Pattern Printing

1.Square Pattern

CODE:

```cpp
#include <bits/stdc++.h>
using namespace std;

void pattern1(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n;j++){
            cout<<"* ";
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern1(n);
    return 0;
}
```

2. Right Angled Triangle Pattern

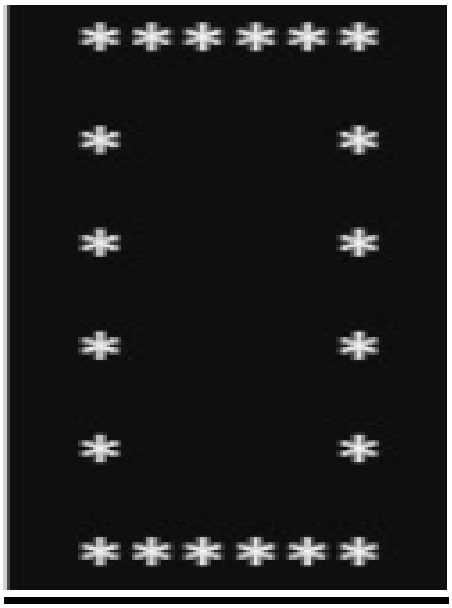
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern2(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){
            cout<<"* ";
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern2(n);
    return 0;
}
```

3. Hollow Square Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern3(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){
                cout<<"*";
            }
            else {
                cout<<" ";
            }
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern3(n);
    return 0;
}
```

4. Hollow Square Pattern with Diagonal

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern4(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 || j==(n-i+1) || i==j ){
            cout<<"*";
        }
        else {
                cout<<" ";
           }
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern4(n);
    return 0;
}
```

5. Hollow Right Angled Triangle Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern5(int n){
 for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++){
        if( i==n || j==1 || i==j ){
            cout<<"*";
        }
        else {
            cout<<" ";
          }
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern5(n);
    return 0;
}
```

6. Mirrored Right Angle Triangle Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern6(int n){
 for(int i=1;i<=n;i++){
    for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        for(int k=1;k<=i;k++){
            cout<<"*";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern6(n);
    return 0;
}
```

7. Mirrored Hollow Right Angle Triangle Pattern

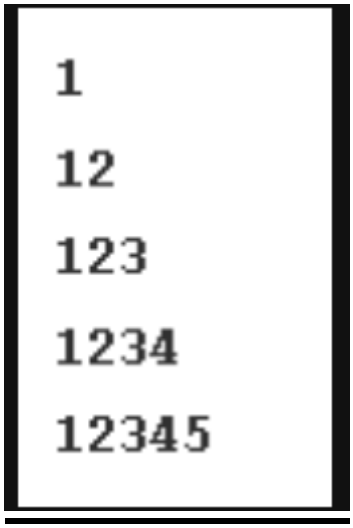
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern12(int n){
    int m=1;
    for(int i=n; i>=1; i--) {
        for(int j=1; j<=i-1; j++) {
            cout << " ";
        }
        for(int k=1; k<=m; k++) {
            if(k==1 || k==m || m==n)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
        m++;
    }
}

int main(){
    int n;
    cin>>n;
    pattern12(n);
    return 0;
}
```

8. Number Pyramid Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern8(int n){
 for(int i=1;i<=n;i++){
    for(int j=1;j<=i;j++){
            cout<<j<<" ";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern8(n);
    return 0;
}
```

9. Inverted Pyramid Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern9(int n){
 for(int i=1;i<=n;i++){
    for(int j=1;j<=n-i+1;j++){
            cout<<"* ";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern9(n);
    return 0;
}
```

10. F;ipped Inverted Pyramid Pattern

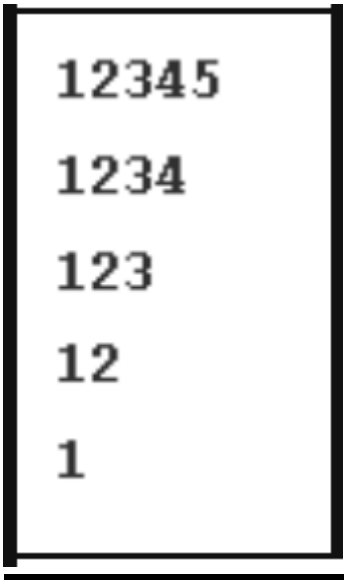
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern10(int n){
 for(int i=n;i>=1;i--){
    for(int j=1;j<=i;j++){
            cout<<"* ";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern10(n);
    return 0;
}
```

11. Inverted Number Pyramid Pattern



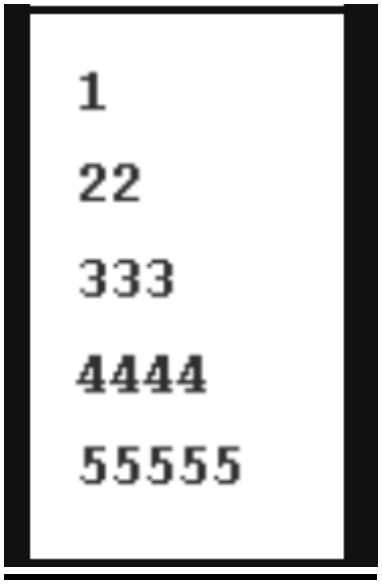
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern11(int n){
 for(int i=1;i<=n;i++){
    for(int j=1;j<=n-i+1;j++){
            cout<<j<<" ";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern11(n);
    return 0;
}
```

12. Repeated Number Pyramid Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern12(int n){
    int c=0;
 for(int i=1;i<=n;i++){
     c=i;
    for(int j=1;j<=i;j++){

            cout<<c<<" ";
        }
    cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern12(n);
    return 0;
}
```

13. Half Diamond Star Pattern

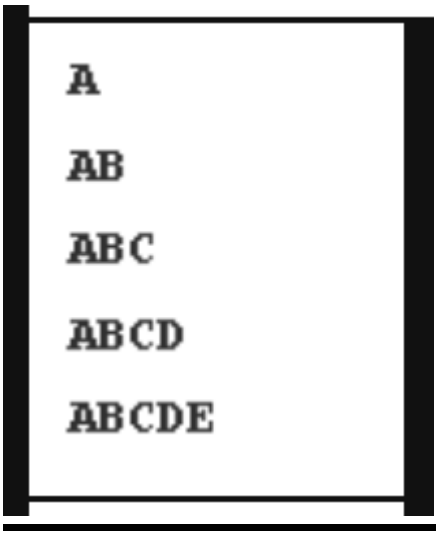
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern13(int n){
    for(int i=1;i<=(2*n)-1;i++){
        int c=0;
        if(i<=n){
            c=i;
        }
        else{
            c=(2*n) - i;
        }
    for(int j=1;j<=c;j++){
        cout<<"* ";
        }
     cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern13(n);
    return 0;
}
```

14. Increasing Letter Triangle Pattern



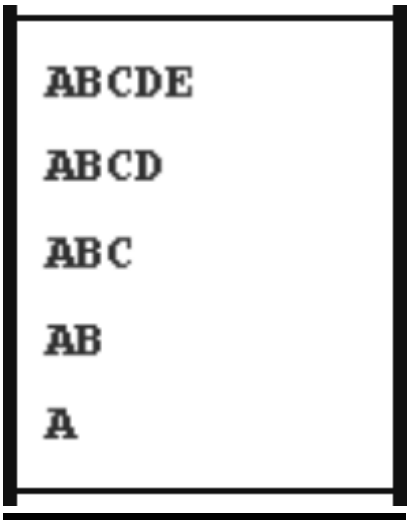
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern14(int n){
    for(int i=1;i<=n;i++){
        for(char ch='A';ch<='A'+i-1;ch++){
            cout<<ch<<" ";
        }
     cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern14(n);
    return 0;
}
```

15. Reverse Letter Triangle Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern15(int n){
    for(int i=1;i<=n;i++){
        for(char ch='A';ch<='A'+(n-i);ch++){
            cout<<ch<<" ";
        }
     cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern15(n);
    return 0;
}
```

16. Star Pyramid Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern16(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<<"*";
        }
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        cout<<endl;
    }

}

int main(){
    int n;
    cin>>n;
    pattern16(n);

    return 0;
}
```

17. Inverted Star Pyramid Pattern

CODE:

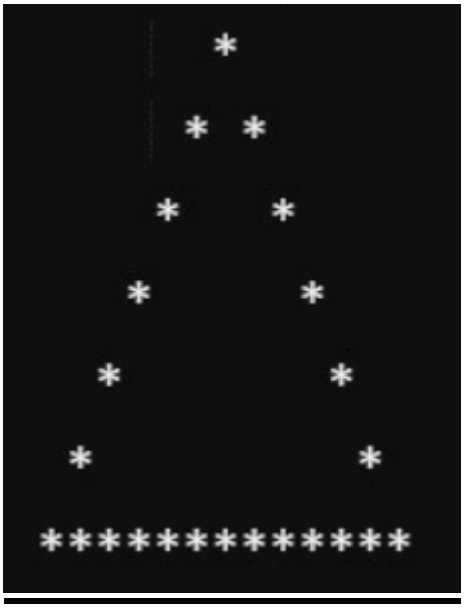
```
#include <bits/stdc++.h>
using namespace std;

void pattern17(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<<"*";
        }
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        cout<<endl;
    }

}

int main(){
    int n;
    cin>>n;
    pattern17(n);
    return 0;
}
```

18. Hollow Star Pyramid Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern18(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;
    }

}

int main(){
    int n;
    cin>>n;
    pattern18(n);
    return 0;
}
```

19. Inverted Hollow Star Pyramid Pattern

CODE:

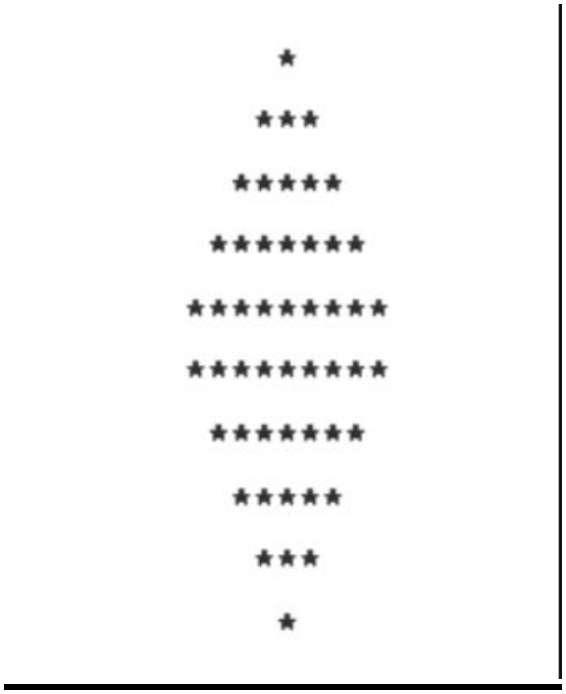
```
#include <bits/stdc++.h>
using namespace std;

void pattern19(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;
    }

}

int main(){
    int n;
    cin>>n;
    pattern19(n);
    return 0;
}
```

20. Diamond Star Pattern



CODE:

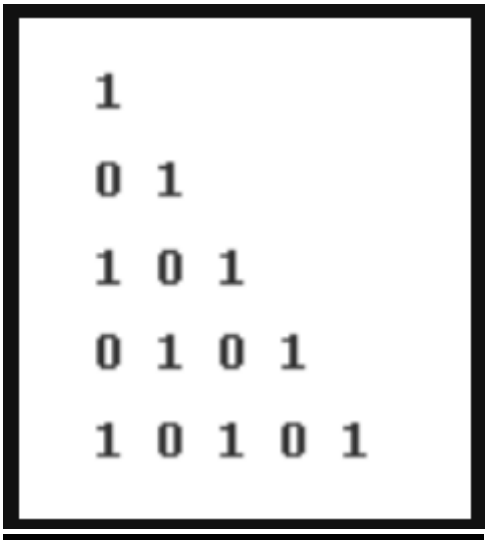
```
#include <bits/stdc++.h>
using namespace std;

void pattern20(int n){
      for(int i=1;i<=n/2;i++){
          for(int j=1;j<=n-i;j++){
              cout<<" ";
          }
          for(int j=1;j<=(2*i)-1;j++){
              cout<<"*";
              }
            for(int j=1;j<=(2*i)-1;j++){
              cout<<" ";
              }
          cout<<endl;
      }

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

int main(){
      int n;
      cin>>n;
      pattern20(n);
      return 0;
}
```

21. Binary Number Triangle Pattern



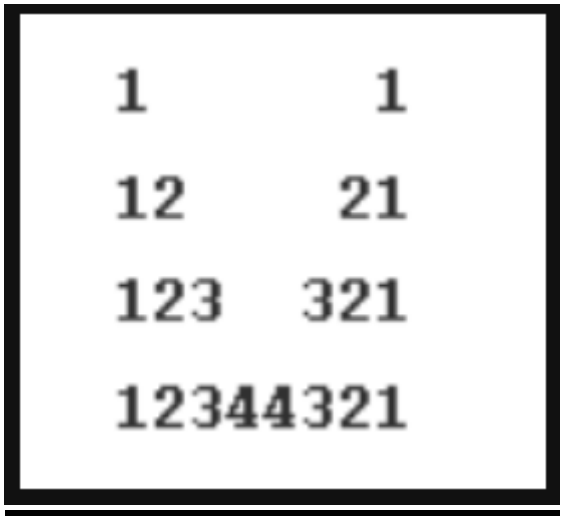
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern21(int n){
      int start=1;
      for(int i=1;i<=n;i++){
          if(i%2==0){
              start=0;
          }
          else{
              start=1;
          }
      for(int j=1;j<=i;j++){
          cout<<start<<" ";
          start=1-start;
      }
      cout<<endl;
      }
}

int main(){
    int n;
    cin>>n;
    pattern21(n);
    return 0;
}
```

22.Number Crown Pattern



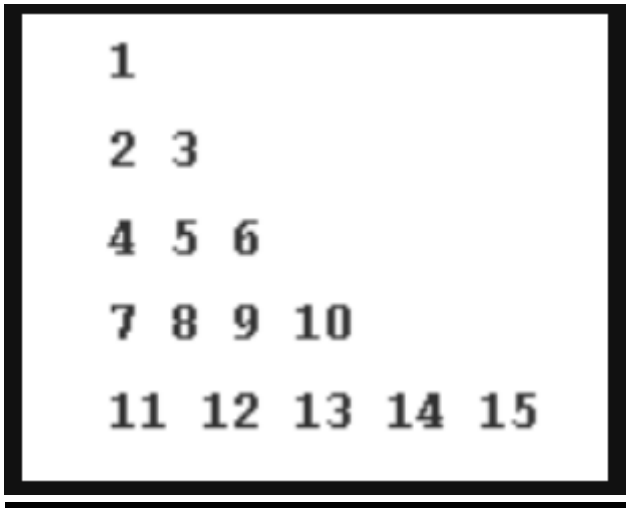
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern22(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){
            cout<<j;
        }
         for(int j=1;j<=2*(n-i);j++){
            cout<<" ";
        }
         for(int j=i;j>=1;j--){
            cout<<j;
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern22(n);
    return 0;
}
```

23. Increasing Number Triangle Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern23(int n){
    int start=1;
     for(int i=1;i<=n;i++){
         for(int j=1;j<=i;j++){
             cout<<start<<" ";
             start=start+1;
         }

         cout<<endl;
     }
}

int main(){
    int n;
    cin>>n;
    pattern23(n);
    return 0;
}
```

24. Rhombus Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern24(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        for(int j=1;j<=n;j++){
            cout<<"*";
        }
        for(int j=1;j<=i-1;j++){
            cout<<" ";
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern24(n);
    return 0;
}
```

25.Hollow Rhombus Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern25(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        for(int j=1;j<=n;j++){
            if(i==1 || i==n || j==1 || j==n){
                cout<<"*";
            }
            else{
                cout<<" ";
            }
        }
        for(int j=1;j<=i-1;j++){
            cout<<" ";
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern25(n);
    return 0;
}
```

26. Mirrored Rhombus Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern26(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i-1;j++){
            cout<<" ";
        }
        for(int j=1;j<=n;j++){
            cout<<"*";
        }
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
         cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern26(n);
    return 0;
}
```

27. Hollow Mirrored Rhombus Pattern

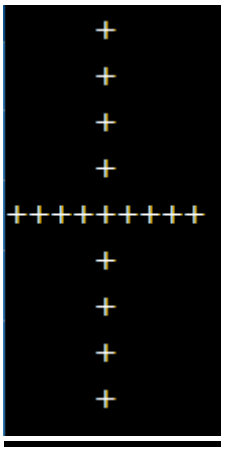
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern27(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i-1;j++){
            cout<<" ";
        }
        for(int j=1;j<=n;j++){
            if(i==1 || i==n || j==1 || j==n)
            cout<<"*";
            else
            cout<<" ";
        }
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
         cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern27(n);
    return 0;
}
```

28. Plus Pattern



CODE:

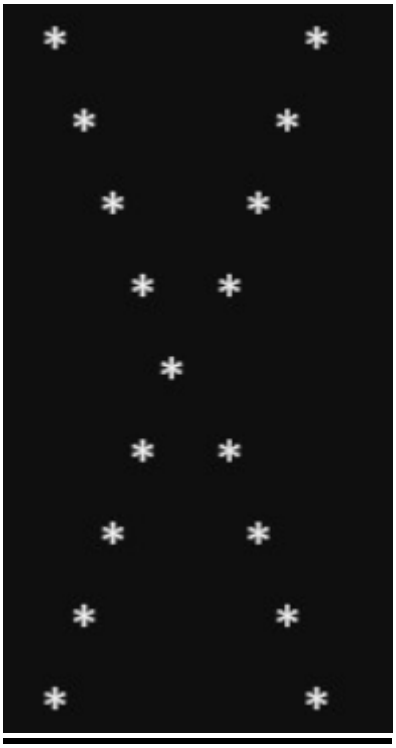
```
#include <bits/stdc++.h>
using namespace std;

void pattern28(int n){
   for(int i=1;i<=(2*n)-1;i++){
       if(i==(((2*n)-1)/2)+1){
           for(int j=1;j<=(2*n)-1;j++){
               cout<<"+";
           }
       }
       else{
            for(int j=1;j<=(2*n)-1;j++){
                if(j==(((2*n)-1)/2)+1){
                    cout<<"+";
                }
                else{
                    cout<<" ";
                }
           }
       }

        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern28(n);
    return 0;
}
```

29. Cross Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern29(int n){
    for(int i=1;i<=(2*n)-1;i++){
        for(int j=1;j<=(2*n)-1;j++){
            if(i==j || j==2*n-i)
            cout<<"*";
            else
            cout<<" ";
        }
         cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern29(n);
    return 0;
}
```

30. Alpha-Ramp Pattern

CODE:

```cpp
#include <bits/stdc++.h>
using namespace std;

void pattern30(int n){
    for(int i=1;i<=n;i++){
        char start='A'+i-1;
        for(int j=1;j<=i;j++){
            cout<<start;
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern30(n);
    return 0;
}
```

31. Alpha-Hill Pattern

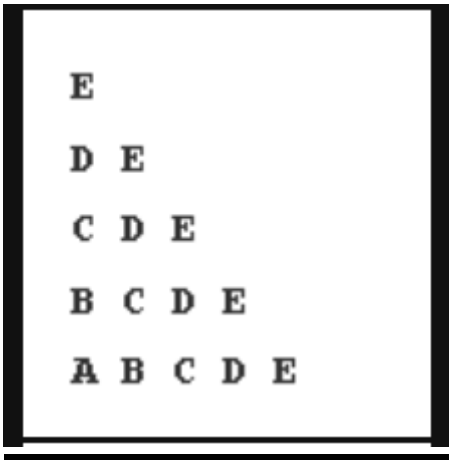
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern31(int n){
    for(int i=1;i<=n;i++){
        char start='A';
        for(int j=1;j<=(n-i);j++){
            cout<<" ";
        }
        int breakpoint=((2*i)-1)/2;
        for(int j=1;j<=(2*i)-1;j++){
            cout<<start;
            if(j>breakpoint){
                start--;
            }
            else{
                start++;
            }
        }
        for(int j=1;j<=(n-i);j++){
            cout<<" ";
        }
         cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern31(n);
    return 0;
}
```

32. Alpha-Triangle Pattern



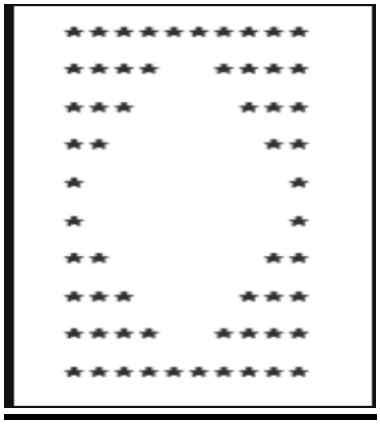
CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern32(int n){
   for(int i=1;i<=n;i++){
      for(char ch='E'-i+1;ch<='E';ch++){
          cout<<ch<<" ";
      }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern32(n);
    return 0;
}
```

33. Symmetric-Void Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern33(int n){
      for(int i=1; i<=n; i++) {
        for(int j=1;j<=n-i+1;j++){
            cout<<"*";
        }
        for(int j=1;j<=(2*i)-2;j++){
            cout<<" ";
        }
         for(int j=1;j<=n-i+1;j++){
            cout<<"*";
        }
        cout<<endl;
   }
    for(int i=1; i<=n; i++) {
        for(int j=1;j<=i;j++){
            cout<<"*";
        }
       for(int j=1;j<=(2*n)-(2*i);j++){
            cout<<" ";
        }
         for(int j=1;j<=i;j++){
            cout<<"*";
        }
        cout<<endl;
   }
}

int main(){
    int n;
    cin>>n;
    pattern33(n);
    return 0;
}
```

34. Symmetric-Butterfly Pattern

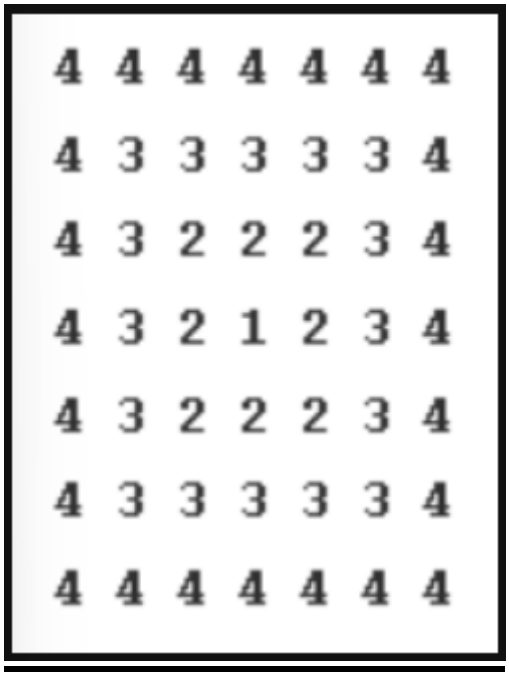
CODE:

```cpp
#include <bits/stdc++.h>
using namespace std;

void pattern34(int n) {
    int spaces = 2 * n - 2;
    for (int i = 1; i <= 2 * n - 1; i++) {
        int stars = i;
        if (i > n) {
            stars = 2 * n - i;
        }
        for (int j = 1; j <= stars; j++) {
            cout << "*";
        }
        for (int j = 1; j <= spaces; j++) {
            cout << " ";
        }
        for (int j = 1; j <= stars; j++) {
            cout << "*";
        }
        cout << endl;
        if (i < n) spaces -= 2;
        else spaces += 2;
    }
}

int main(){
    int n;
    cin>>n;
    pattern34(n);
    return 0;
}
```

35. The Number Pattern



CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern35(int n) {
    for (int i= 1; i<= 2*n - 1;i++) {
        for (int j= 1;j<= 2*n -1;j++) {
            int top= i-1;
            int bottom= j-1;
            int right= (2*n-2) - (j-1);
            int left= (2*n-2) - (i-1);
            cout<<(n - min(min(top, bottom), min(left, right)))<< " ";
        }
        cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern35(n);
    return 0;

}
```

36. Right Arrow Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern36(int n){
      for(int i=0; i<n; i++) {
        for(int j=0; j<i; j++) {
            cout << " ";
        }
        for(int k=1; k<=n-i; k++) {
            cout << "*";
        }
        cout << endl;
    }
    for(int i=1; i<n; i++) {
        for(int j=1; j<n-i; j++) {
            cout << " ";
        }
        for(int k=1; k<=i+1; k++) {
            cout << "*";
        }
        cout<<endl;
   }
}

int main(){
    int n;
    cin>>n;
    pattern36(n);
    return 0;
}
```

37. Left Arrow Pattern

CODE:

```
#include <bits/stdc++.h>
using namespace std;

void pattern37(int n){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n-i;j++){
            cout<<" ";
        }
        for(int j=0;j<=n-i;j++){
            cout<<"*";
        }
        cout<<endl;
    }
    for(int i=1;i<n;i++){
        for(int j=1;j<i+1;j++){
            cout<<" ";
        }
       for(int j=1;j<=i+1;j++){
            cout<<"*";
        }
         cout<<endl;
    }
}

int main(){
    int n;
    cin>>n;
    pattern37(n);
    return 0;
}
```