Loops

Prepared by: Lec Tasmiah Tamzid Anannya, CS Dept, AIUB

The infinite loop

- A loop becomes an infinite loop if a condition never becomes false. The **for** loop is traditionally used for this purpose.
- Since none of the three expressions that form the 'for' loop are required, you can make an endless loop by leaving the conditional expression empty.
- When the conditional expression is absent, it is assumed to be true.

```
int main () {
  for( ; ; ) {
    printf("This loop will run forever.\n");
  }
  return 0;
}
```

The infinite loop

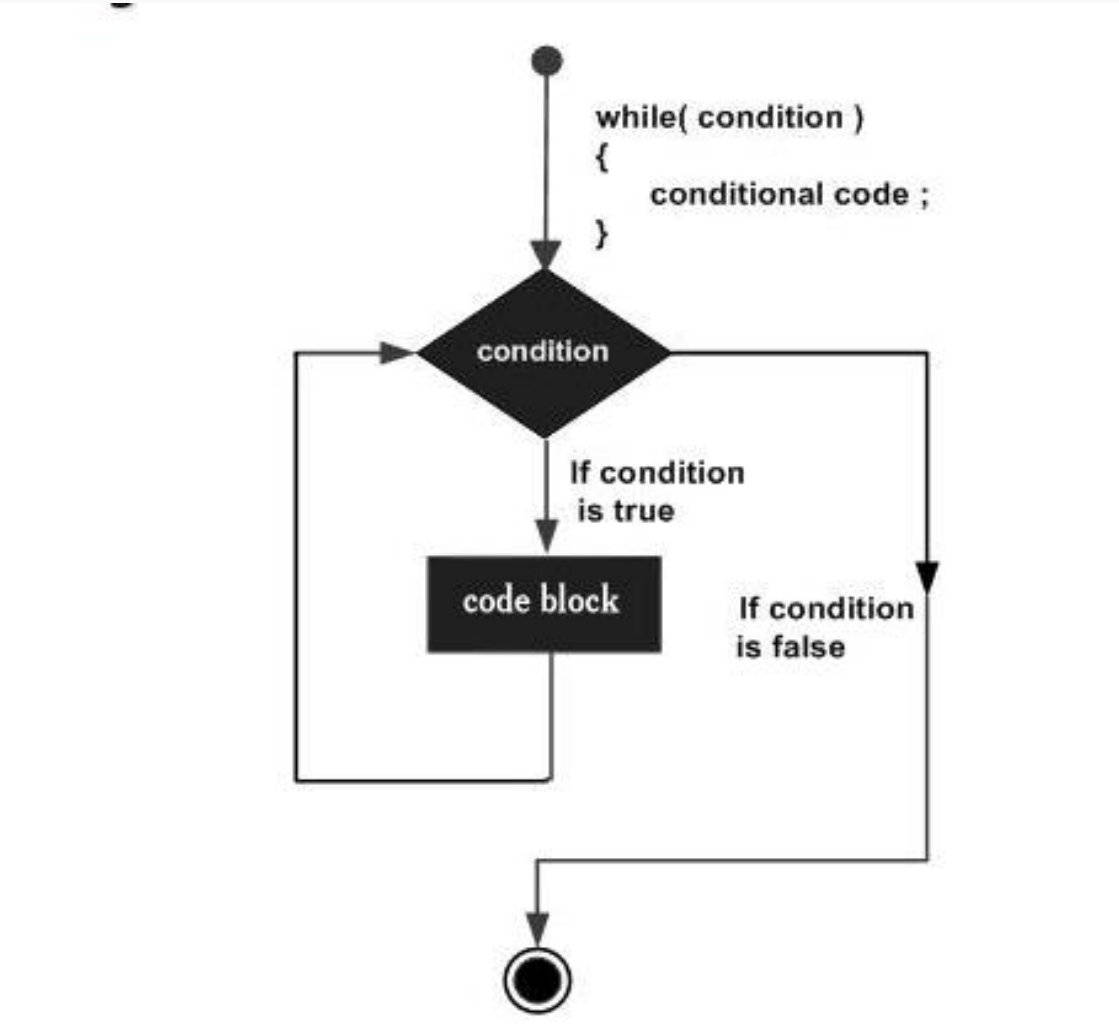
```
#include<stdio.h>

int main () {
int i;
   for(i=1;i>0 ;i++ ) {
      printf("This loop will run forever.\n");
   }

   return 0;
}
```

While loop

```
while(expression){
        statement(s);}
```

- As long as the expression is true, the statements are executed.
- If the expression is false, the loop will not execute even once.
- Here, the key point to note is that a **while** loop might not execute at all. When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed

Flowchart


while(condition)
{
conditional code ;
}
condition
If condition
is true
code block
If condition
is false

Example

```
#include<stdio.h>
int main()
{
    int i;
    char c;
    scanf("%c", &c);
    for(i=1;c!='q'; i++)
    {
        scanf("%c", &c);
    }
    return 0;
}
```

Example

```
#include<stdio.h>
int main()
{
    int i;
    char c;
    scanf("%c", &c);
    for(i=1;c!='q'; i++)
    {
        scanf("%c", &c);
    }
    return 0;
}
```

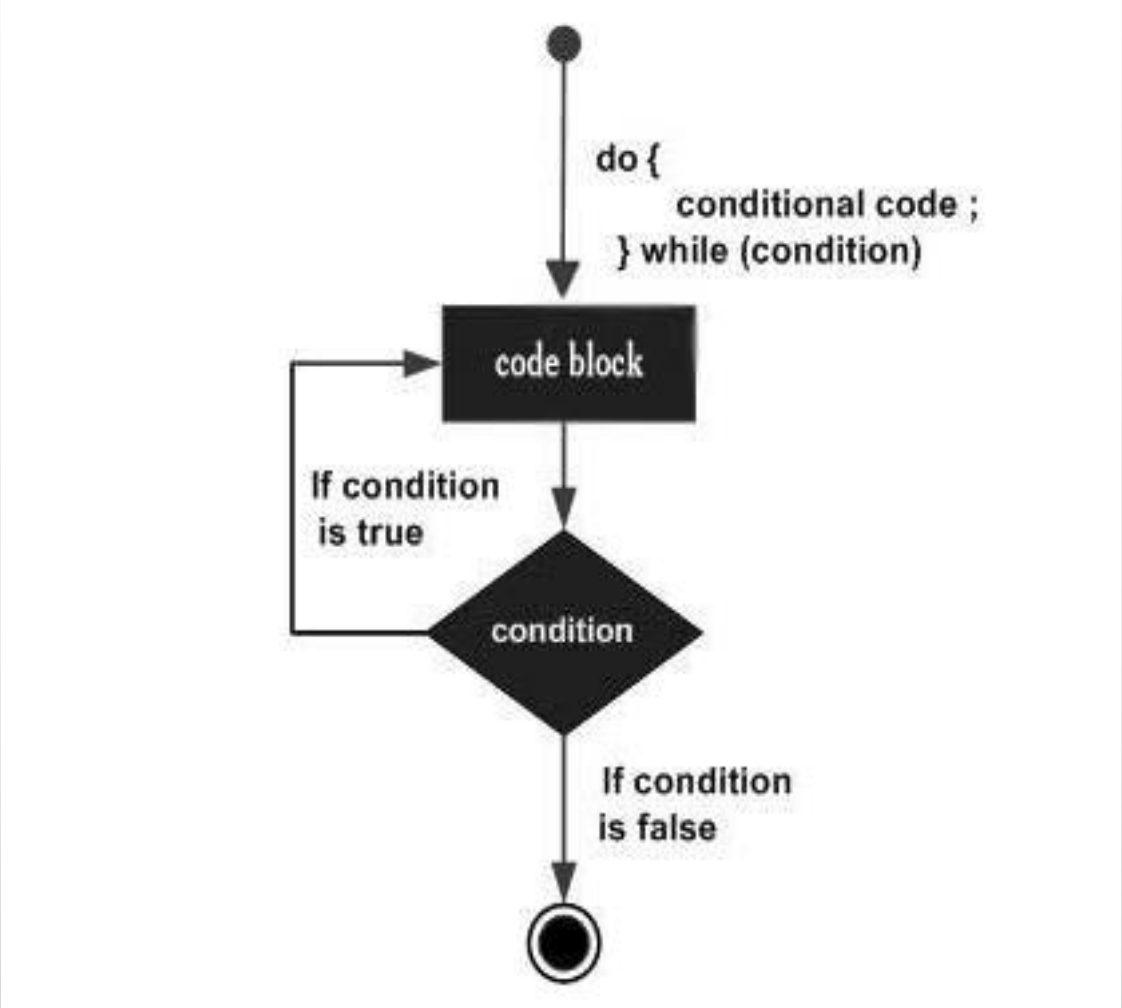
```
#include<stdio.h>
int main()
{
    char c;
    scanf("%c", &c);
    while(c!='q')
    {
        scanf("%c", &c);
    }
    return 0;
}
```

Do-while loop

```
do{
    statement(s);
    }while(condition);
```

- It is unique than other loops, as it will execute the code within the loop at least once.
- Because the expression is tested at the end of the loop.
- A **do...while** loop is similar to a while loop, except the fact that it is guaranteed to execute at least one time.

Flowchart

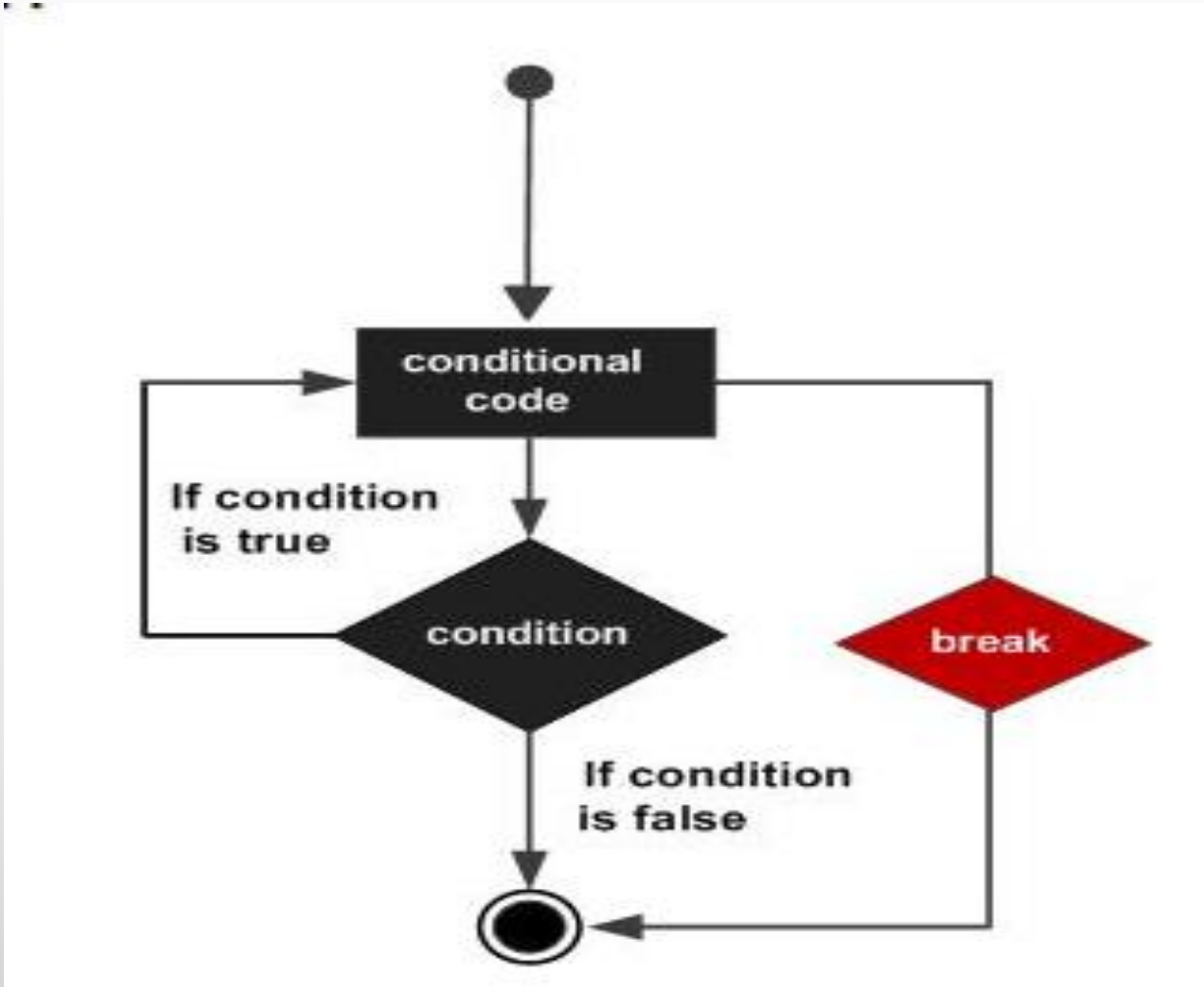

do {
conditional code ;
} while (condition)
code block
If condition
is true
condition
If condition
is false

Example

```
#include<stdio.h>
int main()
{
    char c;
    do{
        scanf("%c", &c);
    }while(c!='q');

}
```

Break to exit a loop

- It is sometimes desirable to skip some statements inside the loop or terminate the loop immediately without checking the test expression.
- The break statement allows to exit a loop from any point within its body, bypassing it's normal terminating expression.
- When the break statement is encountered, the loop is immediately stopped.
- The break statement can be used with all three of C's loops.


conditional
code
If condition
is true
condition
break
If condition
is false

How the break statement works?

```
while (testExpression) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

```
do {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

Use break in a while loop

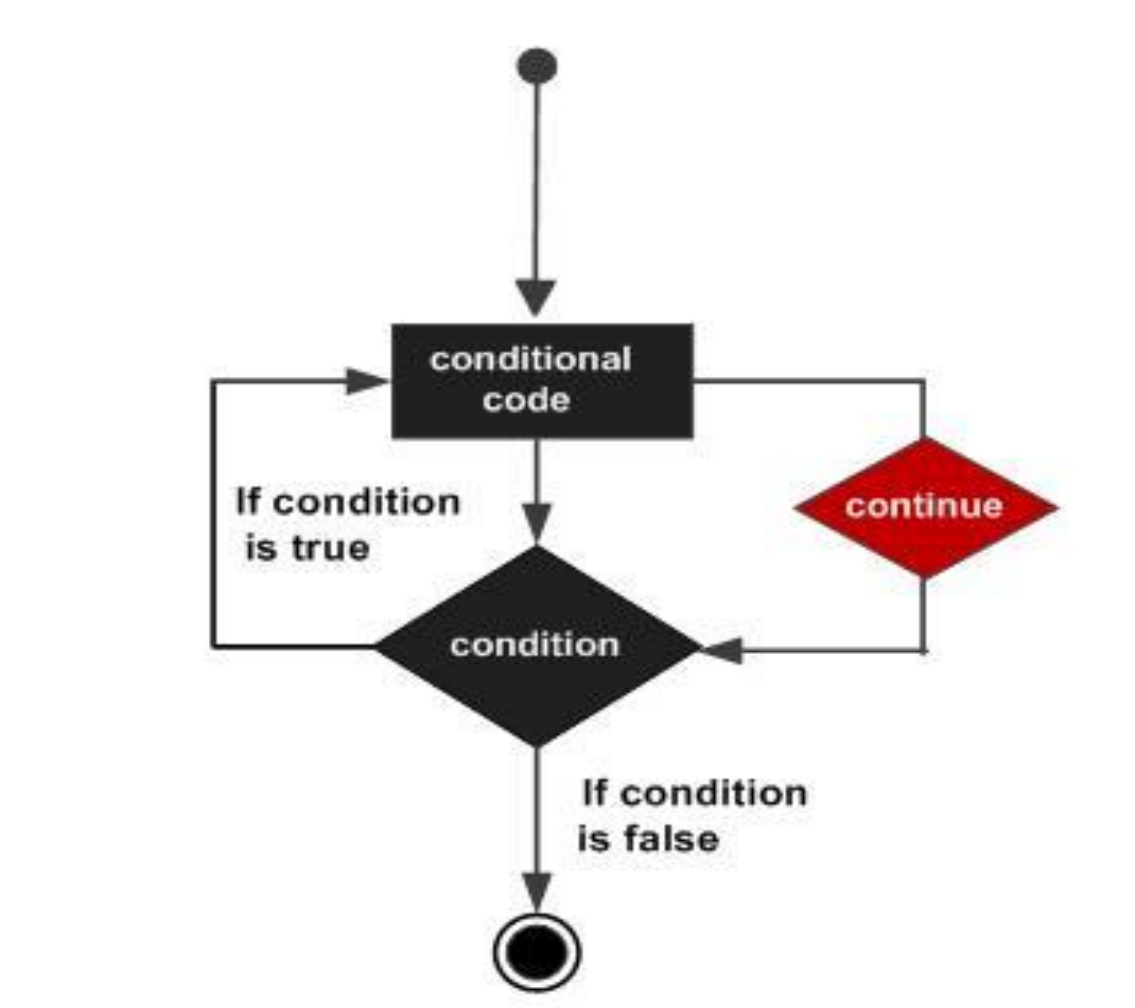
```
#include <stdio.h>
int main () {

   int a = 10;

   while( a < 20 ) {
      printf("value of a: %d\n", a);
      a++;
      if( a > 15)
      {
         /* terminate the loop using break statement */
         break;
      }
   }
   return 0;
}
```

Use break in a for loop

```
#include <stdio.h>
int main()
{
    int var;
    for (var =100; var>=10; var --)
    {
        printf("var: %d\n", var);
        if (var==99)
        {
            break;
        }
    }
    printf("Out of for-loop");
    return 0;
}
```

Continue statement

- The **continue** statement in C programming works somewhat like the **break** statement.
- Instead of forcing termination, it forces the next iteration of the loop to take place, skipping any code in between.


conditional
code
If condition
is true
continue
condition
If condition
is false

How the continue statement works?

```
while (testExpression) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

```
do {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

Use of continue statement in do…while

```
#include <stdio.h>
int main () {
   int a = 10;
   do {
      if( a == 15) {
         /* skip the iteration */
         a = a + 1;
         continue;
      }
      printf("value of a: %d\n", a);
      a++;

   } while( a < 20 );

   return 0;
}
```

Output:
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 16
value of a: 17
value of a: 18
value of a: 19

Use of continue statement in for loop

```
#include <stdio.h>
int main()
{
   for (int j=0; j<=8; j++)
   {
      if (j==4)
      {
           /* The continue statement is encountered when the value of j is equal to 4. */
                continue;
       }
    /* This print statement would not execute for the
    loop iteration where j ==4  because in that case this statement would be skipped.*/
       printf("%d ", j);
   }
   return 0;
}
```

Practice

- Ask the user to give a number and check whether the number is a prime number or not.

Nested Loop

- C programming allows to use one loop inside another loop.

```
outer_loop
{
    inner_loop
    {
        // Inner loop statement/s
    }

    // Outer loop statement/s
}
```

Example- nested for loop

```
for ( init; condition; increment ) { //outer loop

   for ( init; condition; increment ) { //inner loop
      statement(s);
   }
   statement(s);
}
```

Guess the output?

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=10; i++) /* Outer loop */
    {
        for(j=1; j<=5; j++)  /* Inner loop */
        {
            printf("%d ", j);
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

```
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
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
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Initialize i, i=1

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Initialize i, i=1

Initialize j, j=1

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Initialize i, i=1

Initialize j, j=1

i	j
1	1

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j
1	1

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

1<=1, T or F?

i	j
1	1

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Execute

i	j	
1	1	Ex

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Execute

i	j	
1	1	Ex

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Increment j, j=1+1=2

i	j	
1	1	Ex
1	2	

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

2<=1, T or F?

i	j	
1	1	Ex
1	2	Not Ex

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Print a new line

i	j	
1	1	Ex
1	2	Not Ex

```
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i=1+1=2

i	j	
1	1	Ex
1	2	Not Ex
2		

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2		

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

2<=3 T /F?

i	j	
1	1	Ex
1	2	Not Ex
2		

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

2<=3 T /F?

Initialize j, j=1

i	j	
1	1	Ex
1	2	Not Ex
2		

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2	1	

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

1<=2, T or F?

i	j	
1	1	Ex
1	2	Not Ex
2	1	

*

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Execute

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex

```
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Execute

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex

```
*
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Increment j, j=1+1=2

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	

```
*
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

2<=2, T or F?

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	

```
*
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Execute

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex

```
*
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Increment j, j=2+1=3

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	

```
*
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

3<=2, T or F?

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex

```
*
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

Print a new line

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex

```
*
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex

```
*
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex

```
*
**
*
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex
3	2	Ex

```
*
**
**
```

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex
3	2	Ex
3	3	Ex

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

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

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

i	j	Print *
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex
3	2	Ex
3	3	Ex
3	4	Not Ex

Print a pattern

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=3; i++) /* Outer loop */
    {
        for(j=1; j<=i; j++)  /* Inner loop */
        {
            printf("*");
        }
        /* Print a new line */
        printf("\n");
    }
        return 0;
}
```

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

i	j	
1	1	Ex
1	2	Not Ex
2	1	Ex
2	2	Ex
2	3	Not Ex
3	1	Ex
3	2	Ex
3	3	Ex
3	4	Not Ex
4		Outer loop Exit

Practice

- Print the following patterns:

* ** *** **** ***** ******	1 12 123 1234 12345 123456

- Print all the prime numbers within 1 to 100.