261102
Computer Programming

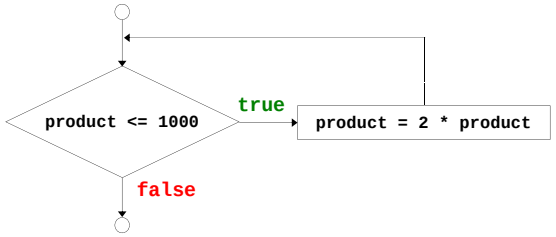
Lecture 4: Repetition Structures

while Repetition Structure

- Action repeated while some condition remains true
- Pseudocode example:

  while the product is still not over 1000
  Double the product

- Flowchart example:

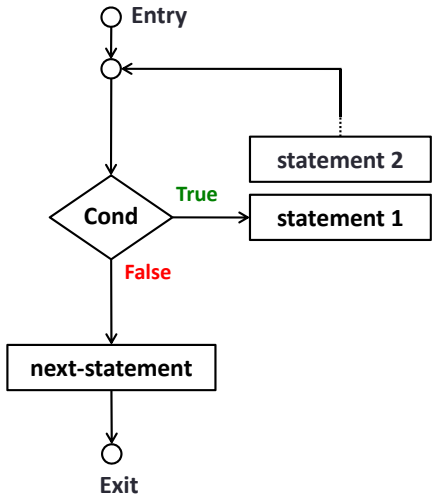


- Translation into C++

```
int product = 2;
while ( product <= 1000 )
   product = 2 * product;
```


while Statement

```
while (condition) {
  statement1;
  statement2;
  ...
}
next-statement;
```



- While the `condition` is `true`
  - `statement1, statement2, …` are repeated again and again
- Until the `condition` is `false`
  - `statement1, statement2, …` are ignored and program continues to `next-statement`

while Statement

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x = 1;
7      while(x < 5){
8          cout << "Print " << x << endl;
9          x++;
10     }
11     return 0;
```

```
Print 1
Print 2
Print 3
Print 4
```

```
7      while(x < 5){
8          cout << "Print " << x++ << endl;
9      }
```

while Statement

What is the output?

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

int main()
{
    int x = 5;
    while(x < 50){
        cout << "Print " << x << endl;
        x *= 2;
    }
    return 0;
}
```


while Statement

What is the output?

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

int main()
{
    int x = 5;
    while(x%3){
        cout << "Print " << x << endl;
        x += 5;
    }
    return 0;
}
```


Example 4-A: Find Average

ver. Counter-Controlled Repetition

```
Enter number [1]: 1
Enter number [2]: 2
Enter number [3]: 3
Enter number [4]: 1
Enter number [5]: 2
Enter number [6]: 3
Enter number [7]: 1
Enter number [8]: 2
Enter number [9]: 3
Enter number [10]: 4
Average = 2.2
```

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

int main()
{
    int num, sum = 0, count = 1;
    while(count <= 10){
        cout << "Enter number [" << count << "]: ";
        cin >> num;
        sum += num;
        count++;
    }
    cout << "Average = " << sum/10.0;
    return 0;
}
```


Example 4-A: Find Average

ver. Counter-Controlled Repetition

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

int main()
{
    int num, sum = 0, count = 1;
    while(count <= 10){
        cout << "Enter number [" << count << "]: ";
        cin >> num;
        sum += num;
        count++;
    }
    cout << "Average = " << sum/count;
    return 0;
}
```

What is the output?

Example 4-A: Find Average

ver. Counter-Controlled Repetition

```
Enter N: 4
Enter number [1]: 10
Enter number [2]: 20
Enter number [3]: 30
Enter number [4]: 40
Average = 25
```

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

int main()
{
    int N, num, sum = 0, count = 1;
    cout << "Enter N: ";
    cin >> N;
    while(count <= N){
        cout << "Enter number [" << count << "]: ";
        cin >> num;
        sum += num;
        count++;
    }
    cout << "Average = " << (double) sum/(count-1);
    return 0;
}
```


Example 4-A: Find Average

ver. Sentinel-Controlled Repetition

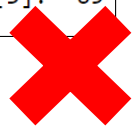
```
Enter -69 to finish.
Enter number [1]: 10
Enter number [2]: 20
Enter number [3]: 30
Enter number [4]: 40
Enter number [5]: -69
Average = 25
```

Sentinel value
- Indicates "*end of data entry*"
- Loop ends when sentinel input
- Sentinel chosen so it cannot be confused with regular input (-69 in this case)

Example 4-A: Find Average

ver. Sentinel-Controlled Repetition

```
Enter -69 to finish.
Enter number [1]: 10
Enter number [2]: 20
Enter number [3]: 30
Enter number [4]: 40
Enter number [5]: -69
Average = 6.2
```



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

int main()
{
    int num = 0, sum = 0, count = 1;
    cout << "Enter -69 to finish.\n";
    while(num != -69){
        cout << "Enter number [" << count << "]: ";
        cin >> num;
        sum += num;
        count++;
    }
    cout << "Average = " << (double) sum/(count-1);
    return 0;
}
```

```
14      cout << "Average = " << (double) (sum+69)/(count-2);
```


Example 4-A: Find Average

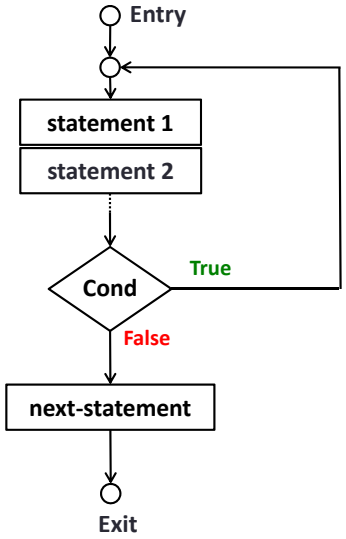
ver. Sentinel-Controlled Repetition

```
Enter -69 to finish.
Enter number [1]: 10
Enter number [2]: 20
Enter number [3]: 30
Enter number [4]: 40
Enter number [5]: -69
Average = 25
```

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

int main()
{
    int num, sum = 0, count = 1;
    cout << "Enter -69 to finish.\n";
    cout << "Enter number [" << count << "]: ";
    cin >> num;
    while(num != -69){
        sum += num;
        count++;
        cout << "Enter number [" << count << "]: ";
        cin >> num;
    }
    cout << "Average = " << (double) sum/(count-1);
    return 0;
}
```

do-while Statement

```
do {
  statement1;
  statement2;
  ...
} while (condition);
next-statement;
```



- Execute **statement1, statement2, …** one time before checking **condition**

Example 4-B: Nuclear Launch Code

```
Please input the nuclear launch code:i love u
Please input the nuclear launch code:123456789
Please input the nuclear launch code:6969
Nuclear weapons are launched!!!
บะคู้มมมมม...
```

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string input, password = "6969";
8      cout << "Please input the nuclear launch code:";
9      getline(cin,input);
10     while(input != password){
11         cout << "Please input the nuclear launch code:";
12         getline(cin,input);
13     }
14     cout << "\a\a\aNuclear weapons are launched!!!\n";
15     cout << "บะคู้มมมมม...";
16     return 0;
17 }
```


Example 4-B: Nuclear Launch Code

```
Please input the nuclear launch code:i love u
Please input the nuclear launch code:123456789
Please input the nuclear launch code:6969
Nuclear weapons are launched!!!
บะคู้มมมมม...
```

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string input, password = "6969";
8      do{
9          cout << "Please input the nuclear launch code:";
10         getline(cin,input);
11     }while(input != password);
12     cout << "\a\a\aNuclear weapons are launched!!!\n";
13     cout << "บะคู้มมมมม...";
14     return 0;
15 }
```

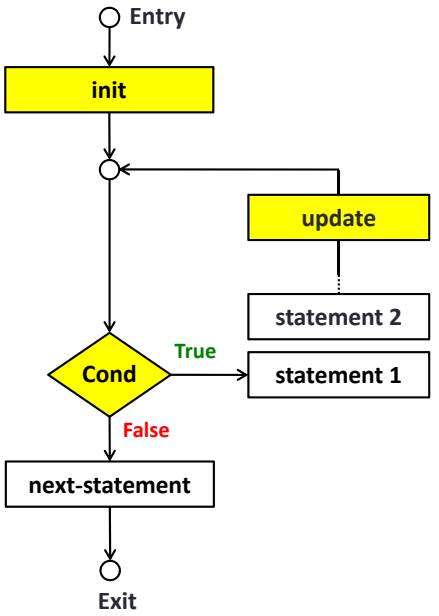

for Statement

Suitable for Counter-Controlled Repetition

```
for (init; cond; update) {
  statement1;
  statement2;
  ...
}
next-statement;
```

Equivalent

```
init;
while (cond) {
  statement1;
  statement2;
  ...
  update;
}
next-statement;
```



Example 4-A: Find Average

ver. Counter-Controlled Repetition

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, sum = 0, count = 1;
7      while(count <= 10){
8          cout << "Enter number [" << count << "]: ";
9          cin >> num;
10         sum += num;
11         count++;
12     }
13     cout << "Average = " << sum/10.0;
14     return 0;
15 }
```

Condition

Initialize

Update

Example 4-A: Find Average

ver. Counter-Controlled Repetition

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num, sum = 0;
7      for(int count = 1;count <= 10;count++){
8          cout << "Enter number [" << count << "]: ";
9          cin >> num;
10         sum += num;
11     }
12     cout << "Average = " << sum/10.0;
13     return 0;
14 }
```


for Statement

a) Vary the control variable from 1 to 100 in increments of 1.

```
for ( int i = 1; i <= 100; ++i )
```

b) Vary the control variable from 100 down to 1 in decrements of 1.

```
for ( int i = 100; i >= 1; --i )
```

c) Vary the control variable from 7 to 77 in steps of 7.

```
for ( int i = 7; i <= 77; i += 7 )
```

d) Vary the control variable from 20 down to 2 in steps of -2.

```
for ( int i = 20; i >= 2; i -= 2 )
```

e) Vary the control variable over the following sequence of values: 2, 5, 8, 11, 14, 17.

```
for ( int i = 2; i <= 17; i += 3 )
```

f) Vary the control variable over the following sequence of values: 99, 88, 77, 66, 55.

```
for ( int i = 99; i >= 55; i -= 11 )
```


for Statement

What is the output?

```
for(int i = 1;i <= 5;i++){
    cout << "Yo!!!\n";
}
```

```
for(int i = 0;i < 3;i++){
    cout << "Yo!!!\n";
}
```

```
for(int i = 10;i < 0;i--){
    cout << "Yo!!!\n";
}
```

for Statement

What is the output?

```
for(int i = 4;i >= 0;i--){
    cout << "Yo!!!\n";
}
```

```
for(int i = 0;i < 7;i+=2){
    cout << "Yo!!!\n";
}
```

```
for(int i = 0;!(i%2);i+=2){
    cout << "Yo!!!\n";
}
```


for Statement

```
 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
26 27 28 29 30
31 32 33 34 35
36 37 38 39 40
41 42 43 44 45
46 47 48 49 50
51 52 53 54 55
56 57 58 59 60
61 62 63 64 65
66 67 68 69 70
71 72 73 74 75
76 77 78 79 80
81 82 83 84 85
86 87 88 89 90
91 92 93 94 95
96 97 98 99 100
```

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

int main()
{
    for(int i = 1;i <= 100 ;i++){
        if(i <= 9) cout << " ";
        cout << i;
        if(i%5 == 0) cout << "\n";
        else cout << " ";
    }
    return 0;
}
```


Example 4-C: Find Max

```
Enter number [1]: 2
Enter number [2]: 25
Enter number [3]: 0
Enter number [4]: 43
Enter number [5]: 7
Enter number [6]: 5
Enter number [7]: 6
Enter number [8]: 12
Enter number [9]: 24
Enter number [10]: 32
Maximum location = 4
Maximum value = 43
```

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

int main()
{
    int num, max, loc;
    for(int i = 1; i <= 10; i++){
        cout << "Enter number [" << i << "]: ";
        cin >> num;
        if(i == 1 || num > max){
            max = num;
            loc = i;
        }
    }
    cout << "Maximum location = " << loc << '\n';
    cout << "Maximum value = " << max;
    return 0;
}
```


Example 4-C: Find Max

```
Enter number [1]: 20
Enter number [2]: 30
Enter number [3]: 40
Enter number [4]: 22
Enter number [5]: 30
Enter number [6]: 40
Enter number [7]: 12
Enter number [8]: 40
Enter number [9]: 0
Enter number [10]: 7
Maximum location = ?
Maximum value = ?
```

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

int main()
{
    int num, max, loc;
    for(int i = 1; i <= 10; i++){
        cout << "Enter number [" << i << "]: ";
        cin >> num;
        if(i == 1 || num > max){
            max = num;
            loc = i;
        }
    }
    cout << "Maximum location = " << loc << '\n';
    cout << "Maximum value = " << max;
    return 0;
}
```

break Statement

break statement is used within

- loops (**while**, **do-while**, **for**)
  - Escape early from a loop
- **switch**
  - Skip the remainder of **switch**

When **break** is executed, program will continue with **next-statement** after that structure.

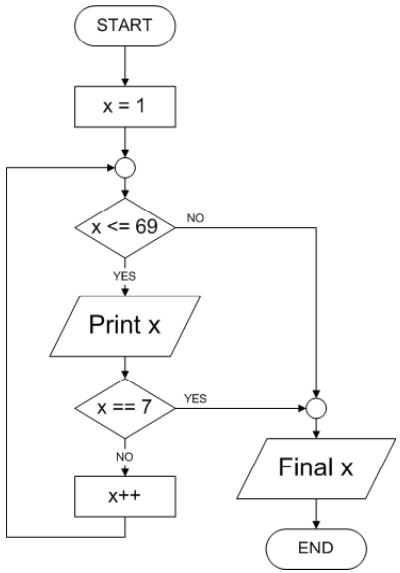
break Statement

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

int main()
{
    int x = 1;
    while(x <= 69){
        cout << "Print " << x << endl;
        if(x == 7) break;
        x++;
    }
    cout << "Final x = " << x << endl;
    return 0;
}
```

```
Print 1
Print 2
Print 3
Print 4
Print 5
Print 6
Print 7
Final x = 7
```


break Statement

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

int main()
{
    int x = 1;
    while(x <= 69){
        cout << "Print " << x << endl;
        if(x == 7) break;
        x++;
    }
    cout << "Final x = " << x << endl;
    return 0;
}
```



break Statement

```cpp
int main () {

    while(1) {

        ...

        if (cond)
            break;

        ...

    }

    return 0;
}
```

```cpp
int main () {

    do {

        ...

        if (cond)
            break;

        ...

    } while(1);

    return 0;
}
```

```cpp
int main () {

    for (;;) {

        ...

        if (cond)
            break;

        ...

    }

    return 0;
}
```

Infinity loop with **break** statement

break Statement

```
5  int main()
6  {
7      string input, password = "6969";
8      do{
9          cout << "Please input the nuclear launch code:";
10         getline(cin,input);
11     }while(input != password);
12     cout << "\a\a\aNuclear weapons are launched!!!\n";
13     cout << "บะคุ้มมมมม...";
14     return 0;
15 }
```

Equivalent

```
5  int main()
6  {
7      string input, password = "6969";
8      do{
9          cout << "Please input the nuclear launch code:";
10         getline(cin,input);
11         if(input == password) break;
12     }while(true);
13     cout << "\a\a\aNuclear weapons are lauched!!!\n";
14     cout << "บะคุ้มมมมม...";
15     return 0;
16 }
```


Example 4-B: Nuclear Launch Code

```
5  int main()
6  {
7      int count = 1;
8      string input, password = "6969";
9      do{
10         if(count >= 4){
11             cout << "Access denied. .... System is locked.\n";
12             break;
13         }
14         cout << "Please input the nuclear launch code:";
15         getline(cin,input);
16         if(input == password){
17             cout << "\a\a\aNuclear weapons are lauched!!!\n";
18             cout << "บะคุ้มมมมม...";
19             break;
20         }
21         count++;
22     }while(true);
23
24     return 0;
```

```
Please input the nuclear launch code:1
Please input the nuclear launch code:2
Please input the nuclear launch code:3
Access denied. .... System is locked.
```

```
Please input the nuclear launch code:I love you
Please input the nuclear launch code:6969
Nuclear weapons are lauched!!!
บะคุ้มมมมม...
```


continue Statement

- **`continue`** statement
  - Used in **`while`**, **`for`**, **`do/while`**
  - Skips remainder of loop body
  - Proceeds with next iteration of loop
- **`while`** and **`do/while`** structure
  - Loop-continuation **`condition`** evaluated immediately after the **`continue`** statement
- **`for`** structure
  - **`update`** expression executed
  - Then, loop-continuation **`condition`** evaluated

continue Statement

while

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x = 0;
7      while(x <= 3){
8          x++;
9          cout << "Print " << x << '\n';
10         cout << "Print " << 2*x << '\n';
11         cout << "Print " << 3*x << '\n';
12         cout << "------------------------\n";
13     }
14     return 0;
15 }
```

```
Print 1
Print 2
Print 3
------------------------
Print 2
Print 4
Print 6
------------------------
Print 3
Print 6
Print 9
------------------------
Print 4
Print 8
Print 12
------------------------
```

continue Statement

while

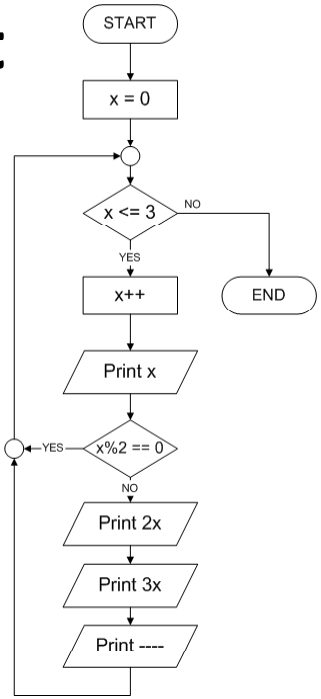
```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x = 0;
7      while(x <= 3){
8          x++;
9          cout << "Print " << x << '\n';
10         if(x%2 == 0) continue;
11         cout << "Print " << 2*x << '\n';
12         cout << "Print " << 3*x << '\n';
13         cout << "------------------------\n";
14     }
15     return 0;
16 }
```

```
Print 1
Print 2
Print 3
------------------------
Print 2
Print 3
Print 6
Print 9
------------------------
Print 4
```


continue Statement

while

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x = 0;
7      while(x <= 3){
8          x++;
9          cout << "Print " << x << '\n';
10         if(x%2 == 0) continue;
11         cout << "Print " << 2*x << '\n';
12         cout << "Print " << 3*x << '\n';
13         cout << "------------------------\n";
14     }
15     return 0;
16 }
```



continue Statement

while

What is the output?

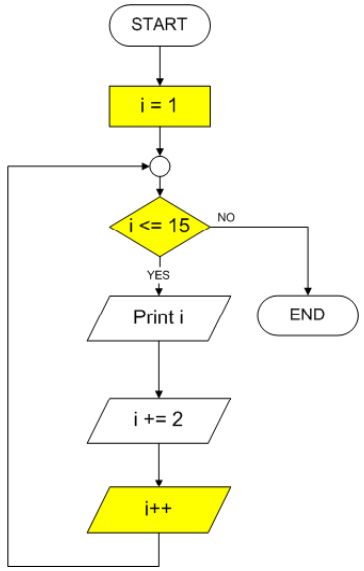
```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x = 0;
7      while(x <= 3){
8          x++;
9          cout << "Print " << x << '\n';
10         if(x%2 == 0) break;
11         cout << "Print " << 2*x << '\n';
12         cout << "Print " << 3*x << '\n';
13         cout << "------------------------\n";
14     }
15     return 0;
16 }
```


continue Statement

for

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 1; i <= 15; i++){
7          cout << "Print" << i << "\n";
8          i+=2;
9      }
10     return 0;
11 }
```

```
Print1
Print4
Print7
Print10
Print13
```

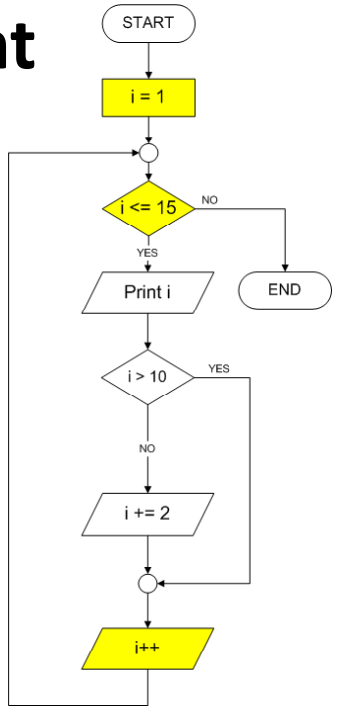


continue Statement

for

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 1; i <= 15; i++){
7          cout << "Print" << i << "\n";
8          if(i > 10) continue;
9          i+=2;
10     }
11     return 0;
12 }
```

```
Print1
Print4
Print7
Print10
Print13
Print14
Print15
```



continue Statement

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 1; i <= 15; i++){
7          cout << "Print" << i << "\n";
8          if(i > 10) continue;
9          i+=2;
10     }
11     return 0;
12 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 1; i <= 15; i++){
7          cout << "Print" << i << "\n";
8          if(i <= 10){
9              i+=2;
10         }
11     }
12     return 0;
13 }
```


Nested Loop Structure

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 0; i < 3; i++){
7          for(int j = 0; j < 5; j++){
8              cout << '*';
9          }
10         cout << '\n';
11     }
12     return 0;
13 }
```

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


Nested Loop Structure

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i = 0; i < 4; i++){
7          for(int j = 0; j <= i; j++){
8              cout << '*';
9          }
10         cout << '\n';
11     }
12     return 0;
13 }
```

Nested Loop Structure

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

int main()
{
    for(int i = 0; i < 4; i++){
        for(int j = 0; j <= i; j++){
            cout << '*';
        }
        cout << '\n';
    }
    return 0;
}
```

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

int main()
{
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < 4; j++){
            if(j > i) break;
            else cout << '*';
        }
        cout << '\n';
    }
    return 0;
}
```


Nested Loop Structure

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

int main()
{
    int N = 5;
    for(int i = 0; i <= N; i++){
        for(int j = 0; j <= 2*N; j++){
            if(abs(j-N) <= i) cout << '*';
            else cout << ' ';
        }
        cout << '\n';
    }
    for(int i = 0; i <= N/2; i++){
        for(int j = 0; j <= 2*N; j++){
            if(abs(j-N) <= N/4) cout << '*';
            else cout << ' ';
        }
        cout << '\n';
    }

    return 0;
}
```

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

N = 5

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

N = 9