LAB 07 TASKS

Question#01) Write a C++ program to input an upper bound by user and print all numbers up to that limit?

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int upperLimit;
    cout << "Enter upper limit: ";
    cin >> upperLimit;
    for (int i = 1; i <= upperLimit; i++) {
        cout << i << " ";
    }
    return 0;
}
```

Output

```
Enter upper limit: 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
--------------------------------
Process exited after 10.23 seconds with return value 0
Press any key to continue . . .
```

Question#02) Write a C++ program to input an upper bound and separate even & odd from that limit?

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int upperLimit;
    cout << "Enter upper limit: ";
    cin >> upperLimit;
    cout << "Even numbers: ";
    for (int i = 2; i <= upperLimit; i += 2) cout << i << " ";
    cout << "\nOdd numbers: ";
    for (int i = 1; i <= upperLimit; i += 2) cout << i << " ";
    return 0;
}
```

Output

```
Enter upper limit: 12
Even numbers: 2 4 6 8 10 12
Odd numbers: 1 3 5 7 9 11
--------------------------------
Process exited after 10.8 seconds with return value 0
Press any key to continue . . .
```

Question #03) Write a C++ program that prints complete alphabets using for loop? (a-z)

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    for (char c = 'a'; c <= 'z'; c++) {
        cout << c << " ";
    }
    return 0;
}
```

Output

```
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 z
--------------------------------
Process exited after 0.1244 seconds with return value 0
Press any key to continue . . .
```

Question #04) Write a C++ program that ask the user a number and prints the table of that number up to 10?

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    for (int i = 1; i <= 10; i++) {
        cout << number << " x " << i << " = " << number * i << "\n";
    }
    return 0;
}
```

Output

```
Enter a number: 10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

--------------------------------
Process exited after 2.478 seconds with return value 0
Press any key to continue . . .
```

Question #05) Write a C++ program that displays the product of all odd numbers from 1 to 10 using for loop?

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int product = 1;
    for (int i = 1; i <= 10; i += 2) {
        product *= i;
    }
    cout << "Product of odd numbers from 1 to 10 is: " << product;
    return 0;
}
```

Output

```
Product of odd numbers from 1 to 10 is: 945
--------------------------------
Process exited after 0.1496 seconds with return value 0
Press any key to continue . . .
```

Question #06) Write a C++ program that finds the sum of squares of integer from 1 to n. Where n is a positive value entered by user (i.e. $1^2 + 2^2 + 3^2 \ldots\ldots\ldots + n^2$).

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int n, sum = 0;
    cout << "Enter a positive number: ";
    cin >> n;
    for (int i = 1; i <= n; i++) {
        sum += i * i;
    }
    cout << "Sum of squares from 1 to " << n << " is: " << sum;
    return 0;
}
```

Output

```
Enter a positive number: 36
Sum of squares from 1 to 36 is: 16206
--------------------------------
Process exited after 5.327 seconds with return value 0
Press any key to continue . . .
```

Question #07) Write a C++ program counts the number of digits in the number entered by user?

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number, count = 0;
    cout << "Enter a number: ";
    cin >> number;
    while (number != 0) {
        count++;
        number /= 10;
    }
    cout << "Number of digits: " << count;
    return 0;
}
```

Output

```
Enter a number: 20
Number of digits: 2
--------------------------------
Process exited after 6.11 seconds with return value 0
Press any key to continue . . .
```

Question # 08) Write a C++ program to reverse a number entered by user using loop.

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number, reverse = 0;
    cout << "Enter a number: ";
    cin >> number;
    while (number != 0) {
        reverse = reverse * 10 + number % 10;
        number /= 10;
    }
    cout << "Reversed number: " << reverse;
    return 0;
}
```

Output

```
Enter a number: 356
Reversed number: 653
--------------------------------
Process exited after 3.604 seconds with return value 0
Press any key to continue . . .
```

Question # 09) Write a program to find first and last digit of a number.

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number, firstDigit, lastDigit;
    cout << "Enter a number: ";
    cin >> number;
    lastDigit = number % 10;
    while (number >= 10) {
        number /= 10;
    }
    firstDigit = number;
    cout << "First digit: " << firstDigit << "\nLast digit: " << lastDigit;
    return 0;
}
```

Output

```
Enter a number: 2048
First digit: 2
Last digit: 8
--------------------------------
Process exited after 6.346 seconds with return value 0
Press any key to continue . . .
```

Question # 10) Write a program to calculate product of digits of a number.

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number, product = 1;
    cout << "Enter a number: ";
    cin >> number;
    while (number != 0) {
        product *= number % 10;
        number /= 10;
    }
    cout << "Product of digits: " << product;
    return 0;
}
```

Output

```
Enter a number: 47
Product of digits: 28
--------------------------------
Process exited after 3.127 seconds with return value 0
Press any key to continue . . .
```

Question # 11) Write a program to find frequency of each digit in a given integer.

Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int number, frequency[10] = {0};
    cout << "Enter a number: ";
    cin >> number;
    while (number != 0) {
        frequency[number % 10]++;
        number /= 10;
    }
    for (int i = 0; i < 10; i++) {
        if (frequency[i] > 0) cout << "Digit " << i << " occurs " << frequency[i] << " times\n";
    }
    return 0;
}
```

Output

```
Enter a number: 272747
Digit 2 occurs 2 times
Digit 4 occurs 1 times
Digit 7 occurs 3 times

--------------------------------
Process exited after 9.861 seconds with return value 0
Press any key to continue . . .
```

Question # 12) Write a program to check whether a number is Armstrong number or not.

Source Code

```cpp
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    int number, original, remainder, result = 0, n = 0;
    cout << "Enter a number: ";
    cin >> number;
    original = number;
    while (original != 0) {
        original /= 10;
        n++;
    }
    original = number;
    while (original != 0) {
        remainder = original % 10;
        result += pow(remainder, n);
        original /= 10;
    }
    if (result == number)
        cout << number << " is an Armstrong number.";
    else
        cout << number << " is not an Armstrong number.";
    return 0;
}
```

Output

```
Enter a number: 36
36 is not an Armstrong number.
--------------------------------
Process exited after 3.922 seconds with return value 0
Press any key to continue . . .
```

Question # 13) Write a program to convert Binary to Decimal number system.

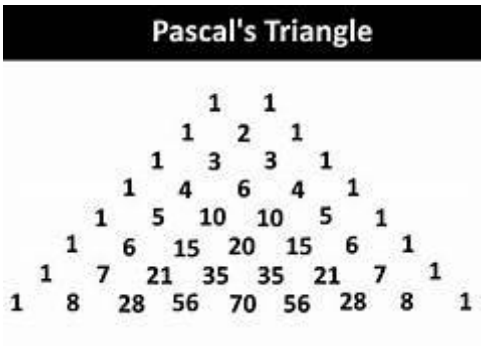
Source Code

```cpp
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    int binary, decimal = 0, base = 1, remainder;
    cout << "Enter a binary number: ";
    cin >> binary;
    while (binary > 0) {
        remainder = binary % 10;
        decimal += remainder * base;
        binary /= 10;
        base *= 2;
    }
    cout << "Decimal equivalent: " << decimal;
    return 0;
}
```

Output

```
Enter a binary number: 12
Decimal equivalent: 4
--------------------------------
Process exited after 14.11 seconds with return value 0
Press any key to continue . . .
```

Question # 14) Write a program to print Pascal triangle upto n rows.



Source Code

```cpp
#include <iostream>
using namespace std;
int main() {
    int rows, coef = 1;
    cout << "Enter number of rows: ";
    cin >> rows;
    for (int i = 0; i < rows; i++) {
        for (int space = 1; space <= rows - i; space++)
            cout << "  ";
        for (int j = 0; j <= i; j++) {
            if (j == 0 || i == 0)
                coef = 1;
            else
                coef = coef * (i - j + 1) / j;
            cout << coef << "   ";
        }
        cout << endl;
    }
    return 0;
}
```

Output

```
Enter number of rows: 5
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1

--------------------------------
Process exited after 3.875 seconds with return value 0
Press any key to continue . . .
```

Question # 15) Write a program to print following patterns by using nested loops.

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

Rhombus Star Pattern

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

Hollow Rhombus Star Pattern

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

Mirrored Rhombus Star Pattern

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

Hollow Mirrored Rhombus Star Pattern

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

Right Triangle Star Pattern

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

Hollow Right Triangle Star Pattern

Mirrored Right Triangle Star Pattern

Hollow Mirrored Right Triangle Star Pattern

Inverted Right Triangle Star Pattern

Hollow Inverted Right Triangle Star Pattern

Inverted Mirrored Right Triangle Star Pattern

Hollow Inverted Mirrored Right Triangle Star Pattern

Pyramid Star Pattern

Hollow Pyramid Star Pattern

Inverted Pyramid Star Pattern

Hollow Inverted Pyramid Star Pattern

Half Diamond Star Pattern

Mirrored Half Diamond Star Pattern

Diamond Star Pattern

Hollow Diamond Star Pattern

Right Arrow Star Pattern

Left Arrow Star Pattern

Plus Star Pattern

X Star Pattern

Eight Star Pattern

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

Heart Star Pattern

Source Code

```cpp
#include <iostream>
using namespace std;
// Function to print Rhombus Star Pattern
void printRhombus(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i; j++)
            cout << " ";
        for (int j = 0; j < n; j++)
            cout << "*";
        cout << endl;
    }
}
// Function to print Hollow Rhombus Star Pattern
void printHollowRhombus(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i; j++)
            cout << " ";
        for (int j = 0; j < n; j++) {
            if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Mirrored Rhombus Star Pattern
void printMirroredRhombus(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++)
            cout << " ";
        for (int j = 0; j < n; j++)
            cout << "*";
        cout << endl;
    }
}
// Function to print Hollow Mirrored Rhombus Star Pattern
void printHollowMirroredRhombus(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++)
            cout << " ";
        for (int j = 0; j < n; j++) {
            if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Right Triangle Star Pattern
void printRightTriangle(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
```

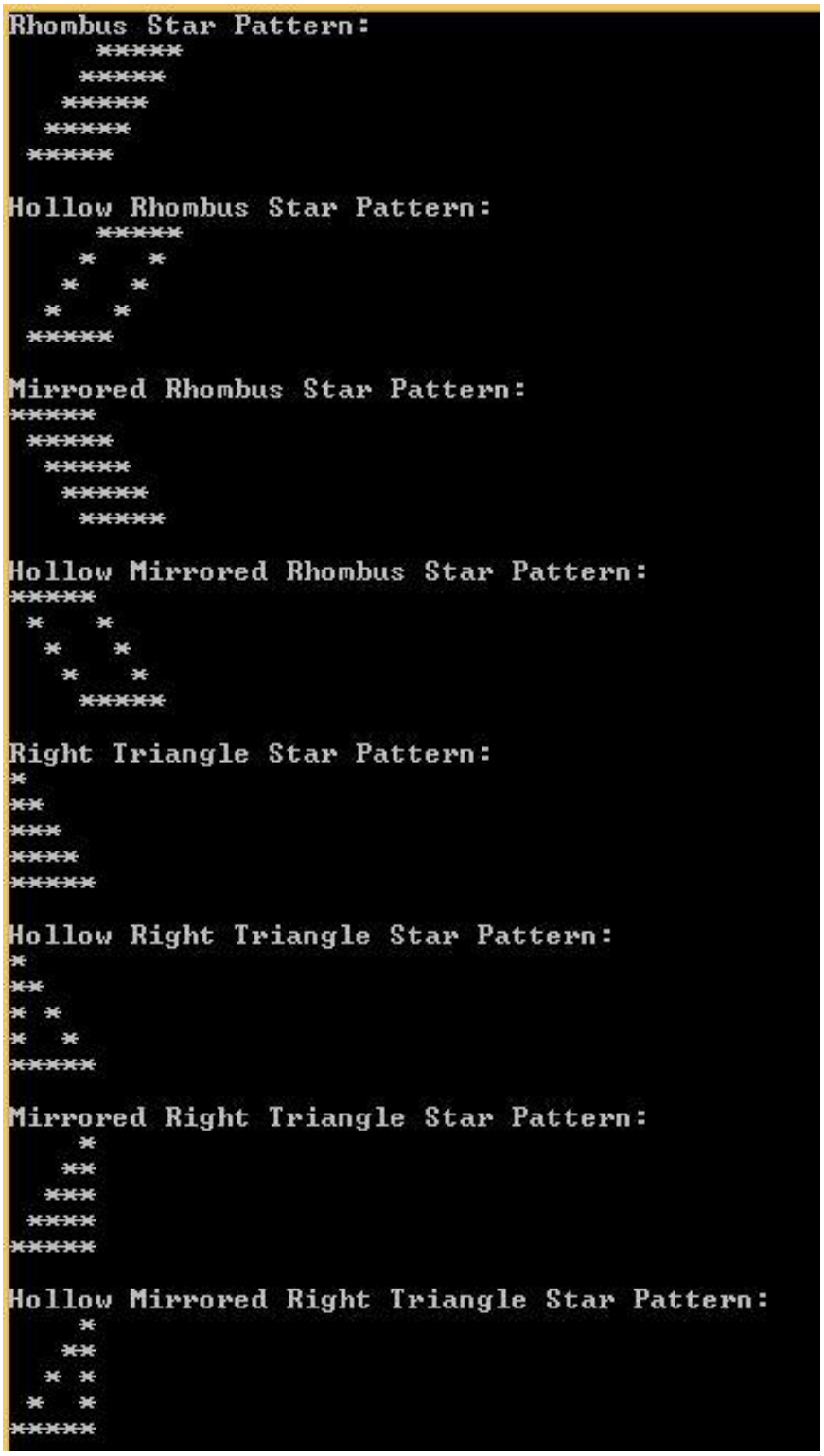
```cpp
// Function to print Hollow Right Triangle Star Pattern
void printHollowRightTriangle(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            if (j == 1 || j == i || i == n)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Mirrored Right Triangle Star Pattern
void printMirroredRightTriangle(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
// Function to print Hollow Mirrored Right Triangle Star Pattern
void printHollowMirroredRightTriangle(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int j = 1; j <= i; j++) {
            if (j == 1 || j == i || i == n)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Inverted Right Triangle Star Pattern
void printInvertedRightTriangle(int n) {
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
// Function to print Hollow Inverted Right Triangle Star Pattern
void printHollowInvertedRightTriangle(int n) {
    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;
    }
}
// Function to print Inverted Mirrored Right Triangle Star Pattern
void printInvertedMirroredRightTriangle(int n) {
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
```

```cpp
// Function to print Pyramid Star Pattern
void printPyramid(int n) {
    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;
    }
}
// Function to print Hollow Pyramid Star Pattern
void printHollowPyramid(int n) {
    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) || i == n)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Inverted Pyramid Star Pattern
void printInvertedPyramid(int n) {
    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;
    }
}
// Function to print Hollow Inverted Pyramid Star Pattern
void printHollowInvertedPyramid(int n) {
    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++) {
            if (j == 1 || j == (2 * i - 1) || i == n)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Half Diamond Star Pattern
void printHalfDiamond(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
    for (int i = n - 1; i >= 1; i--) {
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
```

```cpp
// Function to print Mirrored Half Diamond Star Pattern
void printMirroredHalfDiamond(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
    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 << "*";
        cout << endl;
    }
}
// Function to print Diamond Star Pattern
void printDiamond(int n) {
    printPyramid(n);
    printInvertedPyramid(n);
}
// Function to print Hollow Diamond Star Pattern
void printHollowDiamond(int n) {
    printHollowPyramid(n);
    printHollowInvertedPyramid(n);
}
// Function to print Right Arrow Star Pattern
void printRightArrow(int n) {
    printPyramid(n);
    printInvertedRightTriangle(n);
}
// Function to print Left Arrow Star Pattern
void printLeftArrow(int n) {
    printInvertedRightTriangle(n);
    printPyramid(n);
}
// Function to print Plus Star Pattern
void printPlus(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (j == n / 2 + 1 || i == n / 2 + 1)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print X Star Pattern
void printX(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (j == i || j == n - i + 1)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
```

```cpp
// Function to print Eight Star Pattern
void printEight(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (i == 1 || i == n || j == 1 || j == n || (i == j && i <= n / 2) || (i + j == n + 1 && i <= n / 2))
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}
// Function to print Heart Star Pattern
void printHeart(int n) {
    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;
    }
    for (int i = n; i >= 1; i--) {
        for (int j = i; j < n; j++)
            cout << " ";
        for (int j = 1; j <= (i * 2) - 1; j++)
            cout << "*";
        cout << endl;
    }
}
// Main function to execute the star patterns
int main() {
    int n = 5;
    cout << "Rhombus Star Pattern:\n";
    printRhombus(n);
    cout << "\nHollow Rhombus Star Pattern:\n";
    printHollowRhombus(n);
    cout << "\nMirrored Rhombus Star Pattern:\n";
    printMirroredRhombus(n);
    cout << "\nHollow Mirrored Rhombus Star Pattern:\n";
    printHollowMirroredRhombus(n);
    cout << "\nRight Triangle Star Pattern:\n";
    printRightTriangle(n);
    cout << "\nHollow Right Triangle Star Pattern:\n";
    printHollowRightTriangle(n);
    cout << "\nMirrored Right Triangle Star Pattern:\n";
    printMirroredRightTriangle(n);
    cout << "\nHollow Mirrored Right Triangle Star Pattern:\n";
    printHollowMirroredRightTriangle(n);
    cout << "\nInverted Right Triangle Star Pattern:\n";
    printInvertedRightTriangle(n);
    cout << "\nHollow Inverted Right Triangle Star Pattern:\n";
    printHollowInvertedRightTriangle(n);
    cout << "\nInverted Mirrored Right Triangle Star Pattern:\n";
    printInvertedMirroredRightTriangle(n);
    cout << "\nPyramid Star Pattern:\n";
    printPyramid(n);
    cout << "\nHollow Pyramid Star Pattern:\n";
    printHollowPyramid(n);
    cout << "\nInverted Pyramid Star Pattern:\n";
    printInvertedPyramid(n);
    cout << "\nHollow Inverted Pyramid Star Pattern:\n";
    printHollowInvertedPyramid(n);
    cout << "\nHalf Diamond Star Pattern:\n";
    printHalfDiamond(n);
    cout << "\nMirrored Half Diamond Star Pattern:\n";
    printMirroredHalfDiamond(n);
    cout << "\nDiamond Star Pattern:\n";
    printDiamond(n);
    cout << "\nHollow Diamond Star Pattern:\n";
    printHollowDiamond(n);
    cout << "\nRight Arrow Star Pattern:\n";
    printRightArrow(n);
    cout << "\nLeft Arrow Star Pattern:\n";
    printLeftArrow(n);
    cout << "\nPlus Star Pattern:\n";
    printPlus(n);
    cout << "\nX Star Pattern:\n";
    printX(n);
    cout << "\nEight Star Pattern:\n";
    printEight(n);
    cout << "\nHeart Star Pattern:\n";
    printHeart(n);
    return 0;
}
```

Output


Rhombus Star Pattern:
*****
*****
*****
*****
*****
Hollow Rhombus Star Pattern:
*****
* *
* *
* *
*****
Mirrored Rhombus Star Pattern:
*****
*****
*****
*****
*****
Hollow Mirrored Rhombus Star Pattern:
*****
* *
* *
* *
*****
Right Triangle Star Pattern:
*
**
***
****
*****
Hollow Right Triangle Star Pattern:
*
**
* *
* *
*****
Mirrored Right Triangle Star Pattern:
*
**
***
****
*****
Hollow Mirrored Right Triangle Star Pattern:
*
**
* *
* *
*****

```
Inverted Right Triangle Star Pattern:
*****
****
***
**
*

Hollow Inverted Right Triangle Star Pattern:
*****
*  *
* *
**
*

Inverted Mirrored Right Triangle Star Pattern:
*****
 ****
  ***
   **
    *

Pyramid Star Pattern:
    *
   ***
  *****
 *******
*********

Hollow Pyramid Star Pattern:
    *
   * *
  *   *
 *     *
*********

Inverted Pyramid Star Pattern:
*********
 *******
  *****
   ***
    *

Hollow Inverted Pyramid Star Pattern:
*********
 *     *
  *   *
   * *
    *

Half Diamond Star Pattern:
*
**
***
****
*****
****
***
**
*

Mirrored Half Diamond Star Pattern:
    *
   **
  ***
 ****
*****
 ****
  ***
   **
    *

Diamond Star Pattern:
    *
   ***
  *****
 *******
*********
*********
 *******
  *****
   ***
    *

Hollow Diamond Star Pattern:
    *
   * *
  *   *
 *     *
*********
*********
 *     *
  *   *
   * *
    *
```


Right Arrow Star Pattern:
*
***
*****
****
****
*****
****
***
**
*
Left Arrow Star Pattern:
*****
****
***
**
*
*
***
*****
****
****
Plus Star Pattern:
*
*
*****
*
*
X Star Pattern:
*   *
* *
*
* *
*   *
Eight Star Pattern:
*****
** **
*   *
*   *
*****
Heart Star Pattern:
**   **
**** ****
****
****
*****
***
*

Process exited after 0.2887 seconds with return value 0
Press any key to continue . . .