patterns

November 24, 2024

1 Patterns

[5]:

```
#include <iostream>
using namespace std;
```

Square Pattern

[ ]:

```
int n = 5;

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

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

Right Triangle Pattern

[7]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

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

Left Triangle Pattern

[8]:
```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

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

Pyramid Pattern

[9]:
```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        cout << "* ";
    }
    cout << endl;
}
```

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

Inverted Pyramid Pattern

[ ]:
```
int n = 5;

for (int i = n; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        cout << "* ";
    }
    cout << endl;
}
```

Diamond Pattern

[10]:
```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        cout << "* ";
    }
    cout << endl;
}

for (int i = n - 1; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        cout << "* ";
    }
    cout << endl;
}
```

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

Hollow Square Pattern*

[11]:
```
int n = 5;

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

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

Hollow Right Triangle Pattern

[12]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        if (i == n || j == 1 || j == i) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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

Hollow Left Triangle Pattern

[13]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        if (i == n || j == 1 || j == i) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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

Hollow Pyramid Pattern

[14]:
```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        if (i == n || j == 1 || j == (2 * i - 1)) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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

Hollow Inverted Pyramid Pattern

[15]:
```
int n = 5;

for (int i = n; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        if (i == n || j == 1 || j == (2 * i - 1)) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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

Hollow Diamond Pattern

[16]:
```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        if (j == 1 || j == (2 * i - 1)) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}

for (int i = n - 1; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= (2 * i - 1); j++) {
        if (j == 1 || j == (2 * i - 1)) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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

Number Square Pattern

[17]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

Number Right Triangle Pattern

[18]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Number Left Triangle Pattern

[19]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
        1
      1 2
    1 2 3
```

```
  1 2 3 4
1 2 3 4 5
```

Number Pyramid Pattern

[20]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

Number Inverted Pyramid Pattern

[23]:

```
int n = 5;

for (int i = n; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

Number Diamond Pattern

[21]:

```
int n = 5;

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << j << " ";
    }
    cout << endl;
}

for (int i = n - 1; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << j << " ";
    }
    cout << endl;
}
```

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

Alphabet Square Pattern

[29]:

```
int n = 5;
char ch = 'A';

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cout << ch << " ";
        ch++;
    }
    cout << endl;
}
```

```
A B C D E
F G H I J
K L M N O
P Q R S T
U V W X Y
```

Alphabet Right Triangle Pattern

[26]:

```
int n = 5;
char ch = 'A';

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
        ch++;
    }
    cout << endl;
}
```

```
A
B C
D E F
G H I J
K L M N O
```

Alphabet Left Triangle Pattern

[30]:

```
int n = 5;
char ch = 'A';

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
        ch++;
    }
    cout << endl;
}
```

```
        A
      B C
    D E F
  G H I J
K L M N O
```

Alphabet Pyramid Pattern

[32]:

```cpp
int n = 5;
char ch = 'A';

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << ch << " ";
    }
    cout << endl;
    ch++;
}
```

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

Alphabet Inverted Pyramid Pattern

[33]:

```cpp
int n = 5;
char ch = 'A' + n - 1;

for (int i = n; i >= 1; i--) {
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
    }
    for (int j = i - 1; j >= 1; j--) {
        cout << ch << " ";
    }
    cout << endl;
    ch--;
}
```

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

Alphabet Diamond Pattern

[34]:

```cpp
int n = 5;
char ch;

for (int i = 1; i <= n; i++) {
    ch = 'A';
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
        ch++;
    }
    ch--;
    for (int j = 1; j < i; j++) {
        ch--;
        cout << ch << " ";
    }
    cout << endl;
}

for (int i = n - 1; i >= 1; i--) {
    ch = 'A';
    for (int j = i; j < n; j++) {
        cout << "  ";
    }
    for (int j = 1; j <= i; j++) {
        cout << ch << " ";
        ch++;
    }
    ch--;
    for (int j = 1; j < i; j++) {
        ch--;
        cout << ch << " ";
    }
    cout << endl;
}
```

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

Cross Pattern

[35]:

```
int n = 5;

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

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

Plus Pattern

[36]:

```
int n = 5;
int mid = n / 2;

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (i == mid || j == mid) {
            cout << "* ";
        } else {
            cout << "  ";
        }
    }
    cout << endl;
}
```

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