

# A Full Program

```
public class FactorialExample extends ConsoleProgram {

    private static final int MAX_NUM = 4;

    public void run() {
        for(int i = 0; i < MAX_NUM; i++) {
            println(i + "! = " + factorial(i));
        }
    }

    private int factorial(int n) {
        int result = 1;
        for (int i = 1; i <= n; i++) {
            result *= i;
        }
        return result;
    }
}
```

# A Full Program

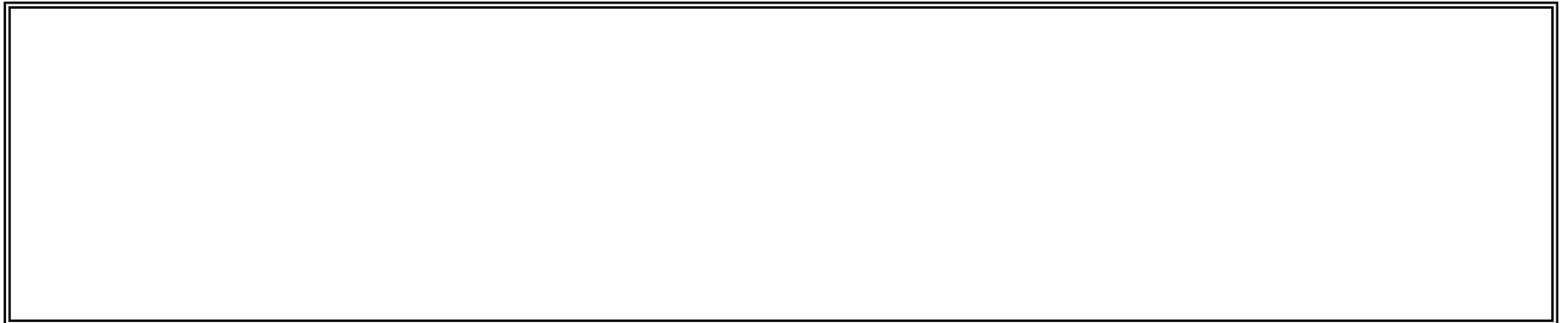
```
public class FactorialExample extends ConsoleProgram {  
  
    private static final int MAX_NUM = 4;  
  
    public void run() {  
        for(int i = 0; i < MAX_NUM; i++) {  
            println(i + "! = " + factorial(i));  
        }  
    }  
  
    private int factorial(int n) {  
        int result = 1;  
        for (int i = 1; i <= n; i++) {  
            result *= i;  
        }  
        return result;  
    }  
}
```

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i

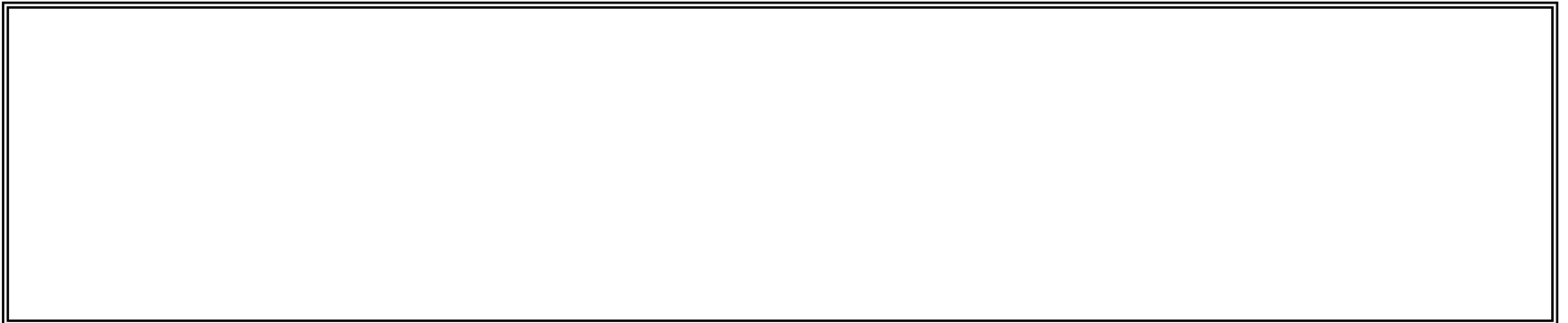
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 0



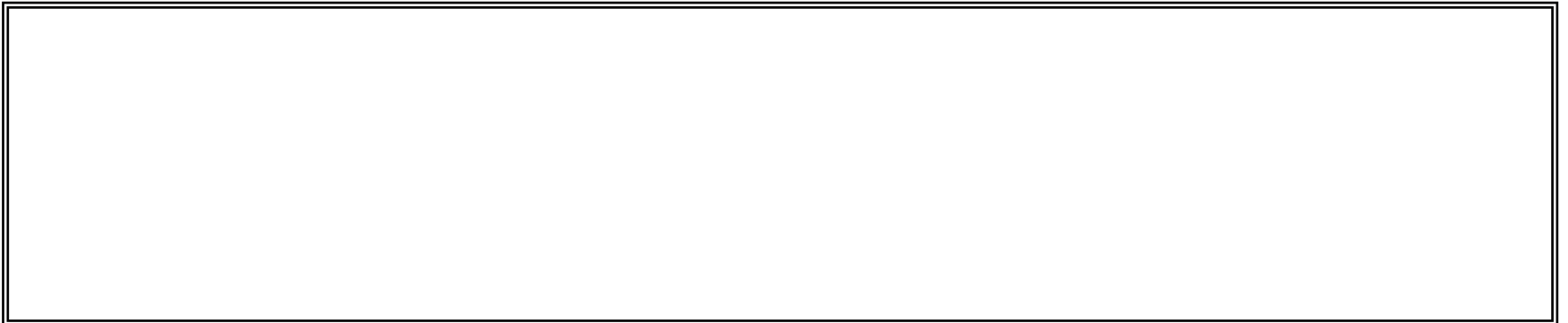
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 0



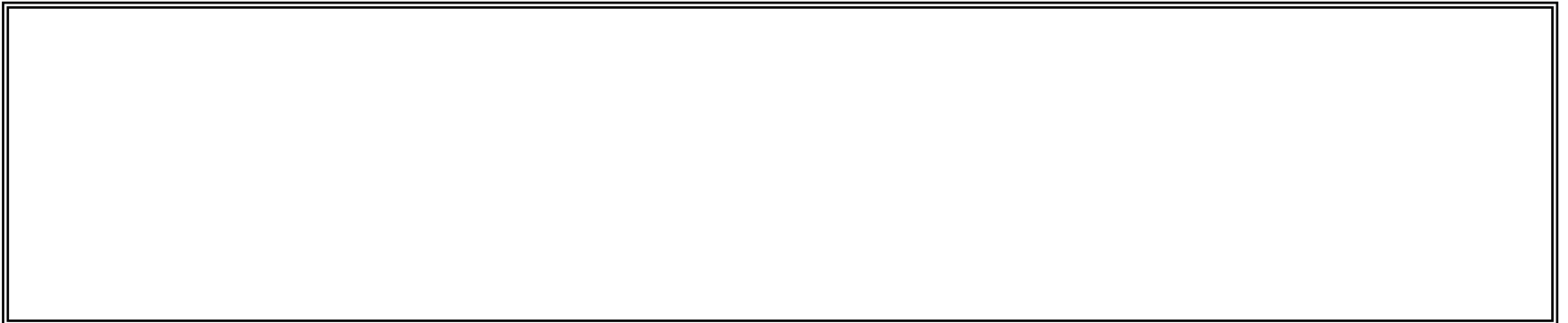
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 0



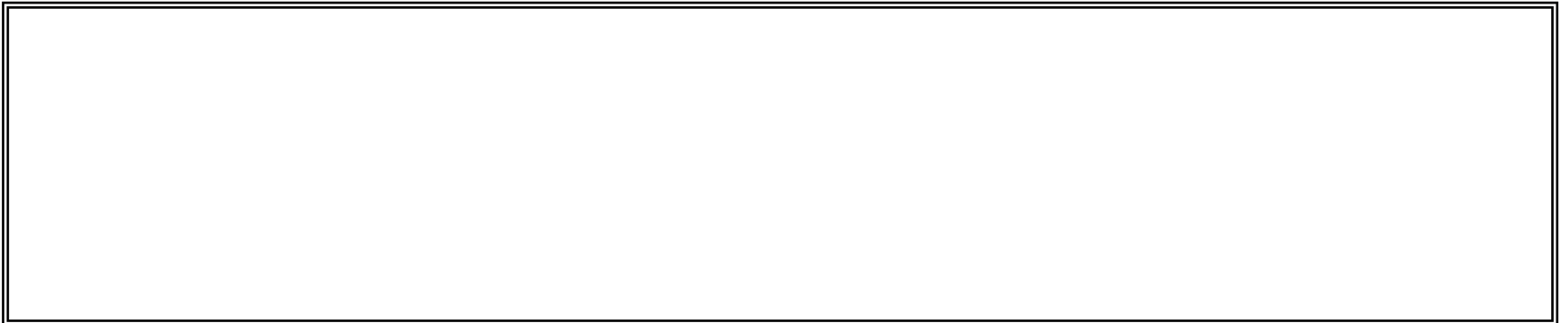
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 0



```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i



```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

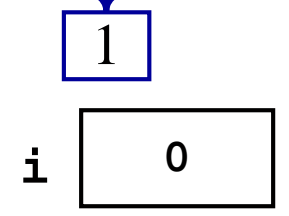
```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```



i 0

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

The diagram illustrates the state of the program during the first iteration of the loop. A blue bracket under the `factorial(i)` call in the `println` statement points to a small box containing the value `1`. Below this, the variable `i` is shown next to a box containing the value `0`, indicating that the loop is currently at `i = 0`.

$0! = 1$

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 1

0! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 1

0! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 

1
---

0! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 

1
---

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

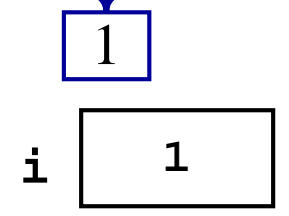
0! = 1

```
private int factorial(int n) {  
    int result = 1;  
    for (int i = 1; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

n  result  i

0! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```



0! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

The diagram illustrates the first iteration of the factorial function. A blue bracket underlines the expression `factorial(i)` in the `println` statement. A blue box containing the number `1` is positioned directly below the bracket, indicating the result of the factorial function for `i = 0`. Below this, the variable `i` is shown next to a black box containing the number `1`, representing the current value of `i` in the loop.

`0! = 1`

`1! = 1`

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 2

0! = 1

1! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 2

0! = 1

1! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 2

0! = 1

1! = 1

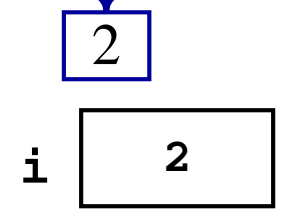
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 2

0! = 1

1! = 1

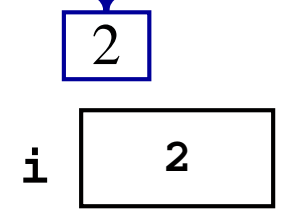
```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```



0! = 1

1! = 1

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```



0! = 1  
1! = 1  
2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 3

0! = 1

1! = 1

2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 3

0! = 1

1! = 1

2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 3

0! = 1

1! = 1

2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

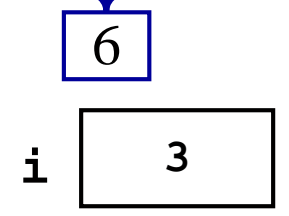
i 3

0! = 1

1! = 1

2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

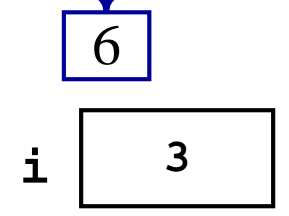


0! = 1

1! = 1

2! = 2

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```



0! = 1  
1! = 1  
2! = 2  
3! = 6

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 4

0! = 1

1! = 1

2! = 2

3! = 6

```
public void run() {  
    for(int i = 0; i < MAX_NUM; i++) {  
        println(i + "! = " + factorial(i));  
    }  
}
```

i 4

0! = 1

1! = 1

2! = 2

3! = 6

# Bad Times with Methods

```
// NOTE: This program is buggy!!
```

```
private void addFive(int x) {  
    x += 5;  
}
```

```
public void run() {  
    int x = 3;  
    addFive(x);  
    println("x = " + x);  
}
```

# Good Times with Methods

// NOTE: This program is **feeling just fine...**

```
private int addFive(int x) {  
    x += 5;  
    return x;  
}
```

```
public void run() {  
    int x = 3;  
    x = addFive(x);  
    println("x = " + x);  
}
```