

# *Conditional and iterative statements*



Jordi Cortadella  
Department of Computer Science

# Maximum of two numbers

---

- Write a program that reads two numbers and prints the maximum value of both.
- Example:
  - The maximum of 20 and 38 is 38.
  - The maximum of -3 and -8 is -3.

# Maximum of two numbers

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

```
// This program reads two numbers and
// prints the maximum value of both
```

```
int main() {
    int x, y;
    cin >> x >> y;
    int m;
    if (x > y) m = x;
    else m = y;
    cout << "The maximum value is "
         << m << endl;
}
```



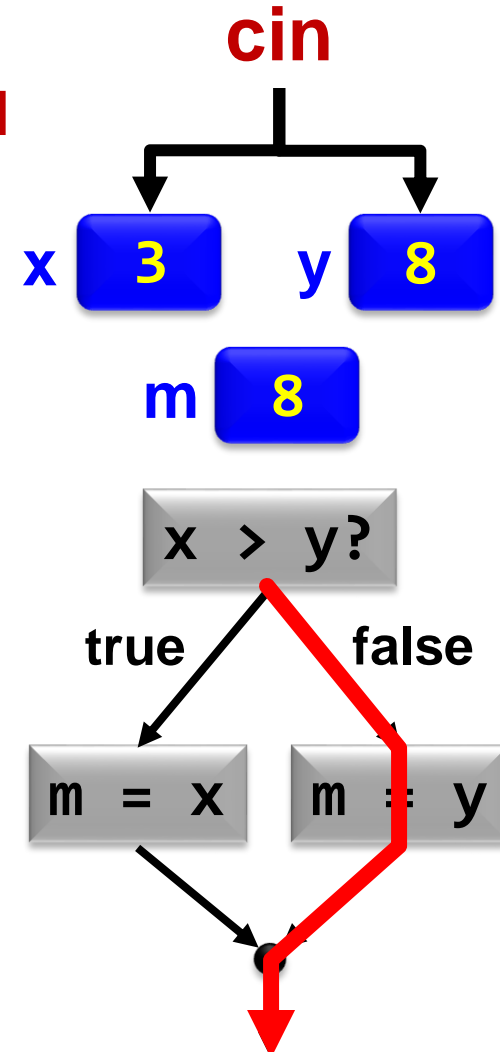
Conditional statement

# Maximum of two numbers

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

```
// This program reads two numbers and
// prints the maximum value of both
```

```
int main() {
    int x, y;
    cin >> x >> y;
    int m;
    if (x > y) m = x;
    else m = y;
    cout << "The maximum value is "
         << m << endl;
}
```



# Maximum of two numbers (II)

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

// This program reads two numbers and
// prints the maximum value of both

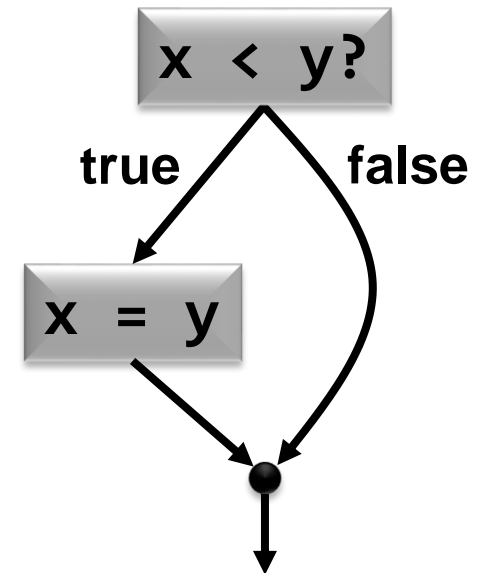
int main() {
    int x, y;
    cin >> x >> y;
    cout << "The maximum value is ";
    if (x > y) cout << x;
    else cout << y;
    cout << endl;
}
```

# Maximum of two numbers (III)

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

// This program reads two numbers and
// prints the maximum value of both

int main() {
    int x, y;
    cin >> x >> y;
    if (x < y) x = y;
    cout << x << endl;
}
```



# Repetitive statements

---

- Assume the following specification:

**Input:** reads a number  $n > 0$

**Output:** prints the sequence 1 2 3 ... n  
(one number per line)

- This specification suggests some algorithm with a *repetitive* procedure.

# Print the numbers 1..N

---

```
> print_numbers
```

```
8
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
>
```



# Print the numbers 1..N

**// Input: reads a number  $n > 0$**   
**// Output: prints the numbers 1..n (one per line)**

```
int main() {  
    int n;  
    cin >> n;  
    cout << 1 << endl;  
    cout << 2 << endl;  
    cout << 3 << endl;  
    cout << 4 << endl;  
        ...  
    cout << n << endl;  
}
```

} How many?

# Print the numbers 1..N

**// Input: reads a number  $n > 0$**   
**// Output: prints the numbers 1...n (one per line)**

```
int main() {  
    int n;  
    cin >> n;  
    int i = 1;  
    while (i <= n) {  
        cout << i << endl;  
        i = i + 1;  
    }  
}
```

n **5**

i	i <= n	Loop body
1	true	cout << 1 << endl; i = 1 + 1;
2	true	cout << 2 << endl; i = 2 + 1;
3	true	cout << 3 << endl; i = 3 + 1;
4	true	cout << 4 << endl; i = 4 + 1;
5	true	cout << 5 << endl; i = 5 + 1;
6	false	

# The *while* statement

- Syntax:

*while* ( condition ) statement;

(the condition must return *true* or *false*)

- Semantics:
  - Similar to the repetition of an *if* statement
  - The condition is evaluated:
    - If *true*, the statement is executed and the control returns to the while statement again.
    - If *false*, the while statement terminates.

# Multiplication table

- Write a program that reads a number  $n$  (between 1 and 9) and prints the multiplication table of  $n$ :
- Example for  $n = 6$ :

```
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
```

# Multiplication table

**// Input: reads a number  $0 < n < 10$**   
**// Output: prints the multiplication table of n**

```
int main() {  
    int n;  
    cin >> n;  
    int i = 1;  
    while (i <= 9) { // or also (i < 10)  
        // print n x i = n*i  
        cout << n << " x " << i  
            << " = " << n*i << endl;  
        i = i + 1;  
    }  
}
```

# Summary

---

- Conditional statements (*if-else*) are used to take decisions that depend on values of variables.
- The *while* statement is the fundamental instruction to iterate under a condition that determines termination.
- The control of conditional and loop statements is determined by Boolean expressions.