**Basic Patterns (1-10)**

**1. Right-Angled Triangle with Numbers**

**Question:** Print the following pattern:

1

12

123

1234

12345

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

cout << j;

}

cout << endl;

}

return 0;

}

**2. Reverse Right-Angled Triangle**

**Question:** Print the following pattern:

54321

5432

543

54

5

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = n; i >= 1; i--) {

for (int j = i; j >= 1; j--) {

cout << j;

}

cout << endl;

}

return 0;

}

**3. Right-Angled Triangle with Stars**

**Question:** Print the following pattern:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**4. Inverted Right-Angled Triangle with Stars**

**Question:** Print the following pattern:

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = n; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**5. Hollow Right-Angled Triangle**

**Question:** Print the following hollow right-angled triangle:

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

if (j == 1 || j == i || i == n) {

cout << "\*";

} else {

cout << " ";

}

}

cout << endl;

}

return 0;

}

**6. Hollow Inverted Right-Angled Triangle**

**Question:** Print the following hollow inverted right-angled triangle:

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = n; i >= 1; i--) {

for (int j = 1; j <= i; j++) {

if (j == 1 || j == i || i == n) {

cout << "\*";

} else {

cout << " ";

}

}

cout << endl;

}

return 0;

}

**7. Right-Angled Triangle with Alphabet**

**Question:** Print the following pattern with alphabets:

A

AB

ABC

ABCD

ABCDE

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

cout << char('A' + j - 1);

}

cout << endl;

}

return 0;

}

**8. Square with Numbers**

**Question:** Print the following square pattern with numbers:

11111

22222

33333

44444

55555

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n; j++) {

cout << i;

}

cout << endl;

}

return 0;

}

**9. Number Pyramid**

**Question:** Print the following number pyramid:

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

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j<<” “;

}

for (int j = i - 1; j >= 1; j--) {

cout << j<<” ”;

}

cout << endl;

}

return 0;

}

**10. Diamond Pattern**

**Question:** Print the following diamond pattern:

1

121

12321

1234321

123454321

1234321

12321

121

1

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

// Upper part of the diamond

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

for (int j = i - 1; j >= 1; j--) {

cout << j;

}

cout << endl;

}

// Lower part of the diamond

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

for (int j = i - 1; j >= 1; j--) {

cout << j;

}

cout << endl;

}

return 0;

}

**Intermediate Patterns (11-15)**

**11. Inverted Number Pyramid**

**Question:** Print the following inverted number pyramid:

123454321

1234321

12321

121

1

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = n; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

for (int j = i - 1; j >= 1; j--) {

cout << j;

}

cout << endl;

}

return 0;

}

**12. Checkerboard Pattern**

**Question:** Print the following checkerboard pattern:

\* \* \* \*

\* \* \*

\* \* \* \*

\* \* \*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n; j++) {

if ((i + j) % 2 == 0) {

cout << "\* ";

} else {

cout << " ";

}

}

cout << endl;

}

return 0;

}

**13. Pascal's Triangle**

**Question:** Print the first n rows of Pascal's Triangle.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**Solution:**

#include <iostream>

using namespace std;

int factorial(int n) {

int fact = 1;

for (int i = 1; i <= n; i++) {

fact \*= i;

}

return fact;

}

int main() {

int n = 5;

for (int i = 0; i < n; i++) {

for (int j = 0; j <= i; j++) {

cout << factorial(i) / (factorial(j) \* factorial(i - j)) << " ";

}

cout << endl;

}

return 0;

}

**14. Right-Angled Triangle with Spaces**

**Question:** Print the following pattern:

1

12

123

1234

12345

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

cout << endl;

}

return 0;

}

**15. Star Pyramid**

**Question:** Print the following star pyramid:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**Advanced Patterns (16-25)**

**16. Diamond Shape with Stars**

**Question:** Print a diamond shape with stars:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

// Upper part of the diamond

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

// Lower part of the diamond

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**18. Heart Pattern**

**Question: Print a heart pattern:**

**\*\*\* \*\*\***

**\*\*\*\*\* \*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\***

**\*\*\***

**\***

**Solution:**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n = 6;**

**// Upper part of the heart**

**for (int i = n / 2; i <= n; i += 2) {**

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

**cout << " ";**

**}**

**for (int j = 1; j <= i; j++) {**

**cout << "\*";**

**}**

**for (int j = 1; j <= n - i; j++) {**

**cout << " ";**

**}**

**for (int j = 1; j <= i; j++) {**

**cout << "\*";**

**}**

**cout << endl;**

**}**

**// Lower part of the heart**

**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;**

**}**

**return 0;**

**}**

**19. Hollow Diamond Pattern**

**Question:** Print a hollow diamond pattern:

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

// Upper part of the diamond

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= (2 \* i - 1); j++) {

if (j == 1 || j == (2 \* i - 1)) {

cout << "\*";

} else {

cout << " ";

}

}

cout << endl;

}

// Lower part of the diamond

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= (2 \* i - 1); j++) {

if (j == 1 || j == (2 \* i - 1)) {

cout << "\*";

} else {

cout << " ";

}

}

cout << endl;

}

return 0;

}

**20. Floyd's Triangle**

**Question:** Print Floyd's Triangle:

Copy code

1

2 3

4 5 6

7 8 9 10

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 4;

int num = 1;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

cout << num << " ";

num++;

}

cout << endl;

}

return 0;

}

**20. Inverted Star Pyramid**

**Question:** Print the inverted star pyramid:

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

for (int i = n; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**21. Hourglass Pattern**

**Question:** Print an hourglass pattern:

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

// Upper part of hourglass

for (int i = n; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

// Lower part of hourglass

for (int i = 2; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

**22. Number Diamond**

**Question:** Print a number diamond pattern:

1

121

12321

1234321

123454321

1234321

12321

121

1

**Solution:**

#include <iostream>

using namespace std;

int main() {

int n = 5;

// Upper part of the diamond

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

for (int j = i - 1; j >= 1; j--) {

cout << j;

}

cout << endl;

}

// Lower part of the diamond

for (int i = n - 1; i >= 1; i--) {

for (int j = 1; j <= n - i; j++) {

cout << " ";

}

for (int j = 1; j <= i; j++) {

cout << j;

}

for (int j = i - 1; j >= 1; j--) {

cout << j;

}

cout << endl;

}

return 0;

}