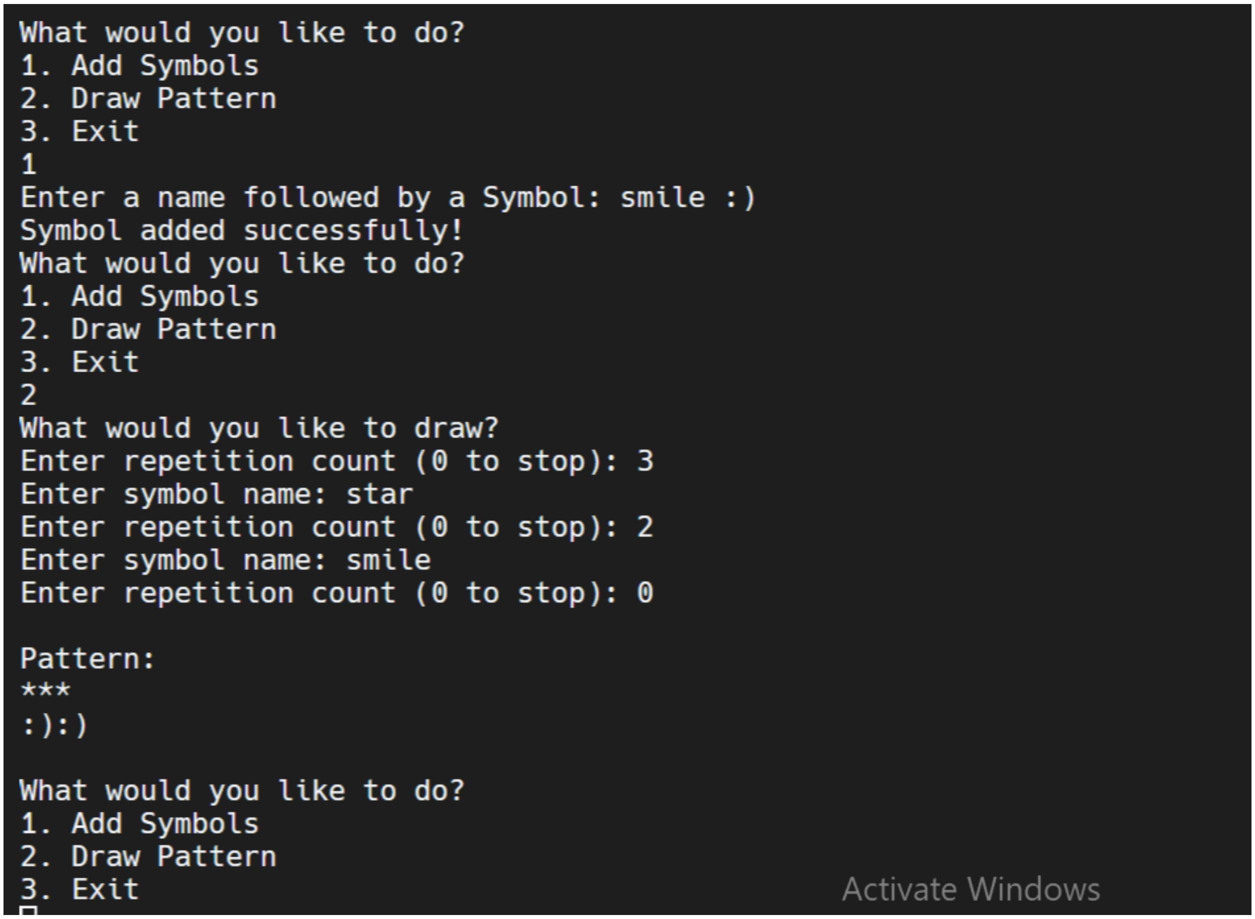
كتابة برنامج يسمح للمستخدمين بإضافة رموز مخصصة ورسم أنماط باستخدام تلك الرموز. يجب أن يوفر البرنامج الوظائف التالية:

إضافة رموز: السماح للمستخدمين بإضافة رموز مخصصة من خلال توفير الاسم والرمز المقابل.

رسم النمط: تمكين المستخدمين من رسم الأنماط باستخدام الرموز المحددة مسبقًا. يمكن للمستخدمين تحديد عدد التكرار لكل رمز، وسيعرض البرنامج النمط الناتج.

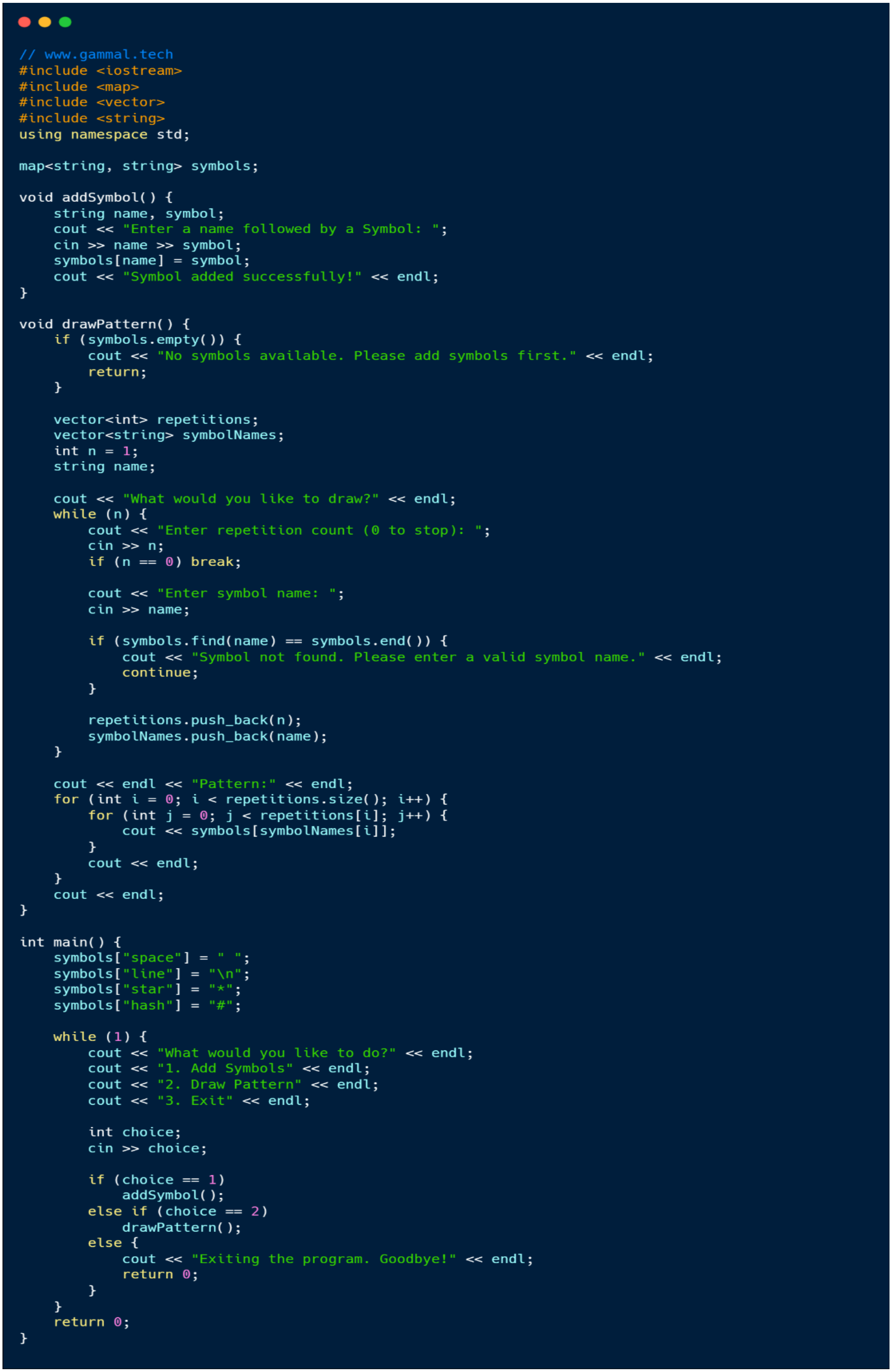
خروج: توفير خيار للخروج من البرنامج.

Input & Output



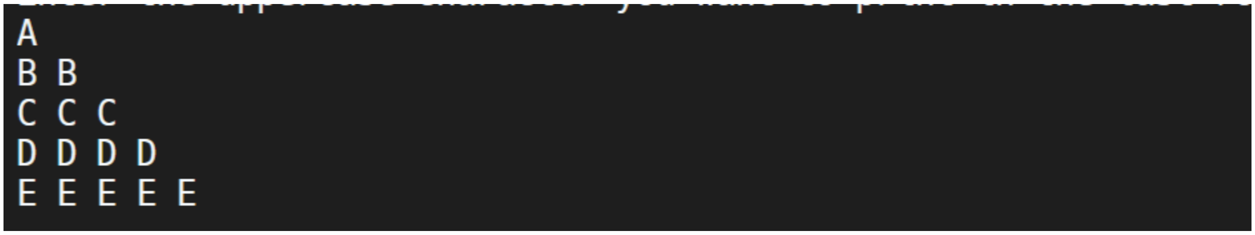
```
What would you like to do?
1. Add Symbols
2. Draw Pattern
3. Exit
1
Enter a name followed by a Symbol: smile :)
Symbol added successfully!
What would you like to do?
1. Add Symbols
2. Draw Pattern
3. Exit
2
What would you like to draw?
Enter repetition count (0 to stop): 3
Enter symbol name: star
Enter repetition count (0 to stop): 2
Enter symbol name: smile
Enter repetition count (0 to stop): 0

Pattern:
***
:):)

What would you like to do?
1. Add Symbols
2. Draw Pattern
3. Exit
```

Solution



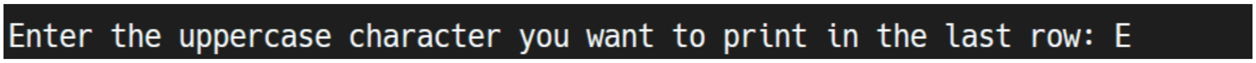
```
// www.gammal.tech
#include <iostream>
#include <map>
#include <vector>
#include <string>
using namespace std;

map<string, string> symbols;

void addSymbol() {
    string name, symbol;
    cout << "Enter a name followed by a Symbol: ";
    cin >> name >> symbol;
    symbols[name] = symbol;
    cout << "Symbol added successfully!" << endl;
}

void drawPattern() {
    if (symbols.empty()) {
        cout << "No symbols available. Please add symbols first." << endl;
        return;
    }

    vector<int> repetitions;
    vector<string> symbolNames;
    int n = 1;
    string name;

    cout << "What would you like to draw?" << endl;
    while (n) {
        cout << "Enter repetition count (0 to stop): ";
        cin >> n;
        if (n == 0) break;

        cout << "Enter symbol name: ";
        cin >> name;

        if (symbols.find(name) == symbols.end()) {
            cout << "Symbol not found. Please enter a valid symbol name." << endl;
            continue;
        }

        repetitions.push_back(n);
        symbolNames.push_back(name);
    }

    cout << endl << "Pattern:" << endl;
    for (int i = 0; i < repetitions.size(); i++) {
        for (int j = 0; j < repetitions[i]; j++) {
            cout << symbols[symbolNames[i]];
        }
        cout << endl;
    }
    cout << endl;
}

int main() {
    symbols["space"] = " ";
    symbols["line"] = "\n";
    symbols["star"] = "*";
    symbols["hash"] = "#";

    while (1) {
        cout << "What would you like to do?" << endl;
        cout << "1. Add Symbols" << endl;
        cout << "2. Draw Pattern" << endl;
        cout << "3. Exit" << endl;

        int choice;
        cin >> choice;

        if (choice == 1)
            addSymbol();
        else if (choice == 2)
            drawPattern();
        else {
            cout << "Exiting the program. Goodbye!" << endl;
            return 0;
        }
    }
    return 0;
}
```

9- Draw this shape



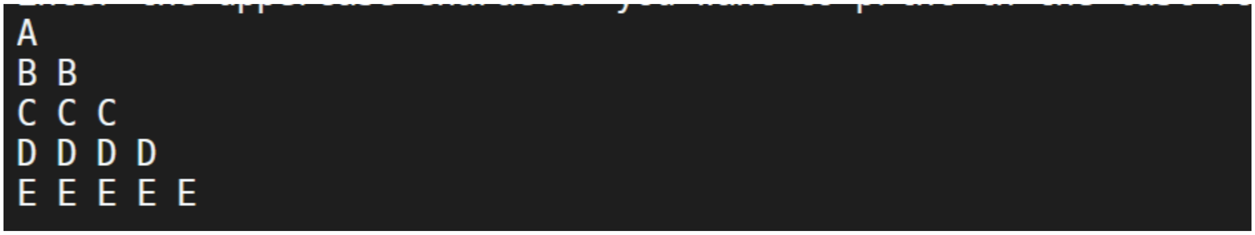
```
A
B B
C C C
D D D D
E E E E E
```

Input



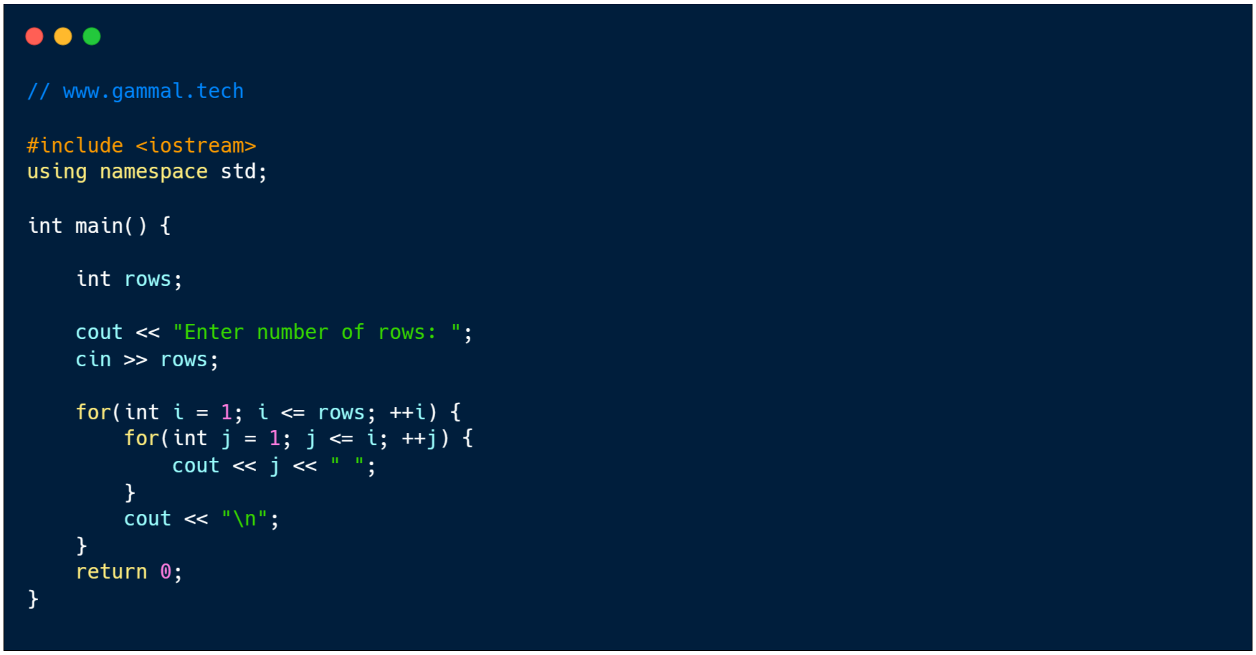
```
Enter the uppercase character you want to print in the last row: E
```

Output



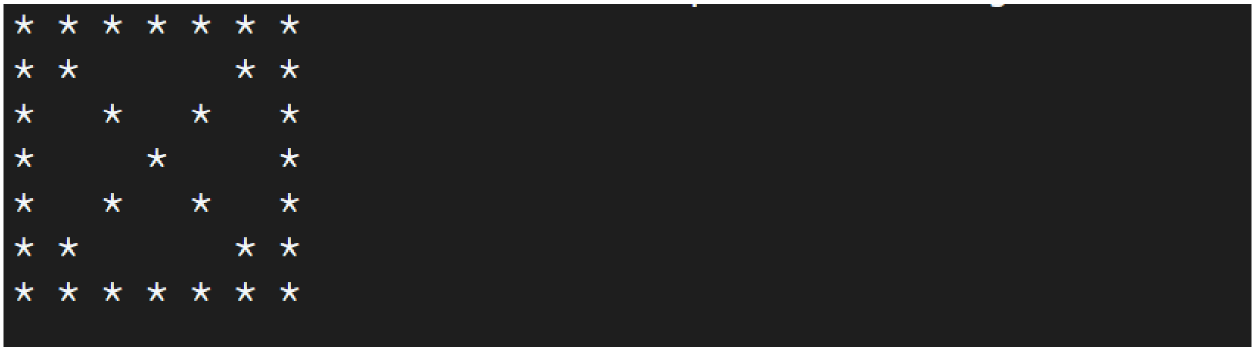
```
A
B B
C C C
D D D D
E E E E E
```

Solution



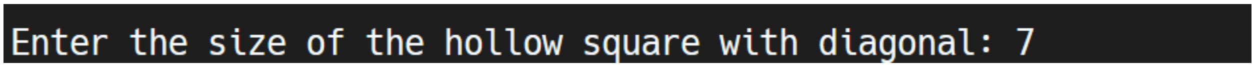
```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {

    int rows;

    cout << "Enter number of rows: ";
    cin >> rows;

    for(int i = 1; i <= rows; ++i) {
        for(int j = 1; j <= i; ++j) {
            cout << j << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

10- Draw Hollow Square with Diagonal



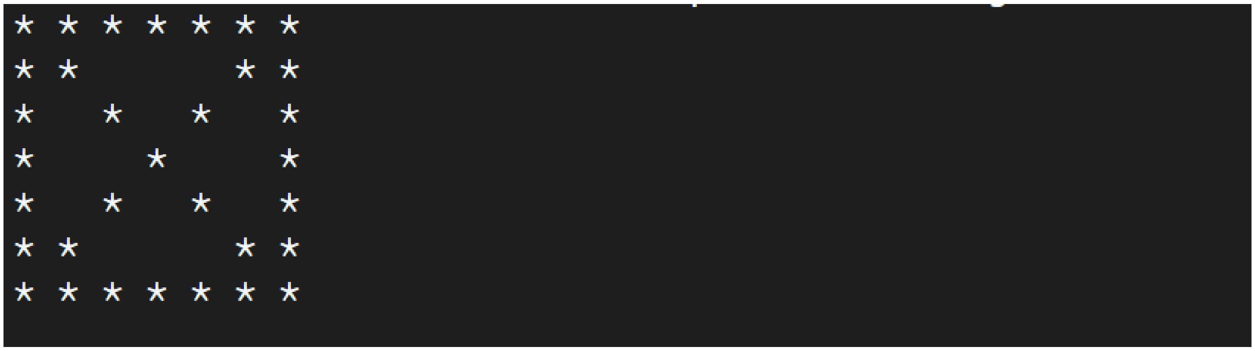
```
* * * * * * *
* *       * *
*   *   *   *
*     *     *
*   *   *   *
* *       * *
* * * * * * *
```

Input



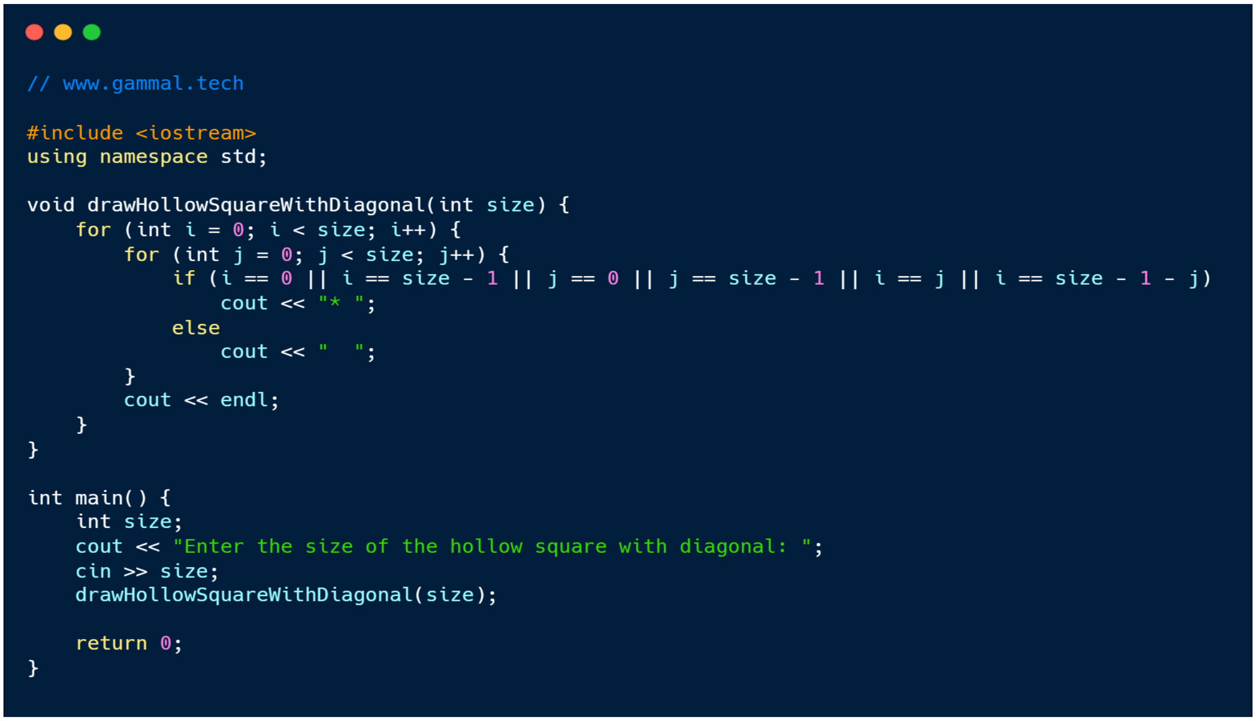
```
Enter the size of the hollow square with diagonal: 7
```

Output



```
* * * * * * *
* *       * *
*   *   *   *
*     *     *
*   *   *   *
* *       * *
* * * * * * *
```

Solution



```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void drawHollowSquareWithDiagonal(int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (i == 0 || i == size - 1 || j == 0 || j == size - 1 || i == j || i == size - 1 - j)
                cout << "* ";
            else
                cout << "  ";
        }
        cout << endl;
    }
}

int main() {
    int size;
    cout << "Enter the size of the hollow square with diagonal: ";
    cin >> size;
    drawHollowSquareWithDiagonal(size);

    return 0;
}
```