Mr C makes a Choice

ESC101: Fundamentals of Computing

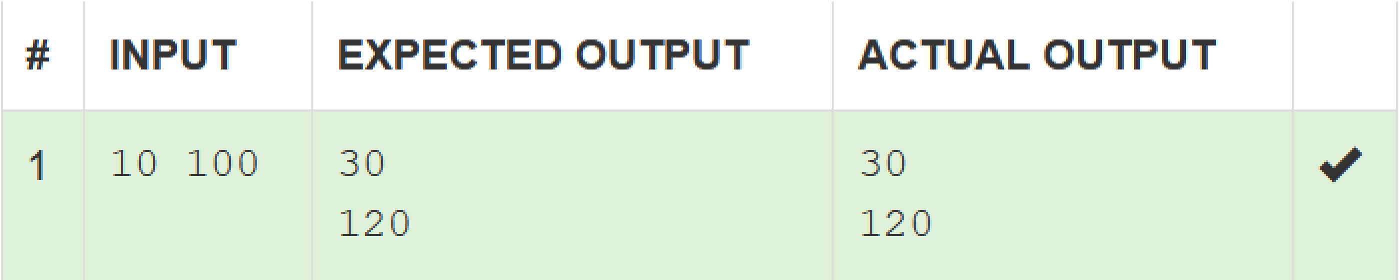
Purushottam Kar

Announcements - Lab

- Solutions uploaded to the website
- Grades released after lab over this evening
- No grace marks for “I wrote at least some code” or “just a small calculation mistake”
- Only when autograder makes mistake – regrading
- Recall, useless regrading requests will be penalized

How autograding was done

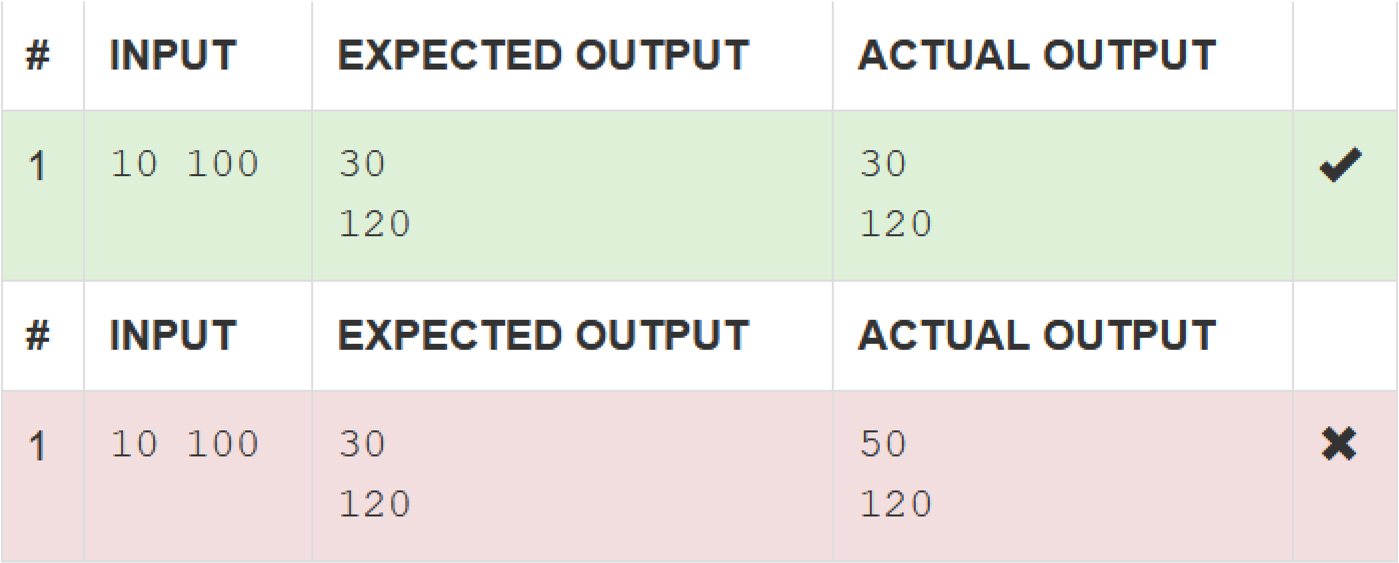
How autograding was done



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

100% marks, both lines correct

How autograding was done



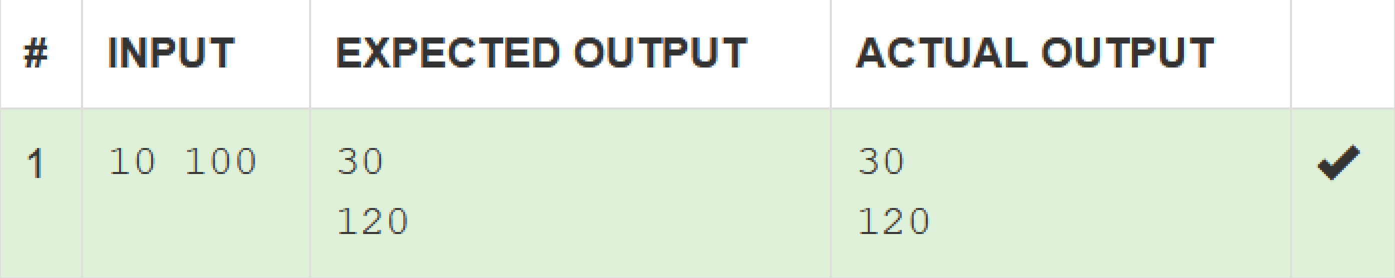
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✖

100% marks, both lines correct

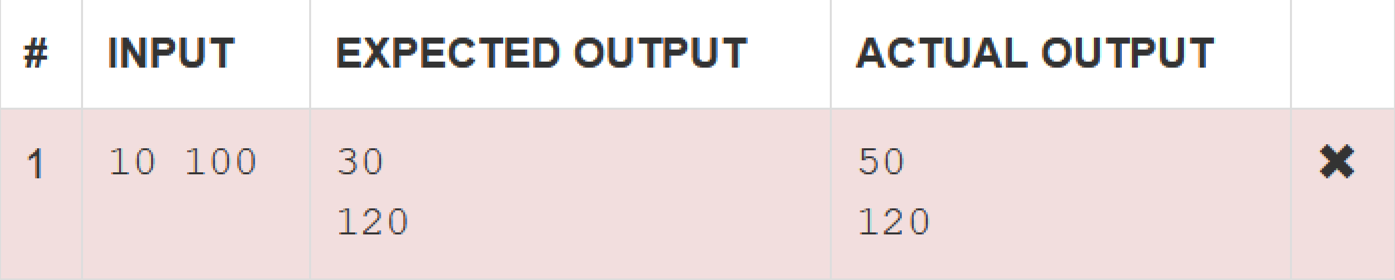
50% marks, first line incorrect but second line correct

How autograding was done



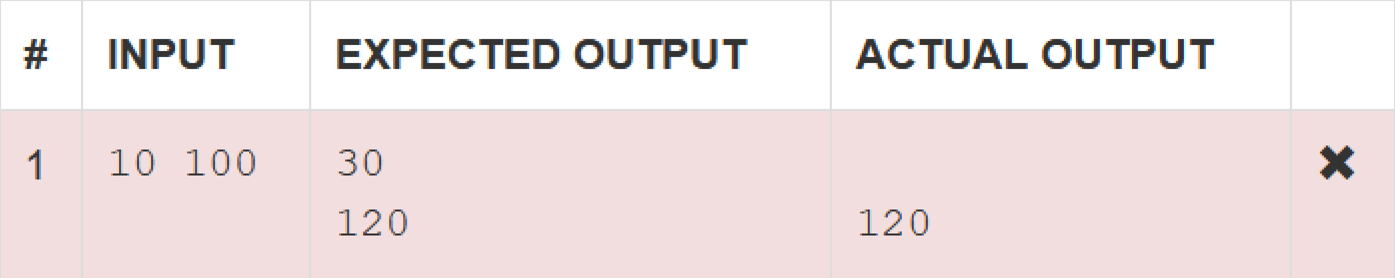
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

100% marks, both lines correct



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✖

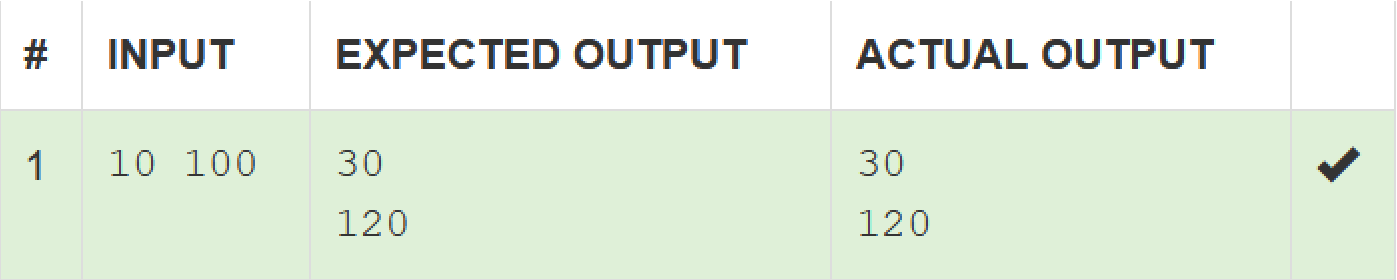
50% marks, first line incorrect but second line correct



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	 120	✖

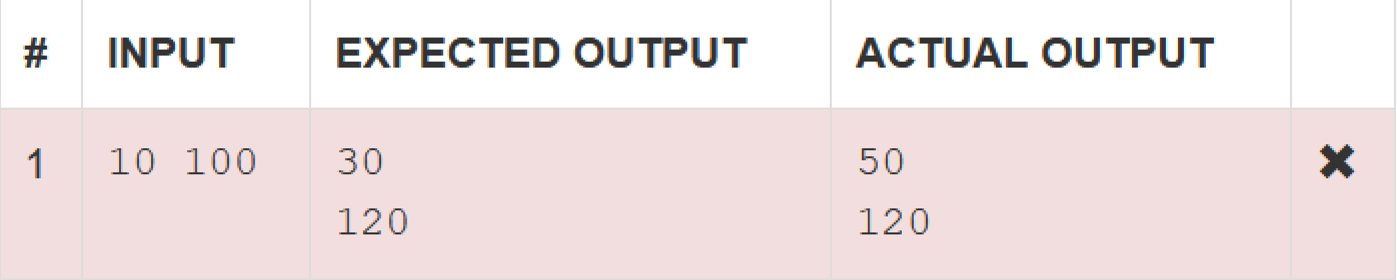
50% marks, first line blank (incorrect), second line correct

How autograding was done



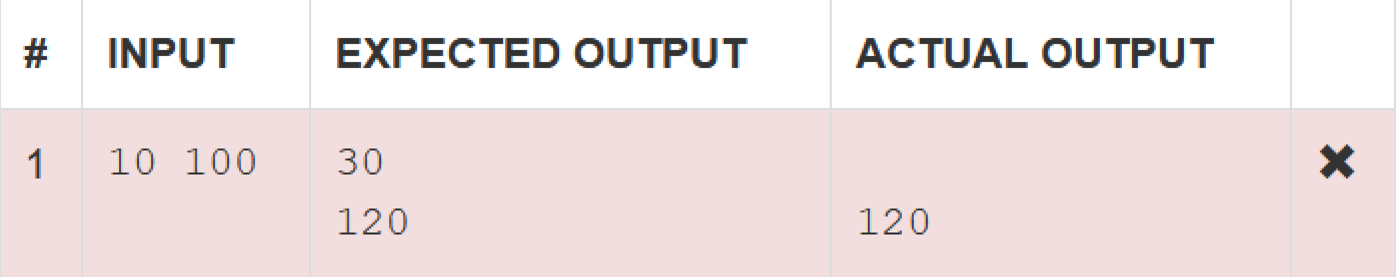
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

100% marks, both lines correct



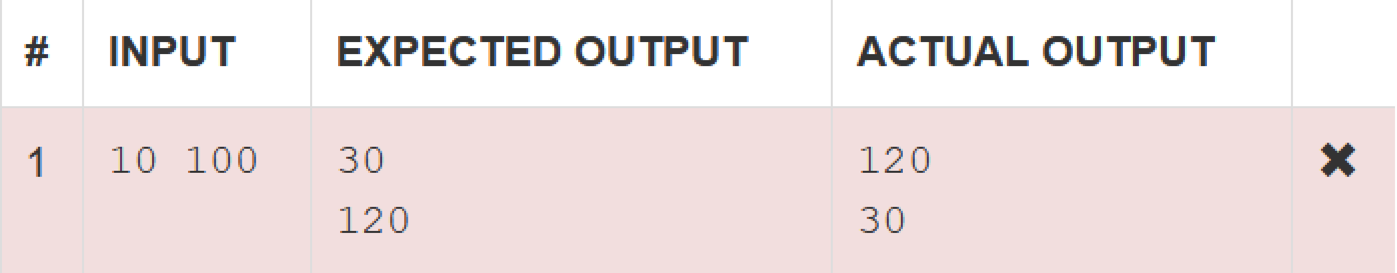
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✗

50% marks, first line incorrect but second line correct



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	 120	✗

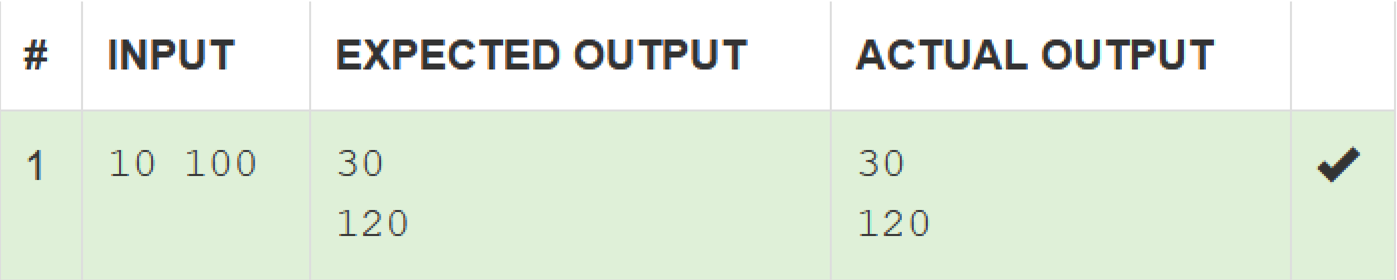
50% marks, first line blank (incorrect), second line correct



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	120 30	✗

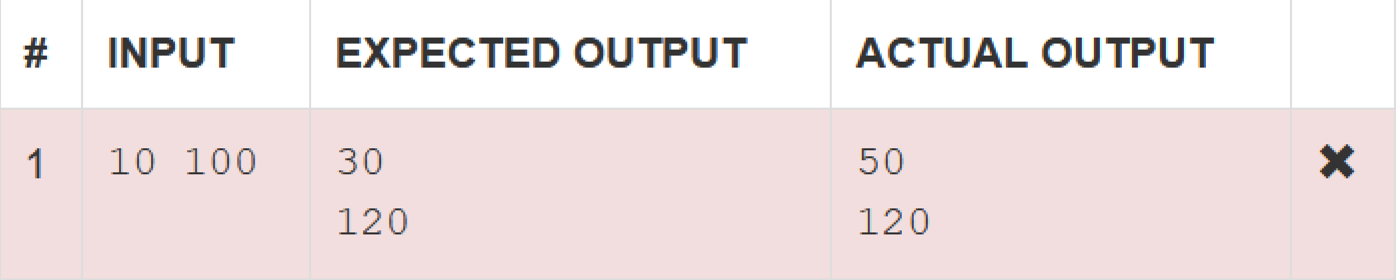
0% marks, no line matches the expected output line (wrong order)

How autograding was done



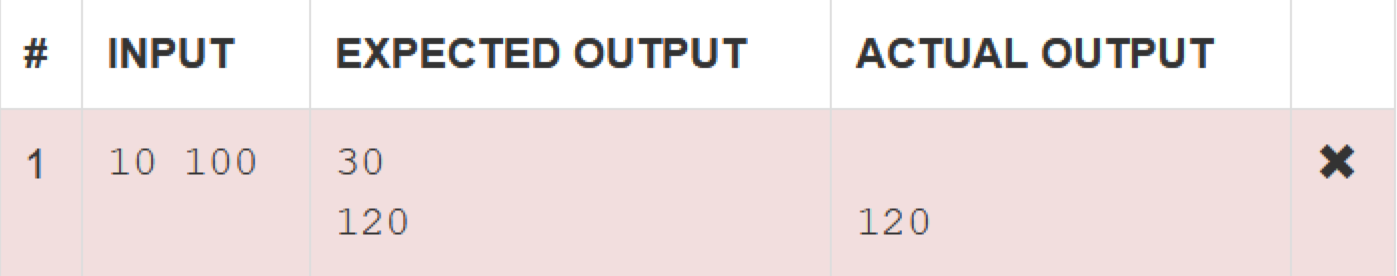
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

100% marks, both lines correct



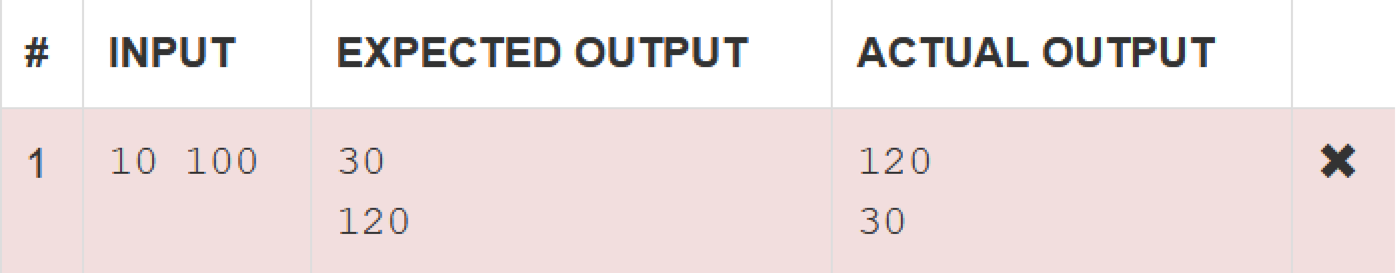
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✖

50% marks, first line incorrect but second line correct



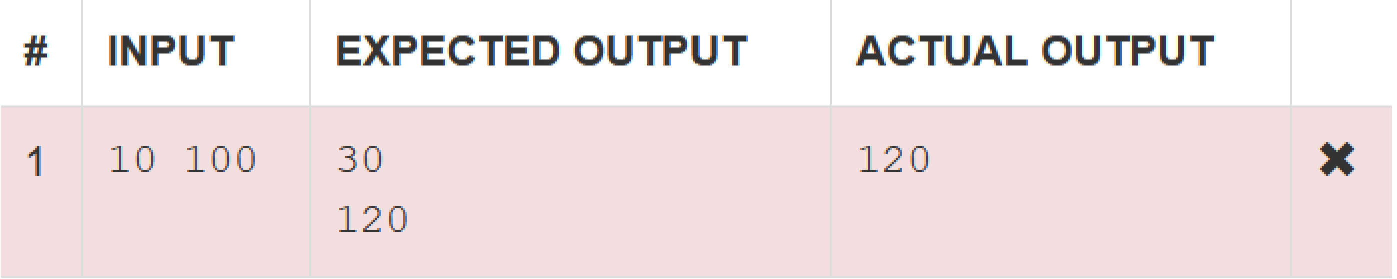
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	 120	✖

50% marks, first line blank (incorrect), second line correct



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	120 30	✖

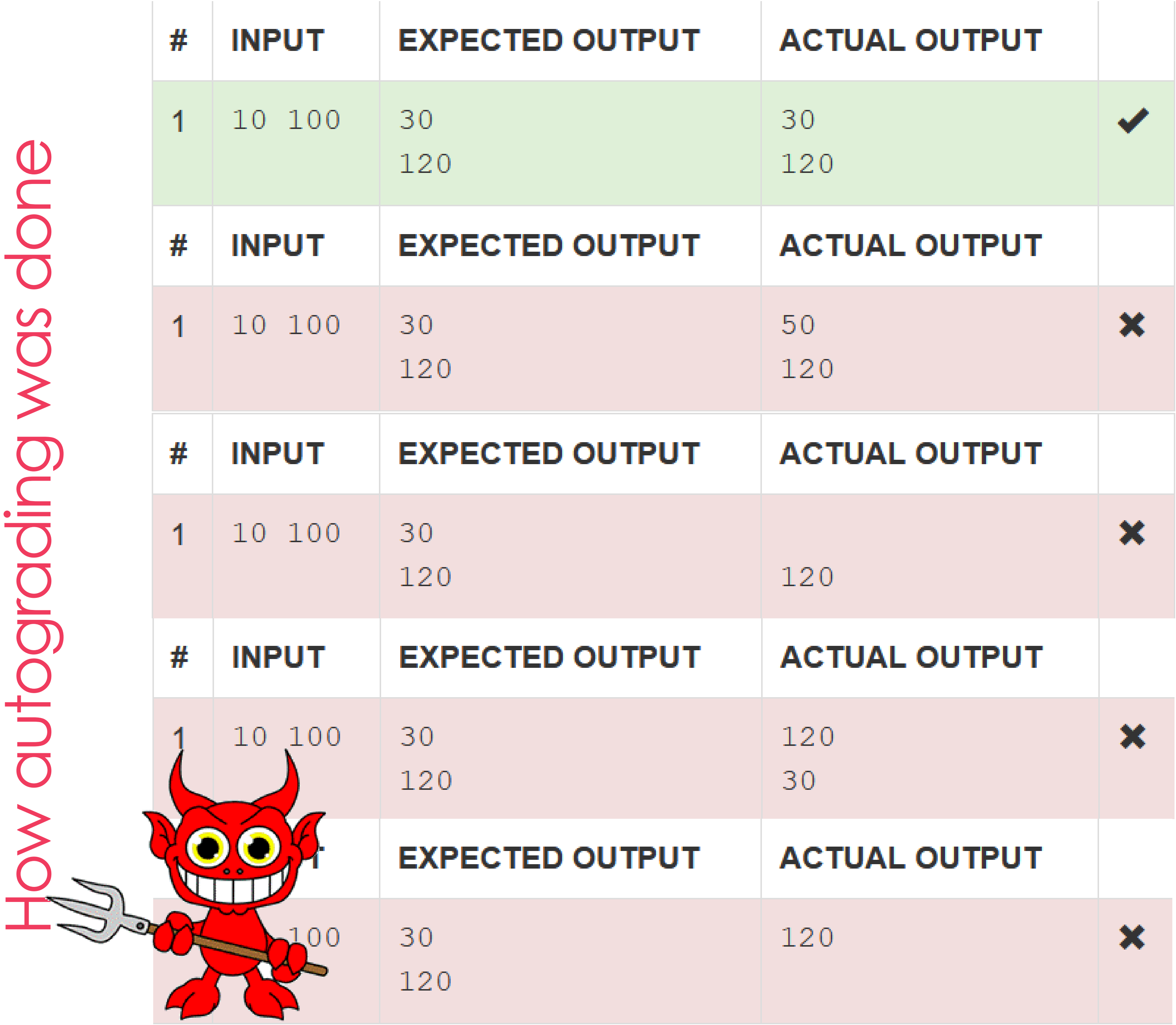
0% marks, no line matches the expected output line (wrong order)



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	120	✖

0% marks, no line matches the expected output line

How autograding was done



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✓

100% marks, both lines correct

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✗

50% marks, first line incorrect but second line correct

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	 120	✗

50% marks, first line blank (incorrect), second line correct

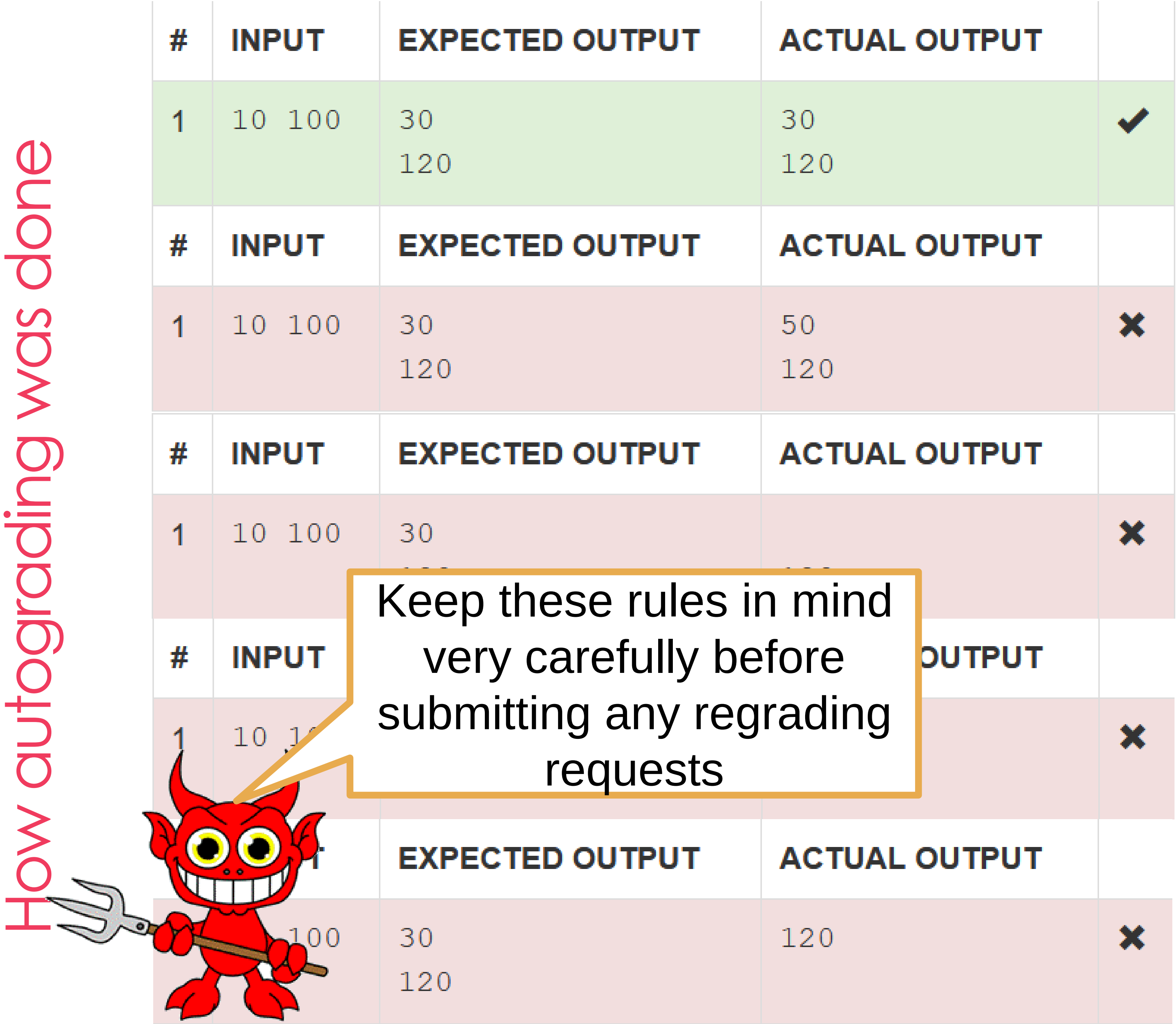
#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	120 30	✗

0% marks, no line matches the expected output line (wrong order)

	T	EXPECTED OUTPUT	ACTUAL OUTPUT	
	100	30 120	120	✗

0% marks, no line matches the expected output line

How autograding was done



#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	30 120	✔

100% marks, both lines correct

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	50 120	✖

50% marks, first line incorrect but second line correct

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	
1	10 100	30 120	 120	✖

50% marks, first line blank (incorrect), second line correct

#	INPUT		OUTPUT	
1	10 1			✖

0% marks, no line matches the expected output line (wrong order)

		EXPECTED OUTPUT	ACTUAL OUTPUT	
	100	30 120	120	✖

0% marks, no line matches the expected output line

Keep these rules in mind very carefully before submitting any regrading requests

Announcements – Minor Quiz

- Will be released very soon
- Some delay in setting up Gradescope
- Sorry for the delay

Announcements - Bonus

Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled

Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course

Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course
- Reports of students taking unfair help from seniors and A batch students with prior programming experience

Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course
- Reports of students taking unfair help from seniors and A batch students with prior programming experience
- No more bonus problems from now on

Announcements - Bonus

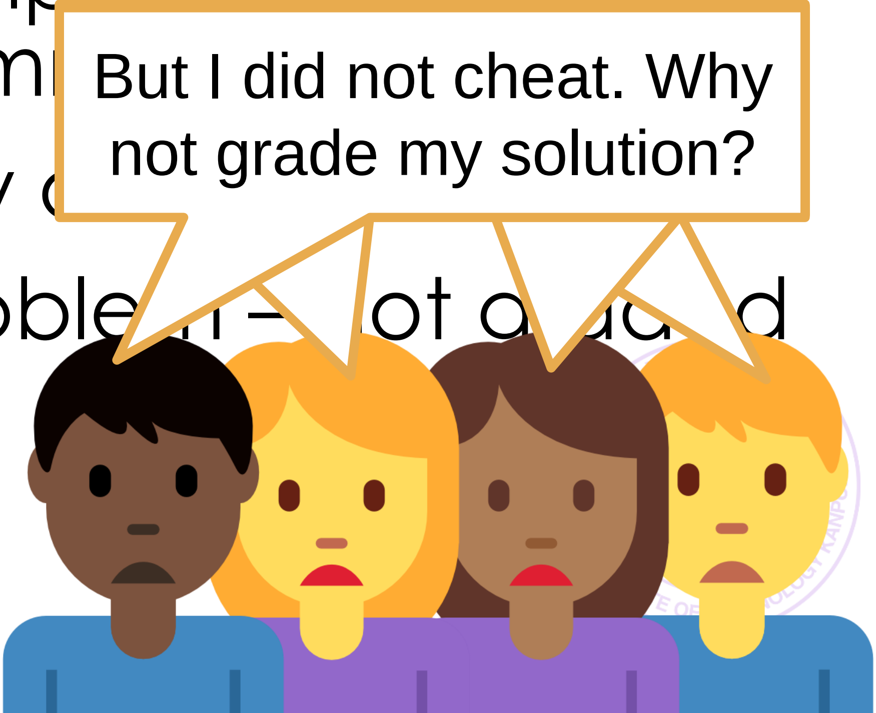
- Bonus question (Monster Multiply) has been cancelled
- Won't be graded, won't add towards course
- Reports of students taking unfair help from seniors and A batch students with prior programming experience
- No more bonus problems from now on
- Sometimes, challenge practice problem – not graded

Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won't be graded, won't add towards course
- Reports of students taking unfair help from seniors and A batch students with prior programming experience
- No more bonus problems from now on
- Sometimes, challenge practice problem – not graded

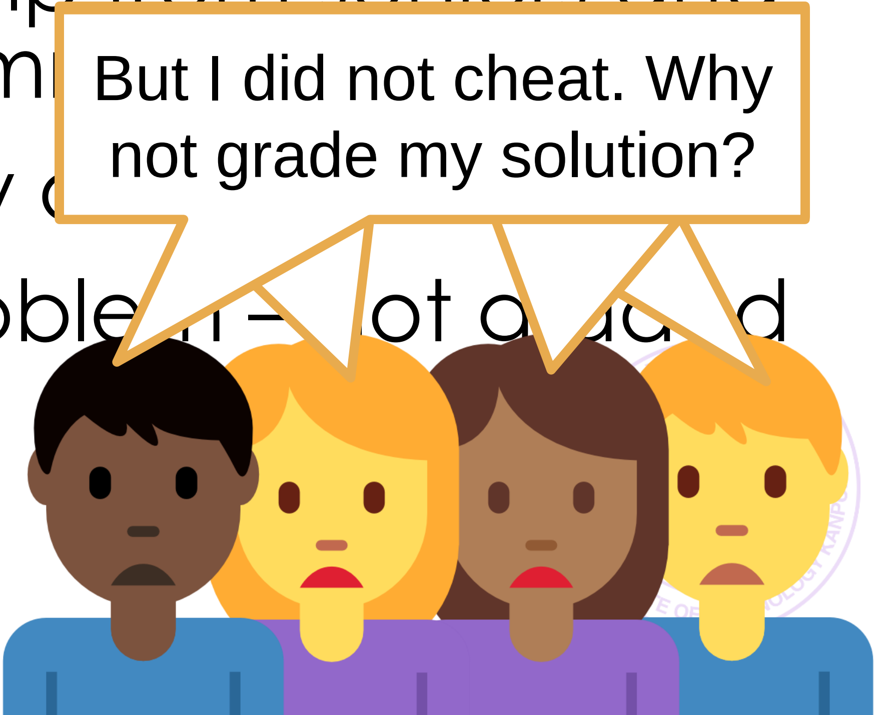
Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course
- Reports of students taking unfair help from seniors and A batch students with prior programm
- No more bonus problems from now o
- Sometimes, challenge practice proble



Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course
- Reports of students taking unfair help from seniors and A batch students with prior program
- No more bonus problems from now o
- Sometimes, challenge practice proble



Announcements - Bonus

- Bonus question (Monster Multiply) has been cancelled
- Won’t be graded, won’t add towards course
- Reports of students taking unfair help from seniors and A batch ogram
- No more n now c
- Someti ce proble ot a d

In June, CBSE question paper got leaked. Only 10-20 students involved in leak but retest was done for all lakhs of students to be fair to all

But I did not cheat. Why not grade my solution?

Error in previous slide (corrected)

Error in previous slide (corrected)

Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right

Error in previous slide (corrected)

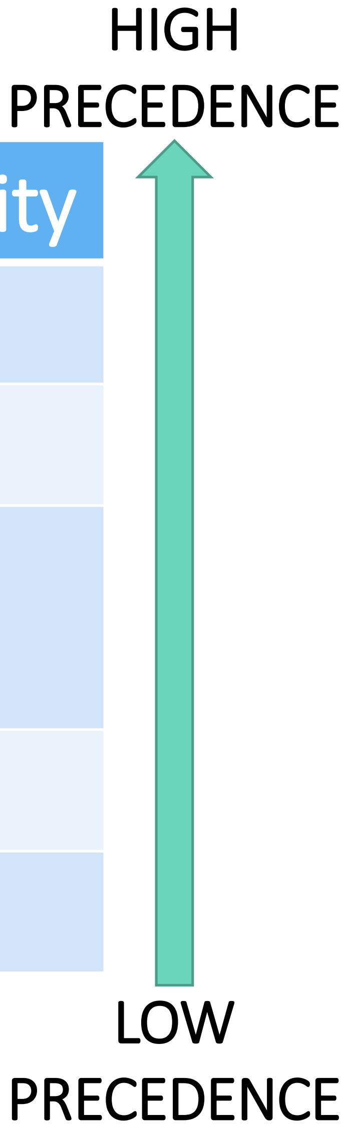
HIGH PRECEDENCE

Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right

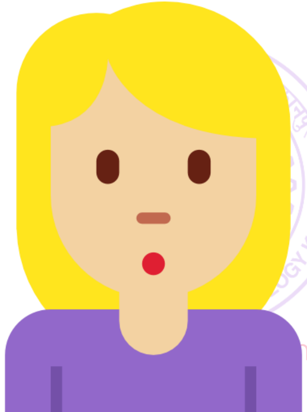
LOW PRECEDENCE

Error in previous slide (corrected)

Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right


HIGH PRECEDENCE

LOW PRECEDENCE



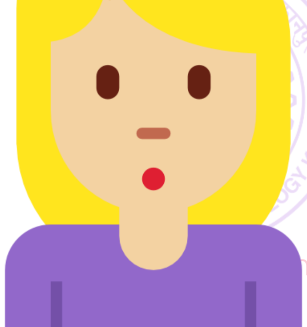
Error in previous slide (corrected)

Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right

HIGH PRECEDENCE

LOW PRECEDENCE

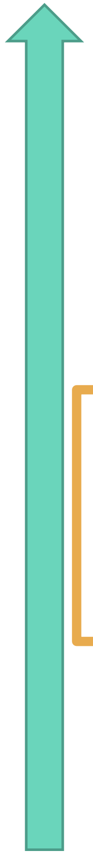
So bracket has higher priority than unary negation



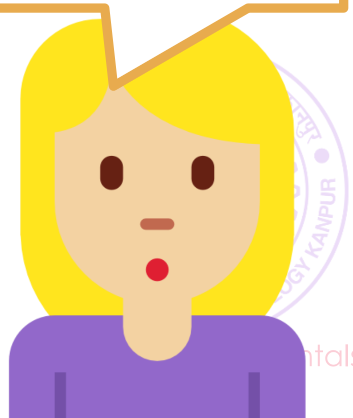
Error in previous slide (corrected)

Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right

HIGH PRECEDENCE



LOW PRECEDENCE



So bracket has higher priority than unary negation

Error in previous slide (corrected)

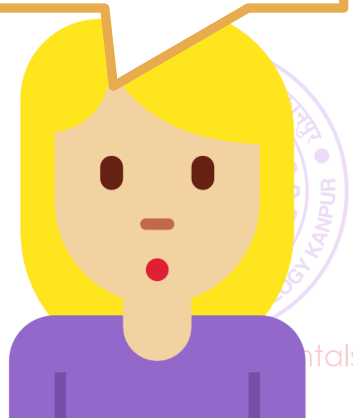
Operator Name	Symbol/Sign	Associativity
Bracket	()	Left
Unary negation	-	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment	=	Right

HIGH PRECEDENCE

LOW PRECEDENCE

Sorry about the error

So bracket has higher priority than unary negation



A Few Helpful Tips

A Few Helpful Tips

Double and float are both happy with %f for printf

A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double

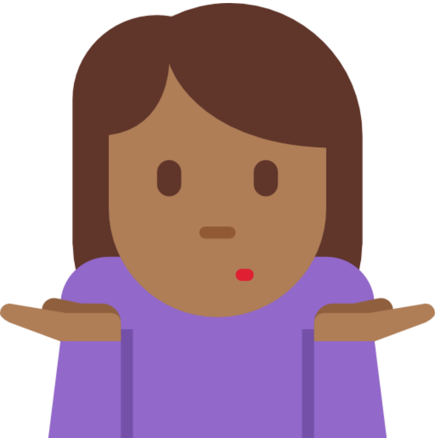
A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double



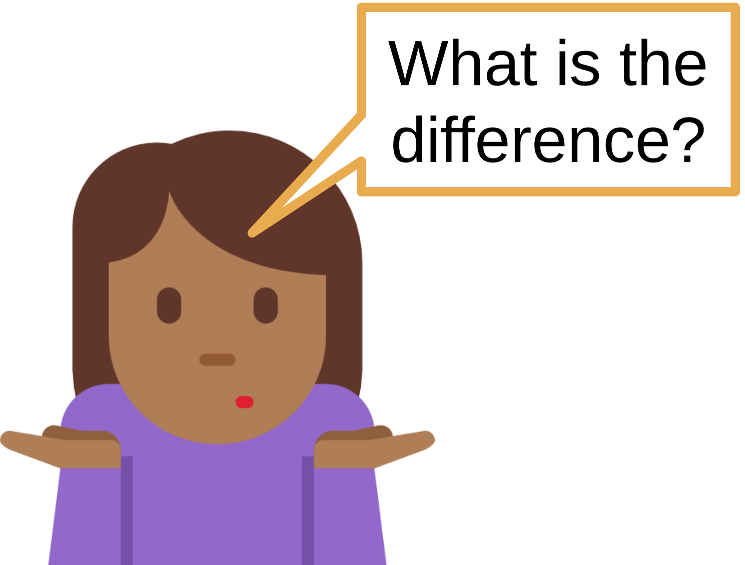
A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double



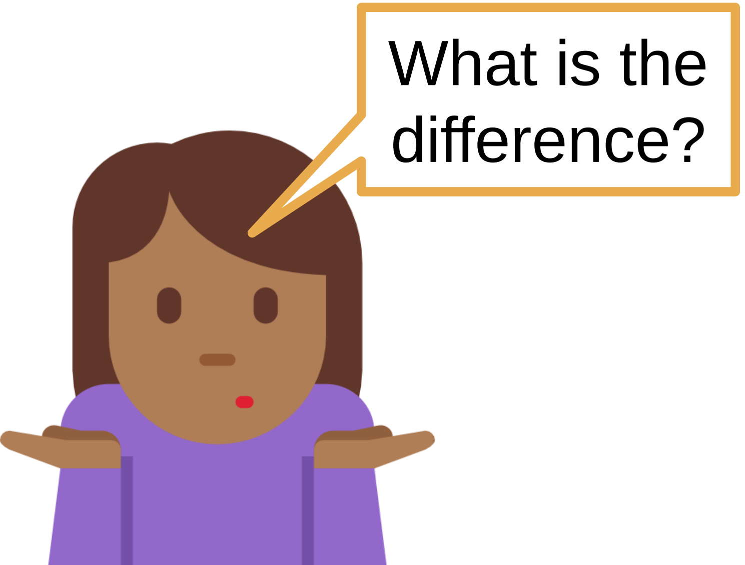
A Few Helpful Tips

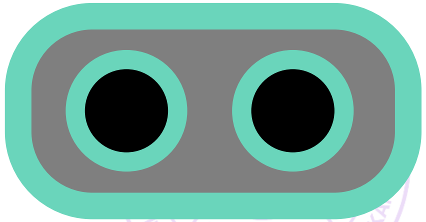
Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double





A Few Helpful Tips

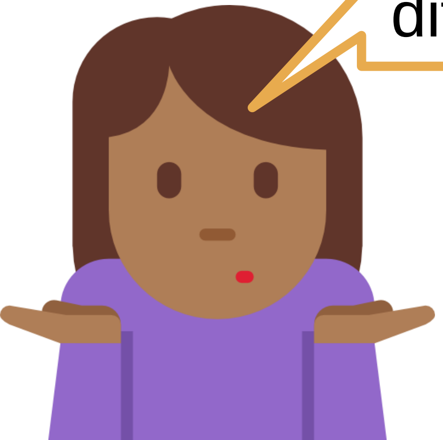
Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

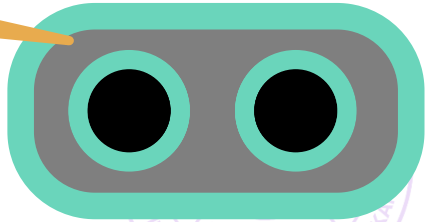
For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double

When you say long a = 3213213210, since the number is within range of long, I will preserve every digit of it carefully



What is the difference?



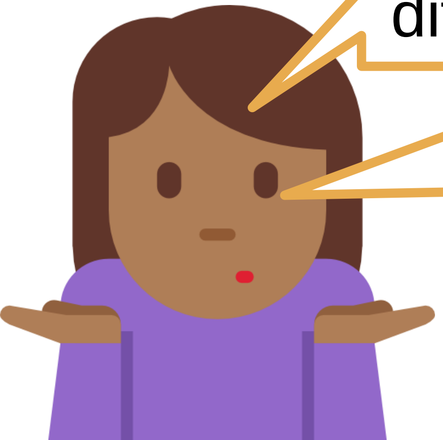
A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

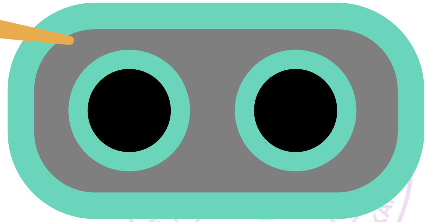
For problems where you do not mind approximations, use float, double



What is the difference?

Range of float is larger. What if I store it as a float?

When you say long a = 3213213210, since the number is within range of long, I will preserve every digit of it carefully



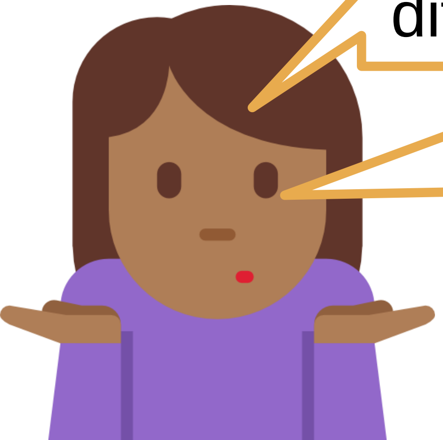
A Few Helpful Tips

Double and float are both happy with %f for printf

However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double

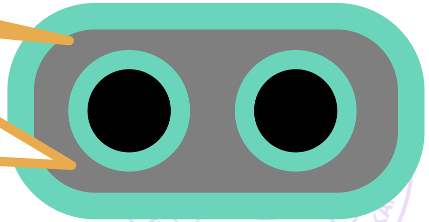


What is the difference?

Range of float is larger. What if I store it as a float?

When you say long a = 3213213210, since the number is within range of long, I will preserve every digit of it carefully

When you say float a = 3213213210, I will store 3213213184.00



A Few Helpful Tips

Double and float are both happy with %f for printf

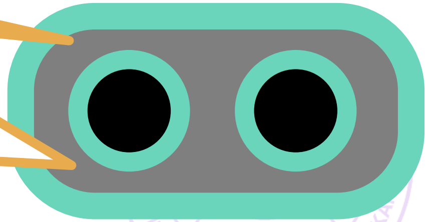
However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double

What is the difference?

Range of float is larger. What if I store it as a float?



When you say long a = 3213213210, since the number is within range of long, I will preserve every digit of it carefully

When you say float a = 3213213210, I will store 3213213184.00

A Few Helpful Tips

Double and float are both happy with %f for printf

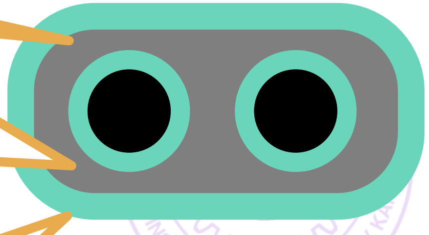
However, in scanf, double insists on %lf (%f gives junk)

For problems where you care about the exact digits (lab problems), avoid float, double as far as possible

For problems where you do not mind approximations, use float, double

What is the difference?

Range of float is larger. What if I store it as a float?



When you say long a = 3213213210, since the number is within range of long, I will preserve every digit of it carefully

When you say float a = 3213213210, I will store 3213213184.00

The number is like 3.2×10^9 and my error was just 26. Don't blame me ☹

A Cute Trick for Higher Precision

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

A Cute Trick for Higher Precision

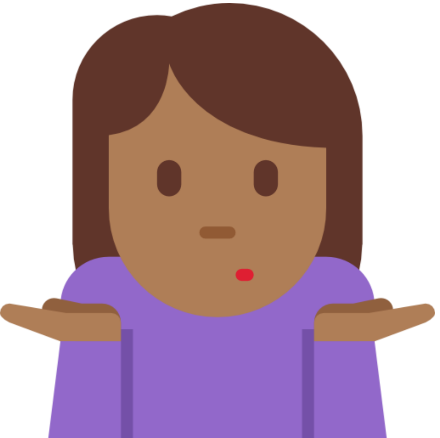
Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

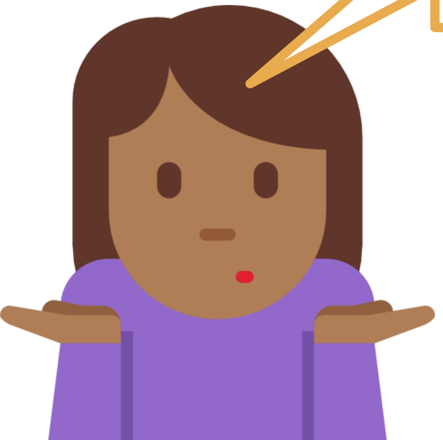
Don’t read it as a float or double or even long double



A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double



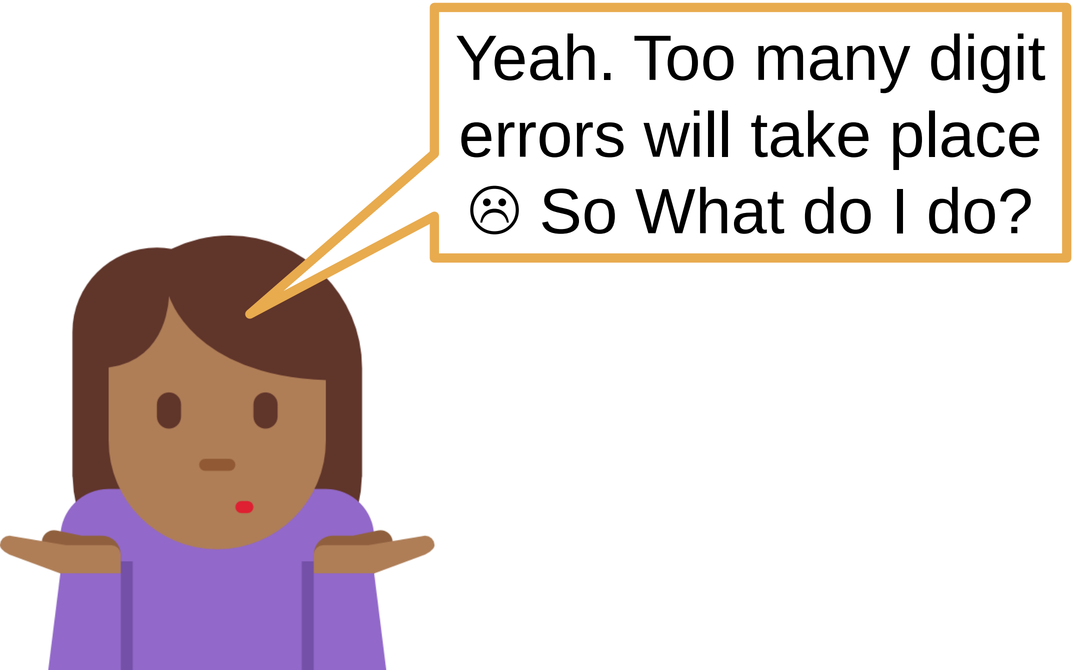
Yeah. Too many digit errors will take place ☹ So What do I do?

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don't read it as a float or double or even long double

Read it as two int variables, separated by a decimal

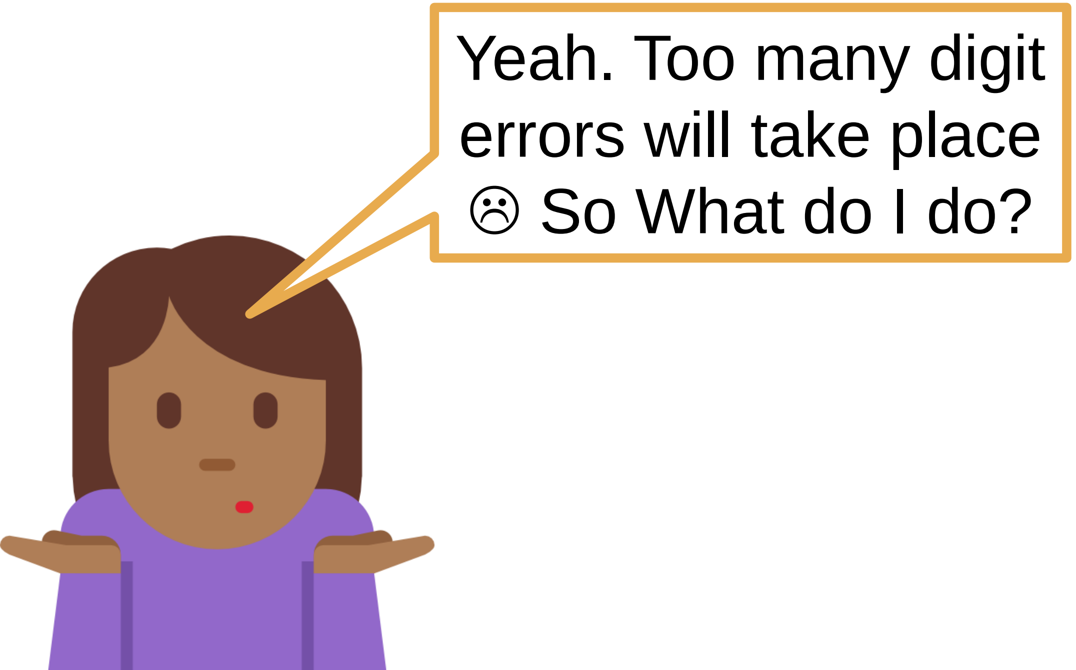

Yeah. Too many digit errors will take place ☹ So What do I do?

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal

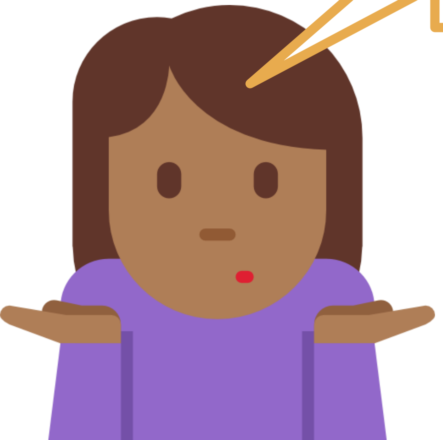

Yeah. Too many digit errors will take place ☹ So What do I do?

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don't read it as a float or double or even long double

Read it as two int variables, separated by a decimal



Yeah. Too many digit errors will take place ☹ So What do I do?

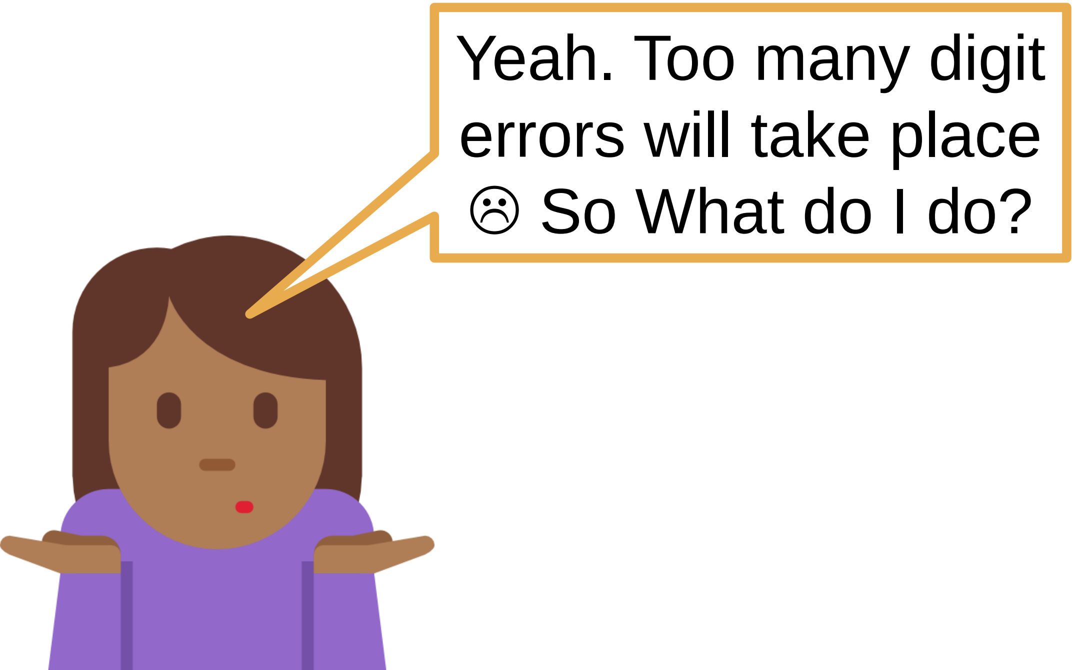
```
#include <stdio.h>
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal


Yeah. Too many digit errors will take place ☹ So What do I do?

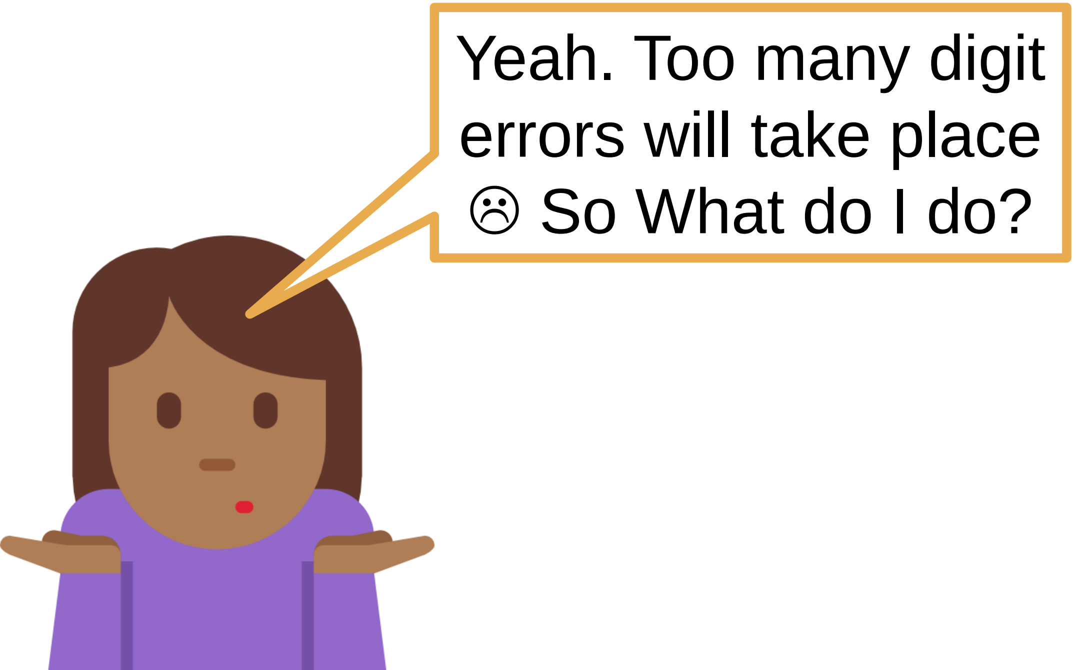
```
#include <stdio.h>
int main(){
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal


Yeah. Too many digit errors will take place ☹ So What do I do?

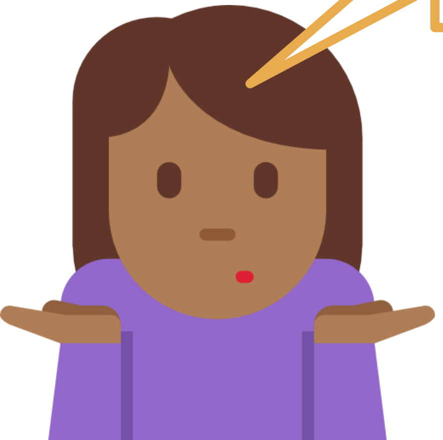
```
#include <stdio.h>
int main(){
    int a, b;
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal



Yeah. Too many digit errors will take place ☹ So What do I do?

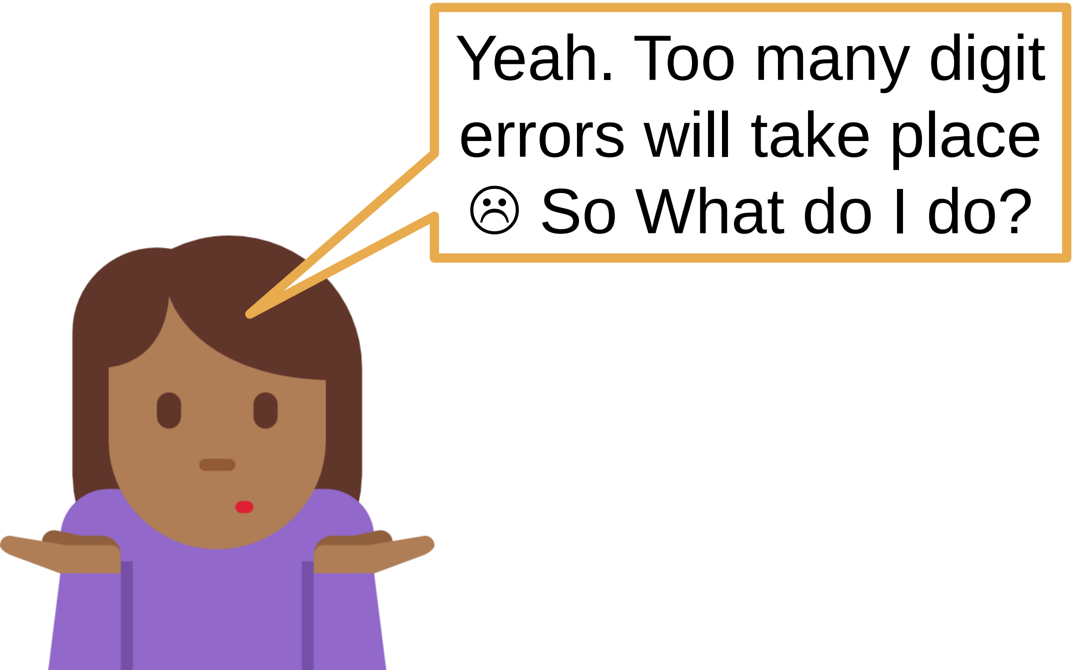
```
#include <stdio.h>
int main(){
    int a, b;
    scanf(“%d.%d”, &a, &b);
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal


Yeah. Too many digit errors will take place ☹ So What do I do?

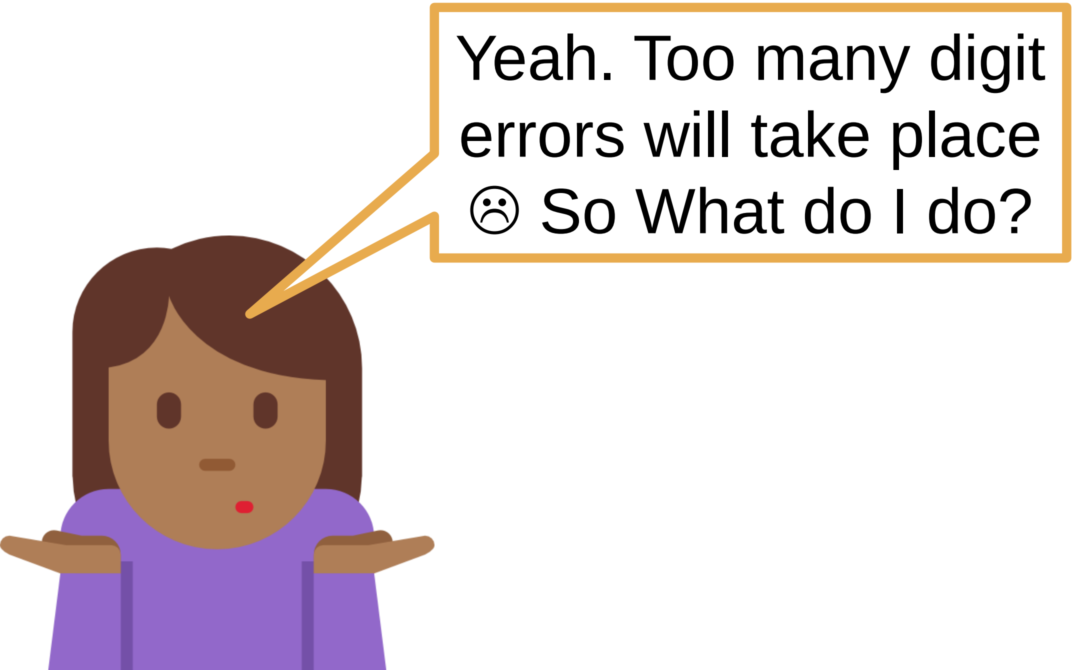
```
#include <stdio.h>
int main(){
    int a, b;
    scanf(“%d.%d”, &a, &b);
    return 0;
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal


Yeah. Too many digit errors will take place ☹ So What do I do?

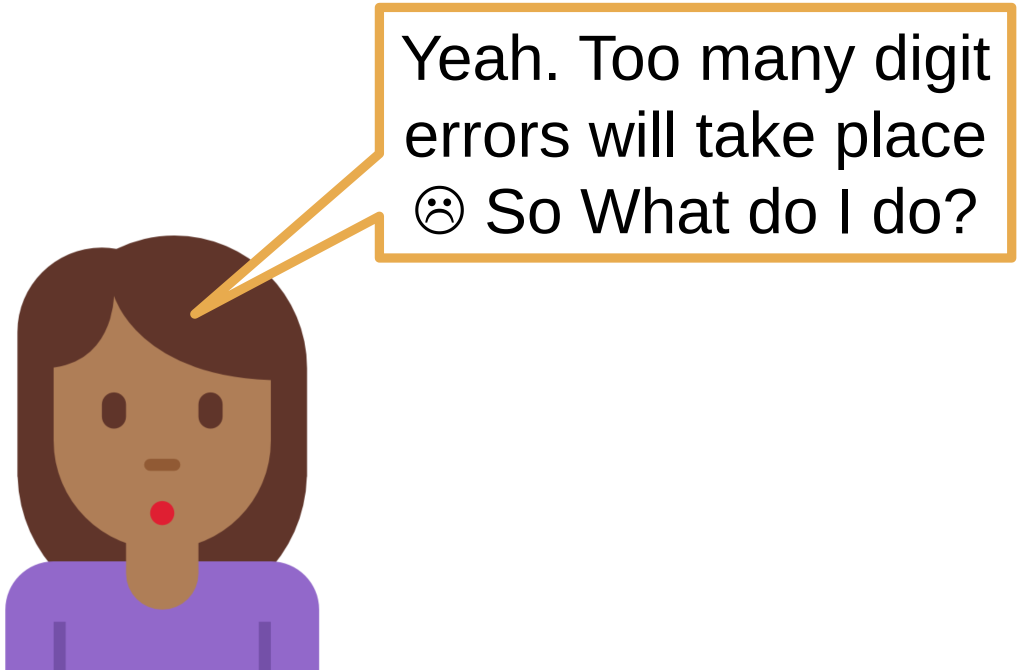
```
#include <stdio.h>
int main(){
    int a, b;
    scanf(“%d.%d”, &a, &b);
    return 0;
}
```

A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal


Yeah. Too many digit errors will take place ☹ So What do I do?

```
#include <stdio.h>
int main(){
    int a, b;
    scanf(“%d.%d”, &a, &b);
    return 0;
}
```

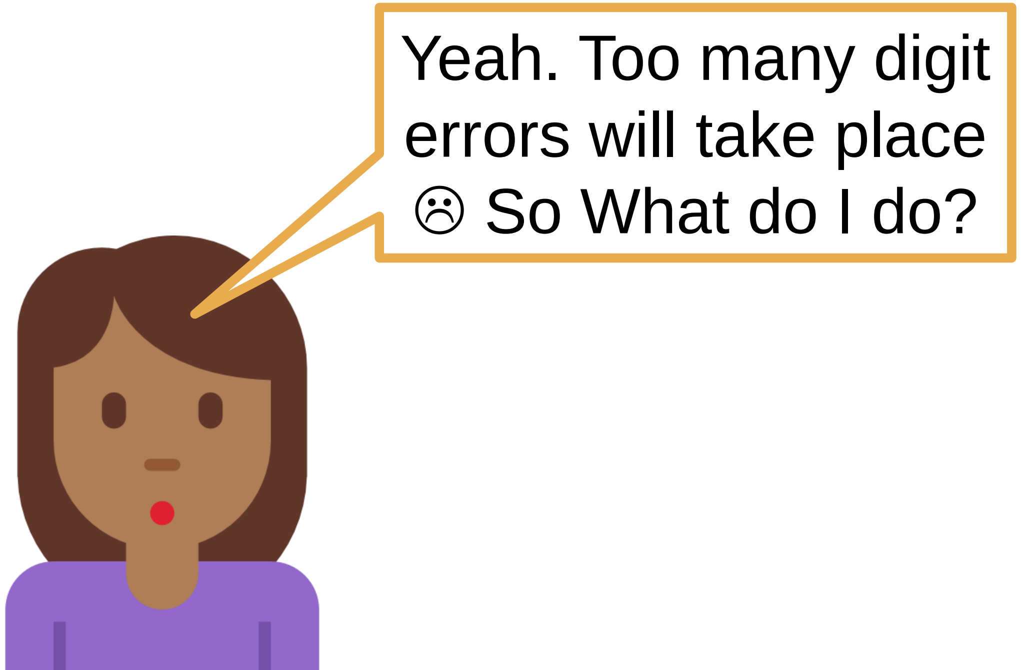
A Cute Trick for Higher Precision

Want to read a number like 123456789.987654321?

Don’t read it as a float or double or even long double

Read it as two int variables, separated by a decimal

Can use long too!


Yeah. Too many digit errors will take place ☹ So What do I do?

```
#include <stdio.h>
int main(){
    int a, b;
    scanf(“%d.%d”, &a, &b);
    return 0;
}
```

Two New Shortcuts

Two New Shortcuts

Incrementing a variable by one

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

```
sum = sum – 1;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

```
sum = sum – 1;
sum--;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Remember, don’t need to write sum = sum++; or sum = ++sum; sum++; will itself assign the incremented value to the sum variable. Similarly sum--; or –sum;

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

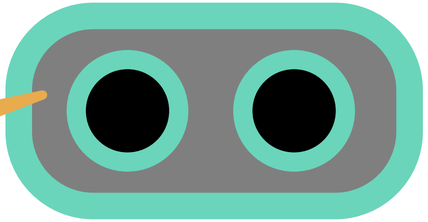
Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

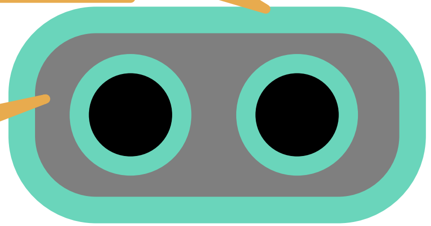
Post-decrement

Pre-decrement



sum++, -- sum etc are valid expressions

Two New Shortcuts



Works only for incrementing or decrementing by one

sum++, -- sum etc are valid expressions

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

Decrementing a variable by one

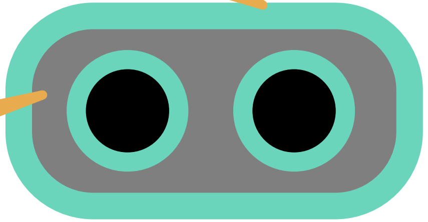
```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

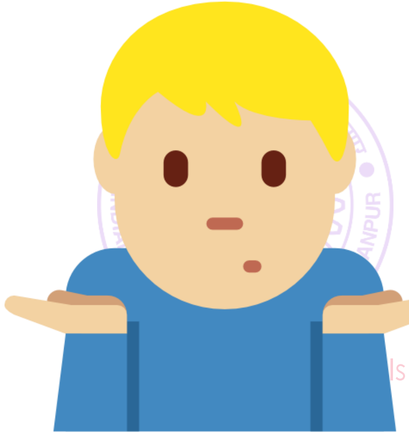
Pre-increment

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

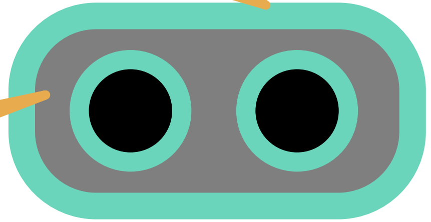
Post-decrement

Pre-decrement



Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

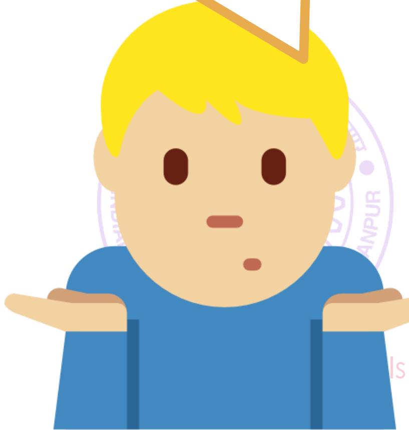
Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

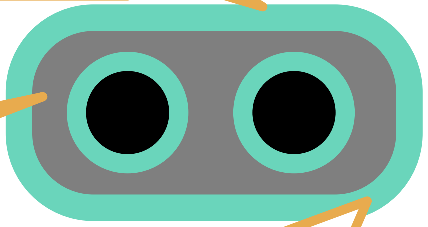
Pre-decrement

What if I want to increment by 2?



Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

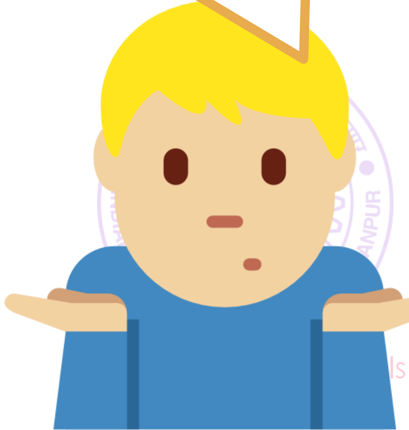
Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

What if I want to increment by 2?



Two New Shortcuts

Works only for incrementing or decrementing by one

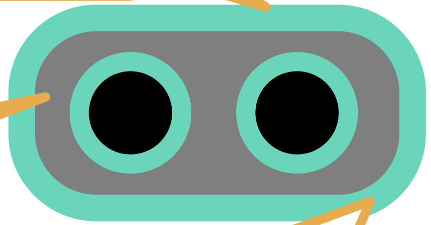
Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

sum++, -- sum etc are valid expressions



There is an entire family of shortcuts

Next slide ☺

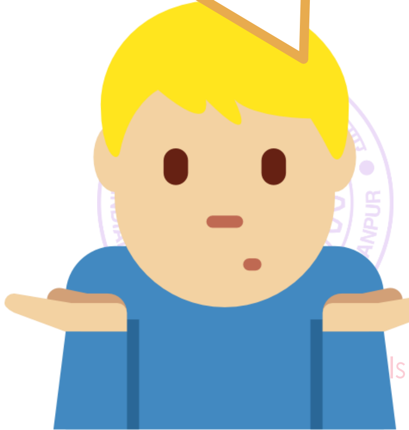
Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

What if I want to increment by 2?



Two New Shortcuts

Works only for incrementing or decrementing by one

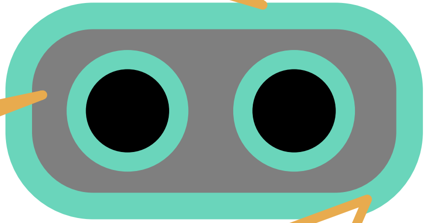
Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

sum++, -- sum etc are valid expressions



There is an entire family of shortcuts

Next slide ☺

Decrementing a variable by one

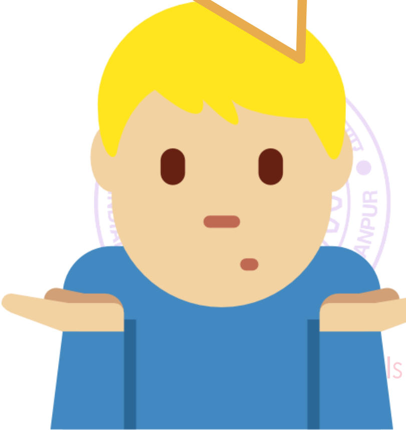
```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

Difference between post and pre?

What if I want to increment by 2?



Two New Shortcuts

Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

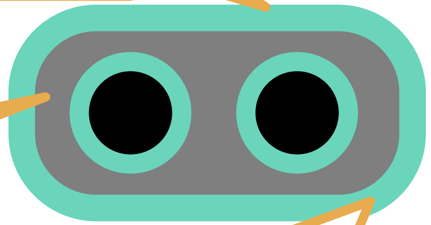
Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

Remember expressions generate values



Works only for incrementing or decrementing by one

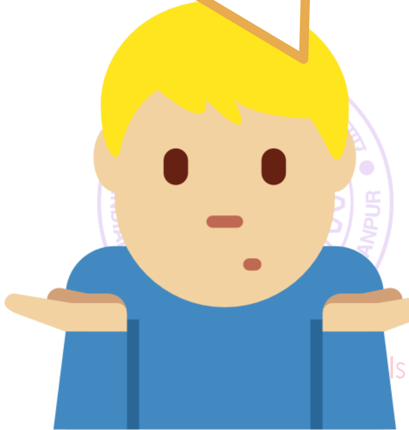
sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Next slide ☺

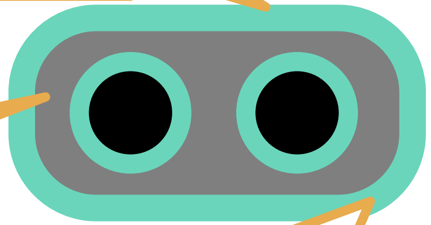
Difference between post and pre?

What if I want to increment by 2?



Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Next slide ☺

Incrementing a variable by one

sum = sum + 1;

sum++; Post-increment

++sum; Pre-increment

Decrementing a variable by one

sum = sum – 1;

sum--; Post-decrement

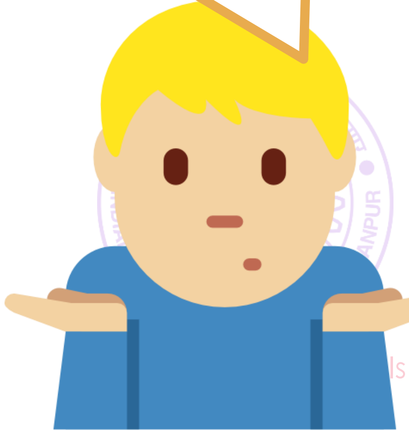
--sum; Pre-decrement

Difference between post and pre?

What if I want to increment by 2?

Remember expressions generate values

int a = 3, b = 5, c;



Two New Shortcuts

Works only for incrementing or decrementing by one

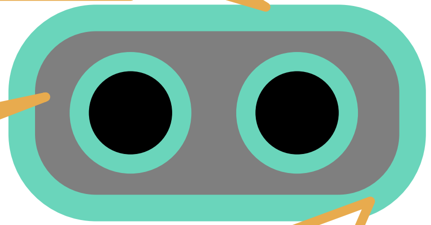
Incrementing a variable by one

```
sum = sum + 1;
sum++;
++sum;
```

Post-increment

Pre-increment

sum++, -- sum etc are valid expressions



There is an entire family of shortcuts

Next slide ☺

Decrementing a variable by one

```
sum = sum – 1;
sum--;
--sum;
```

Post-decrement

Pre-decrement

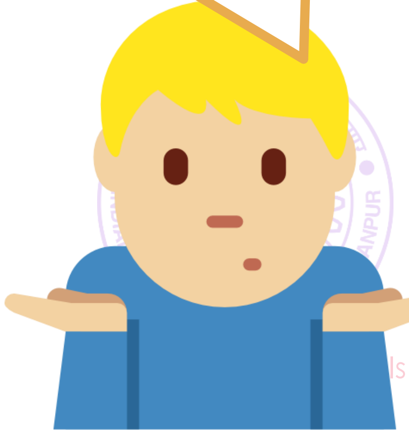
Difference between post and pre?

What if I want to increment by 2?

Remember expressions generate values

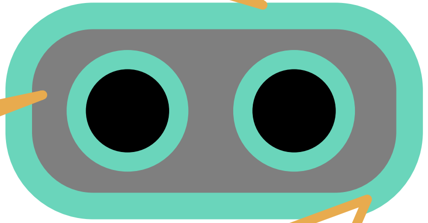
```
int a = 3, b = 5, c;
```

a + b generates value 8, and c = 7 generates value 7



Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Next slide ☺

Incrementing a variable by one

sum = sum + 1;

sum++; Post-increment

++sum; Pre-increment

Decrementing a variable by one

sum = sum – 1;

sum--; Post-decrement

--sum; Pre-decrement

Difference between post and pre?

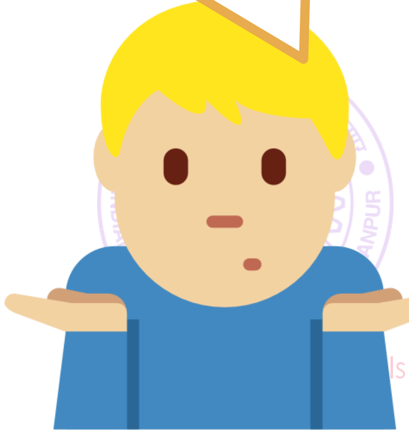
What if I want to increment by 2?

Remember expressions generate values

int a = 3, b = 5, c;

a + b generates value 8, and c = 7 generates value 7

Both a++ and ++a will result in new value of a being 4



Two New Shortcuts

Works only for incrementing or decrementing by one

Incrementing a variable by one

sum = sum + 1;

sum++; Post-increment

++sum; Pre-increment

Decrementing a variable by one

sum = sum – 1;

sum--; Post-decrement

--sum; Pre-decrement

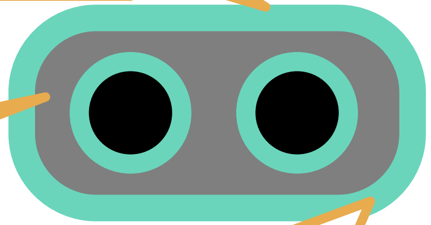
Remember expressions generate values

int a = 3, b = 5, c;

a + b generates value 8, and c = 7 generates value 7

Both a++ and ++a will result in new value of a being 4

But a++ will generate 3 (old value) ++a will generate 4 (new value)



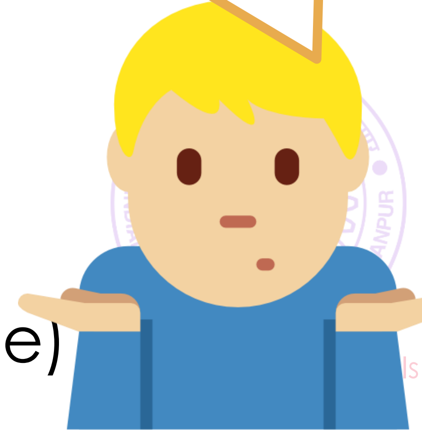
sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Next slide ☺

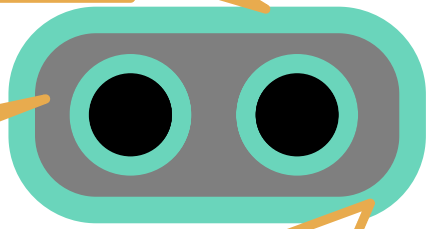
Difference between post and pre?

What if I want to increment by 2?



Two New Shortcuts

Works only for incrementing or decrementing by one



sum++, -- sum etc are valid expressions

There is an entire family of shortcuts

Next slide ☺

Incrementing a variable by one

sum = sum + 1;

sum++; Post-increment

++sum; Pre-increment

Decrementing a variable by one

sum = sum – 1;

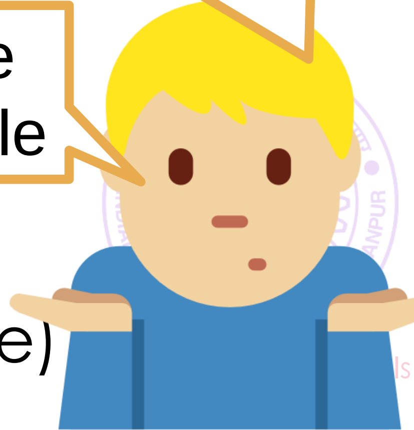
sum--; Post-decrement

--sum; Pre-decrement

Difference between post and pre?

What if I want to increment by 2?

Show me an example



Remember expressions generate values

int a = 3, b = 5, c;

a + b generates value 8, and c = 7 generates value 7

Both a++ and ++a will result in new value of a being 4

But a++ will generate 3 (old value) ++a will generate 4 (new value)

Okay … many more new shortcuts

```
a = a + b;
    a += b;
a = a – b;
    a -= b;
a = a * b;
    a *= b;
a = a / b;
    a /= b;
a = a % b;
    a %= b;
```

BODMAS table has new members

BODMAS table has new members

Operator Name	Symbol/Sign	Associativity
Bracket, **Post increment/decrement**	(), **++**, **--**	Left
Unary negation, **Pre increment/decrement**	-, **++**, **--**	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment, **Compound assignment**	=, **+=**, **-=**, ***=**, **/=**, **%=**	Right

BODMAS table has new members

HIGH PRECEDENCE

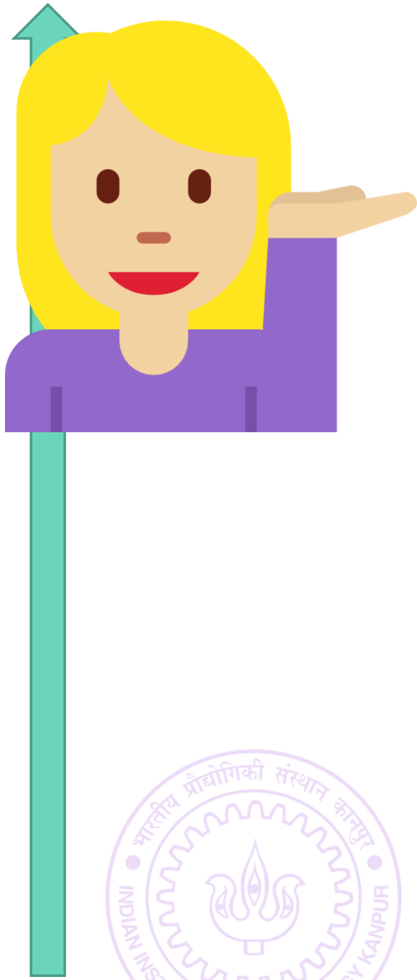
Operator Name	Symbol/Sign	Associativity
Bracket, **Post increment/decrement**	(), **++**, **--**	Left
Unary negation, **Pre increment/decrement**	-, **++**, **--**	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment, **Compound assignment**	=, **+=**, **-=**, ***=**, **/=**, **%=**	Right

LOW PRECEDENCE

BODMAS table has new members

Operator Name	Symbol/Sign	Associativity
Bracket, **Post increment/decrement**	(), **++**, **--**	Left
Unary negation, **Pre increment/decrement**	-, **++**, **--**	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment, **Compound assignment**	=, **+=**, **-=**, ***=**, **/=**, **%=**	Right

HIGH PRECEDENCE



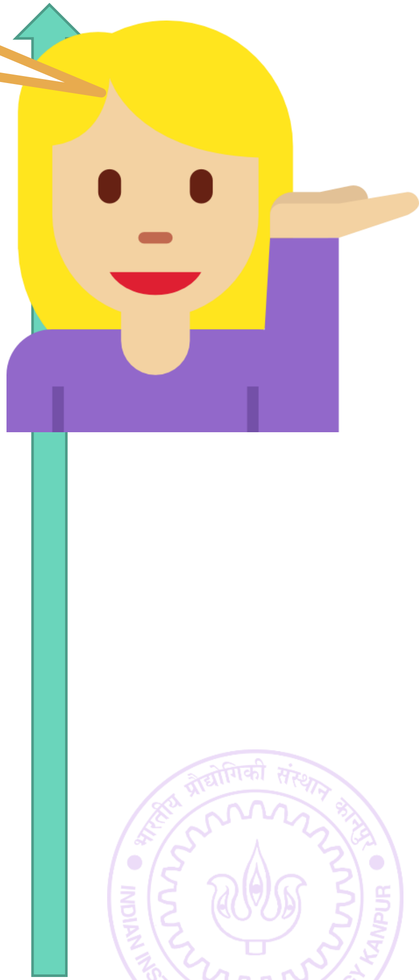
LOW PRECEDENCE

BODMAS table has new members

Write this table down in your notebook. Allowed in labs, quizzes, exams. No need to memorize.

Operator Name		
Bracket, **Post increment/decrement**	(), **++**, **--**	Left
Unary negation, **Pre increment/decrement**	-, **++**, **--**	Right
Multiplication/division/ remainder	*, /, %	Left
Addition/subtraction	+, -	Left
Assignment, **Compound assignment**	=, **+=**, **-=**, ***=**, **/=**, **%=**	Right

HIGH PRECEDENCE



LOW PRECEDENCE

Back to If Statements

- How to write more powerful choices
  - If something happens do this, else do that
  - If this AND this happens do this, else if that OR that happens, do that
- How to avoid common errors

The if statement

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

The if statement

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

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

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf("%d",&m);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodbye

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

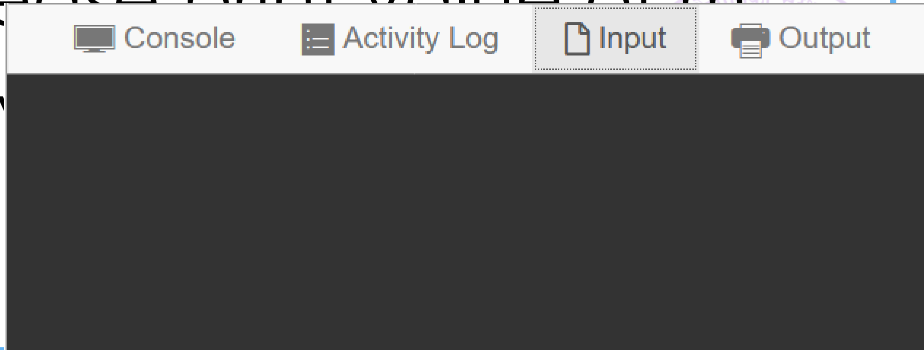
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodby



The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

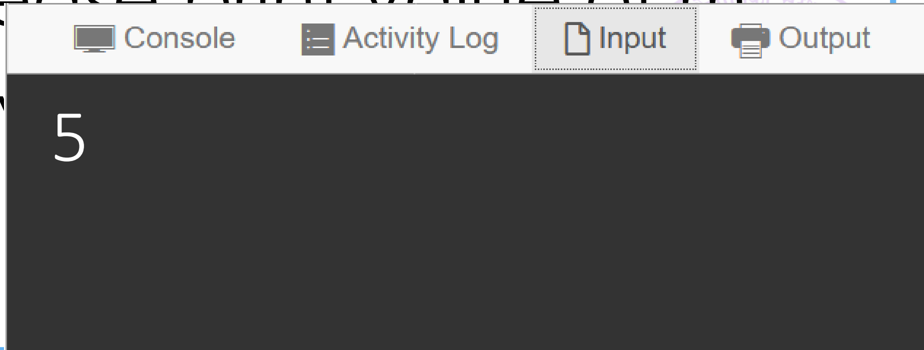
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodby

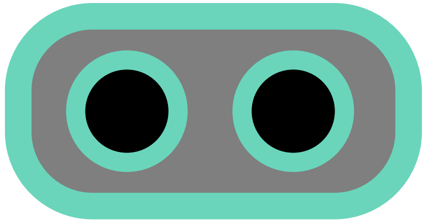


```
5
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

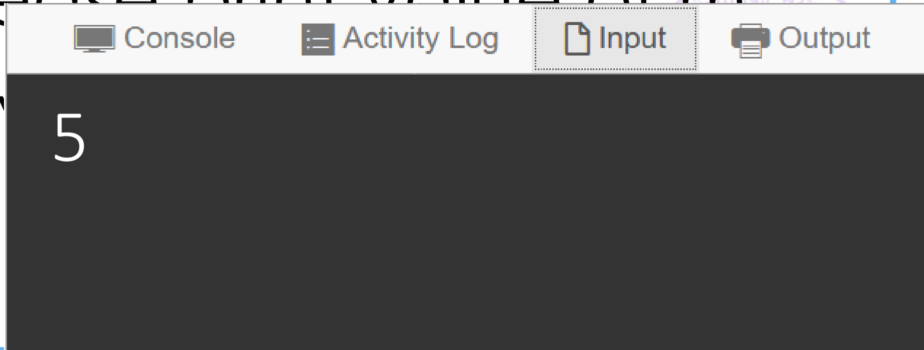
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

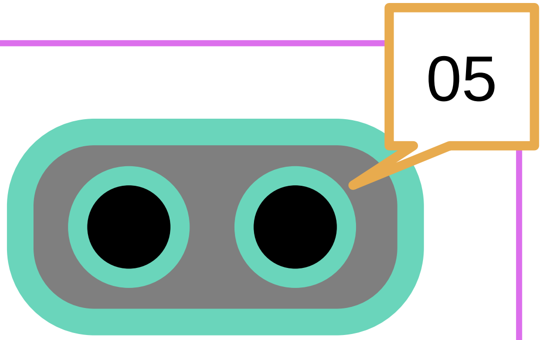
Goodby



```
5
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER



```
#include <stdio.h>
int main(){
    int m;
    scanf("%d",&m);
    if(m < 10){
        printf("0");
    }
    printf("%d",m);
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

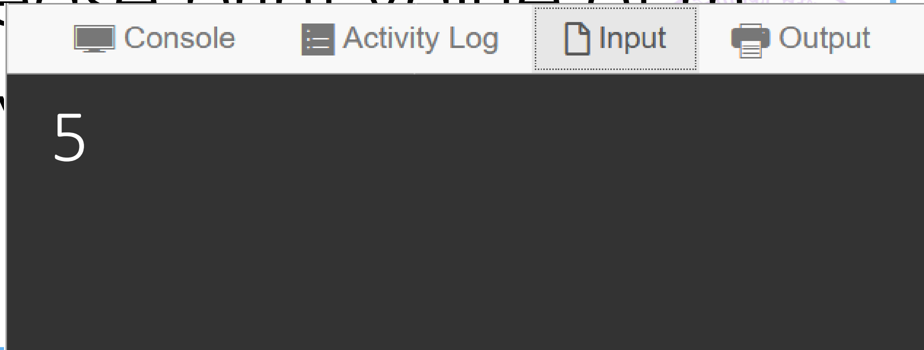
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodby

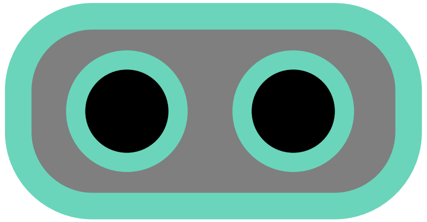


```
5
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

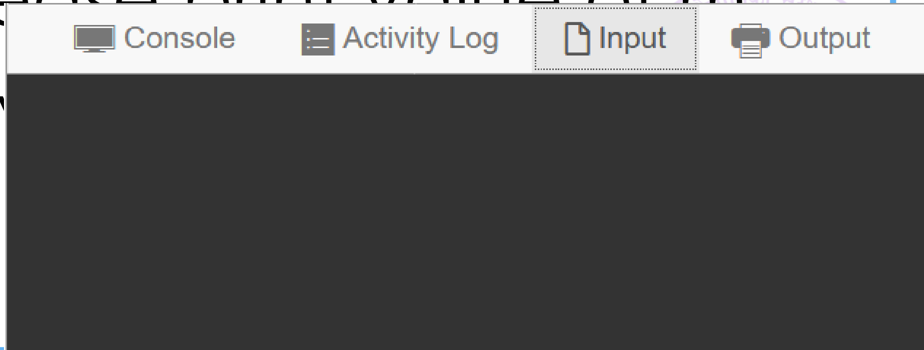
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

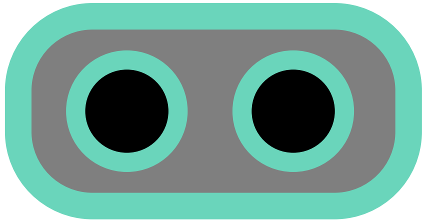
Goodb



The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf("%d",&m);
    if(m < 10){
        printf("0");
    }
    printf("%d",m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

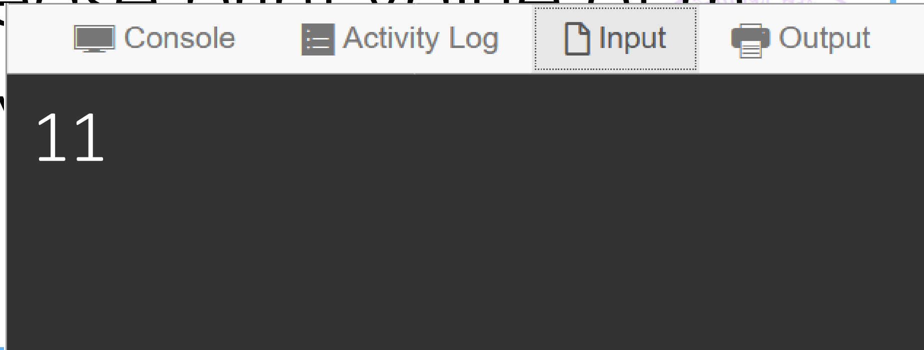
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

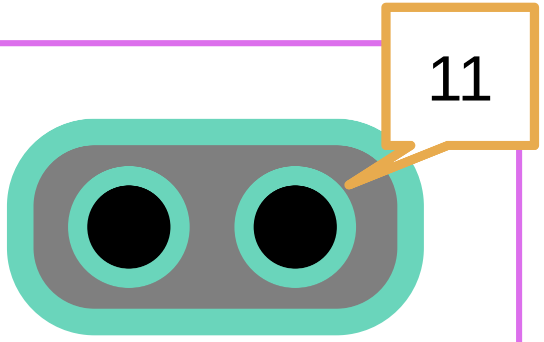
Goodby



```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER



```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

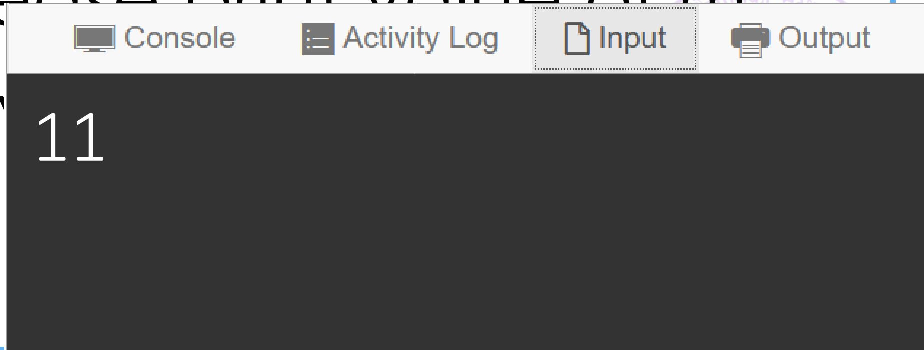
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m.

Goodby

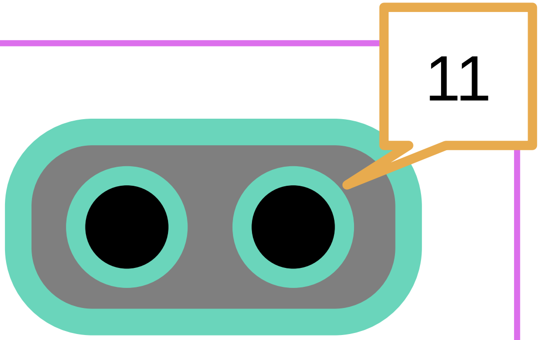


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf("%d",&m);
    if(m < 10){
        printf("0");
    }
    printf("%d",m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

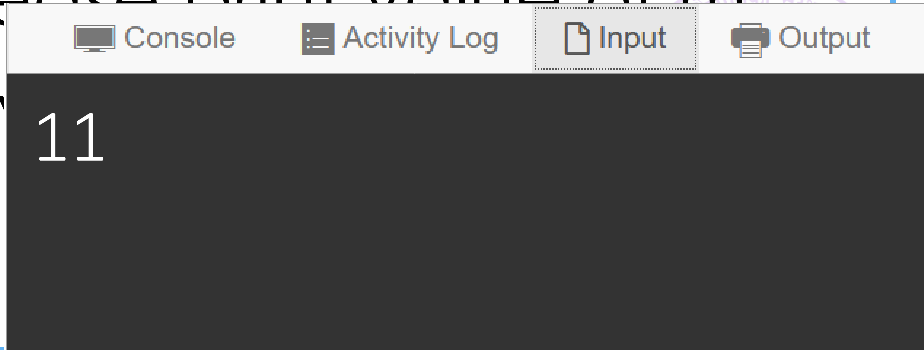
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m.

Goodby

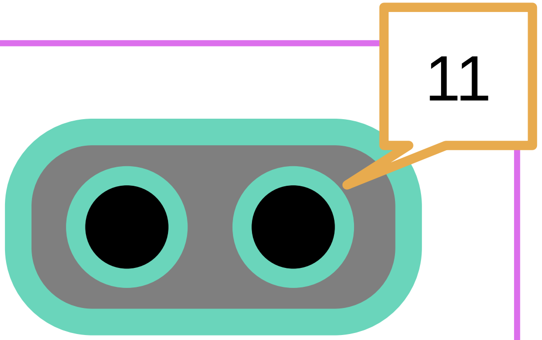


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,
}
```



These curly brackets are used to tell Mr. C what all we want him to do if m < 10

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

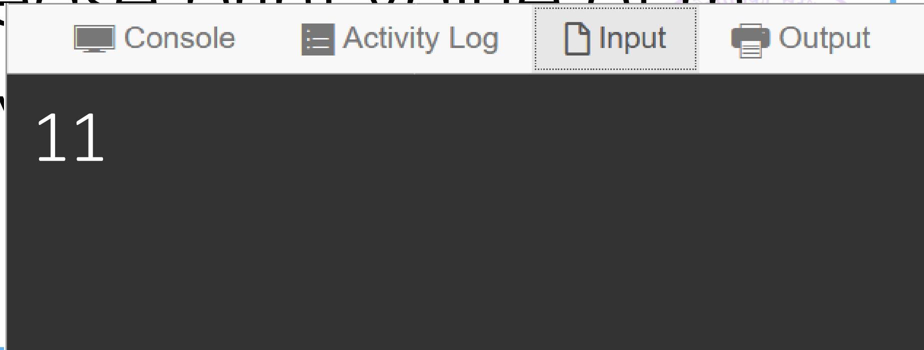
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m.

Goodby

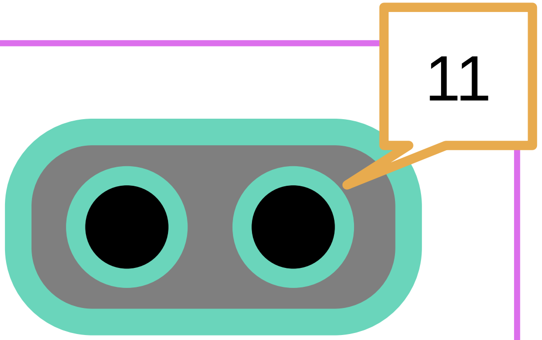


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

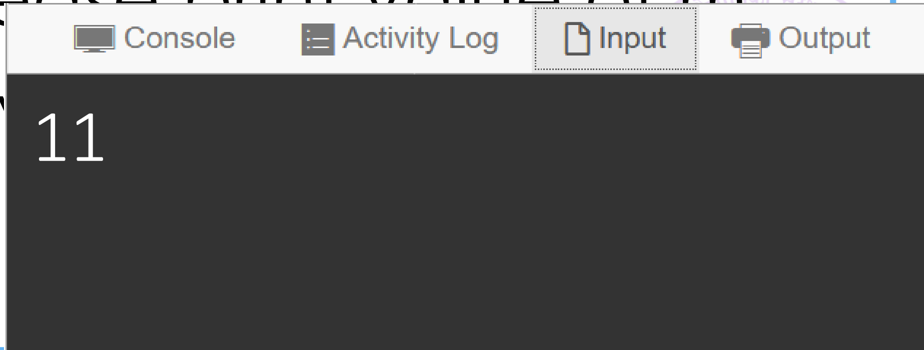
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m.

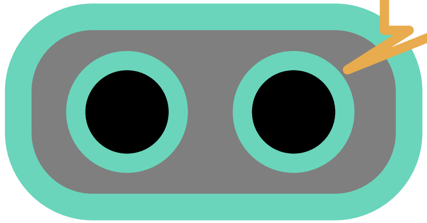
Goodb



```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER



11

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”)
    }
    printf(“%d”,m);
}
```

This statement is always executed whether m < 10 or not

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

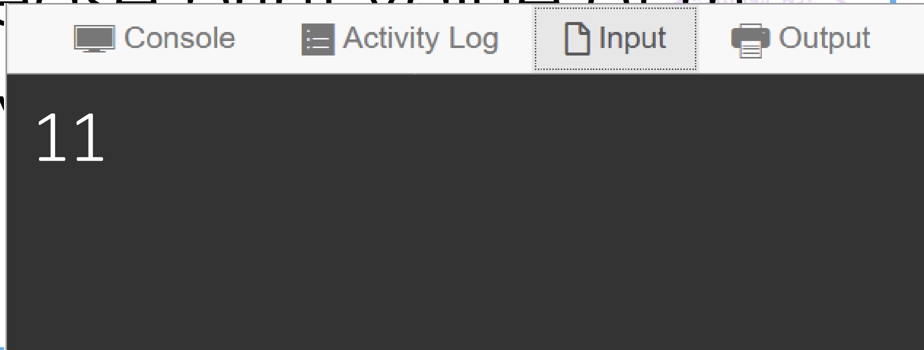
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodby

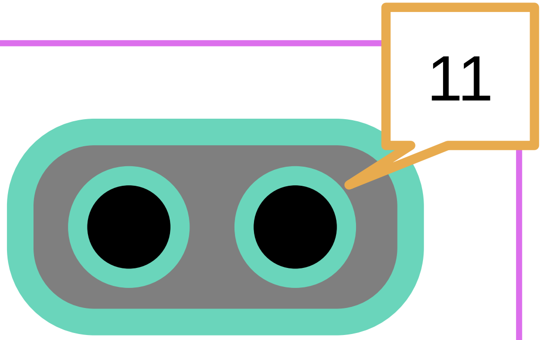


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```



HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

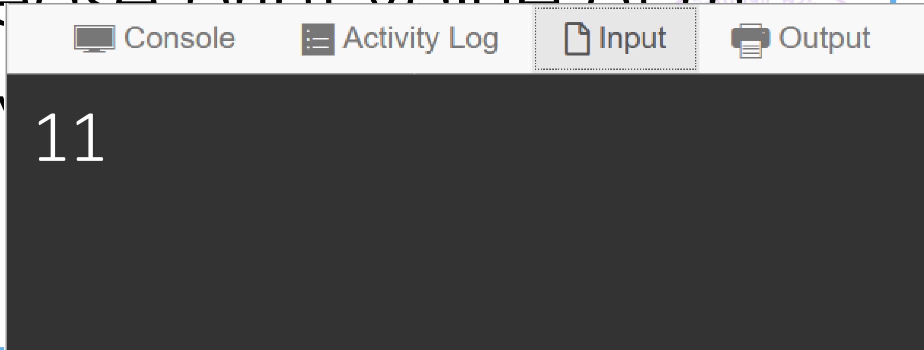
m is an integer variable.

Please ask me for value of m.

If the value of m is less than 10, then please print a 0

Now please print value of m

Goodby

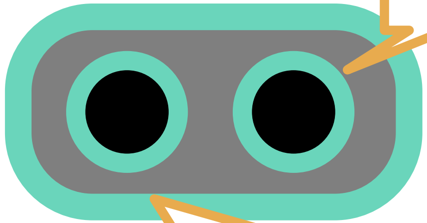


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf("%d",&m);
    if(m < 10){
        printf("0");
    }
    printf("%d",m);
}
```



11

Brackets essential in case you want me to do many things if m < 10, not just print one 0

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

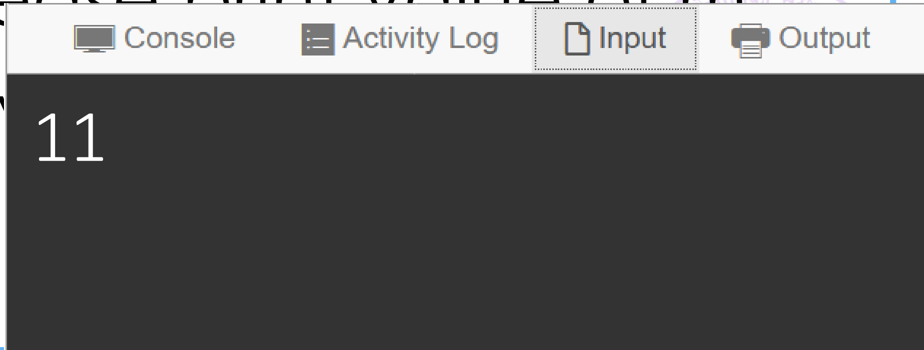
m is an integer variable.

sk me for value of m.

ue of m is less than 10,
then please print a 0

Now please print value of m

Goodby

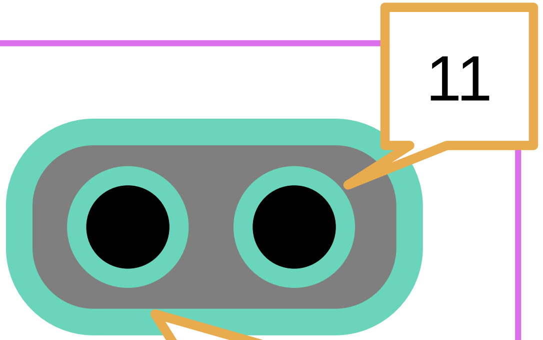


```
11
```

The if statement

HOW WE MUST SPEAK TO MR. COMPILER

```
#include <stdio.h>
int main(){
    int m;
    scanf(“%d”,&m);
    if(m < 10){
        printf(“0”);
    }
    printf(“%d”,m);
}
```



Brackets essential in case you want me to do many things if m < 10, not just print one 0

Missing brackets - common mistake

HOW WE USUALLY SPEAK TO A HUMAN

Do you speak English?

Hello

m is an integer variable.

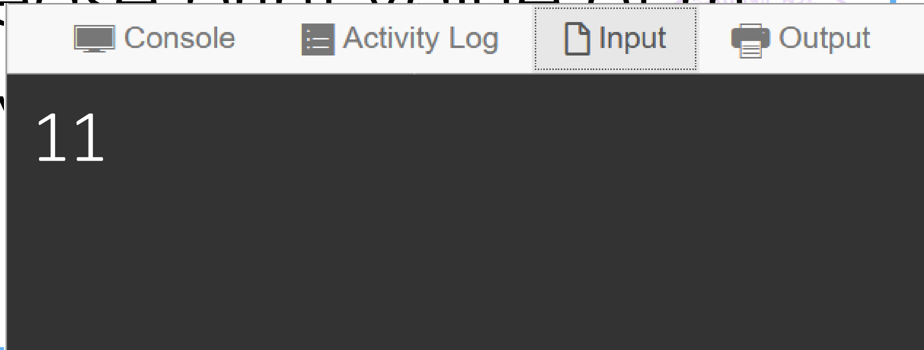
sk me for value of m.

ue of m is less than 10,

ease print a 0

ease print value of m

Goodb



```
11
```

Freedom of Choice

Freedom of Choice

If m is less than 10

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

Freedom of Choice

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

If p is equal to q

```
if(p == q){ ... }
```

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

```
    if(p == q){ ... }
```

If m is not equal to 15

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

Freedom of Choice

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

If p is equal to q

```
if(p == q){ ... }
```

If m is not equal to 15

```
if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

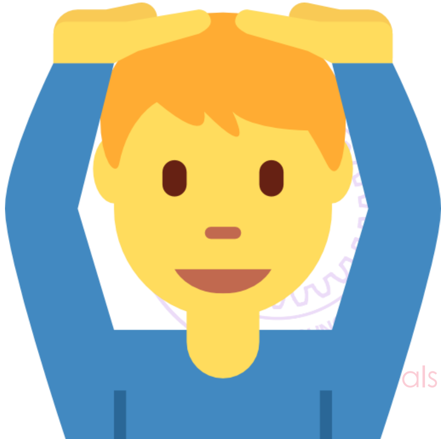
If p is equal to q

```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!



Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

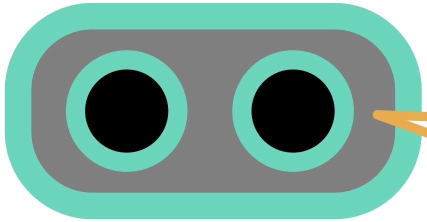
```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

Freedom of Choice



If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

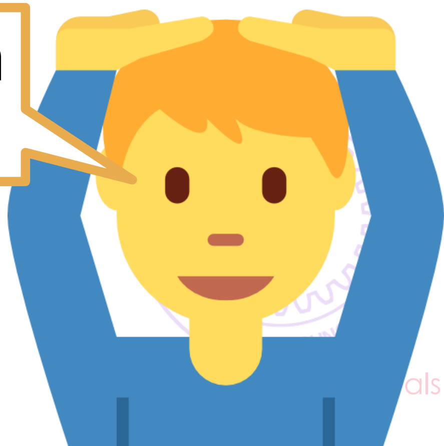
If p is equal to q

```
    if(p == q){ ... }
```

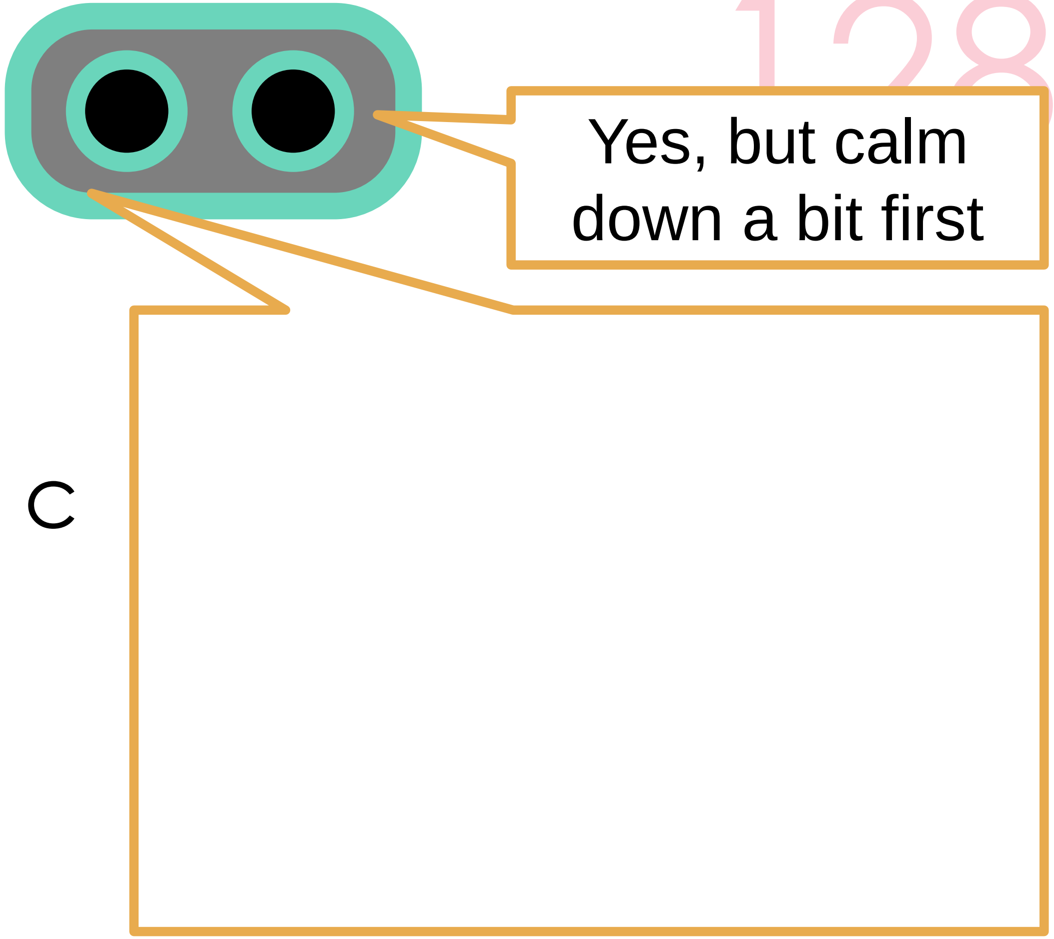
If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!



Freedom of Choice



If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

If p is equal to q

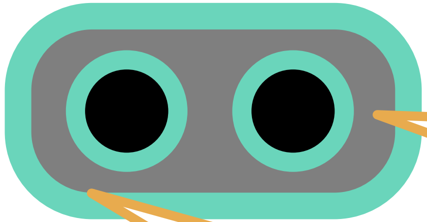
```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

Freedom of Choice



Yes, but calm down a bit first

WORDS OF CAUTION

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

If p is equal to q

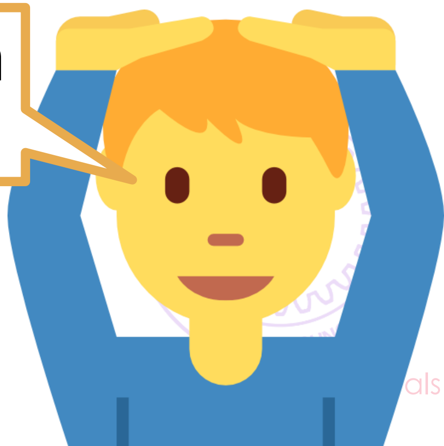
```
if(p == q){ ... }
```

If m is not equal to 15

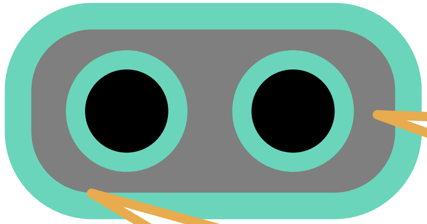
```
if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

I can do so much with these!



Freedom of Choice



Yes, but calm down a bit first

WORDS OF CAUTION

Do not forget brackets

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

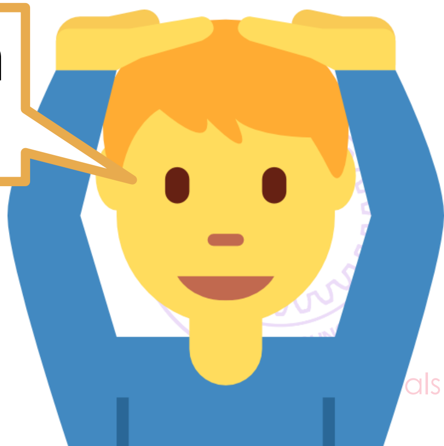
If p is equal to q

```
if(p == q){ ... }
```

If m is not equal to 15

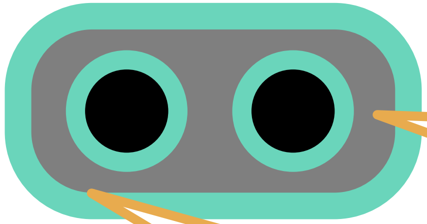
```
if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!



I can do so much with these!

Freedom of Choice



Yes, but calm down a bit first

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

If p is equal to q

```
if(p == q){ ... }
```

If m is not equal to 15

```
if(m != 15){ ... }
```

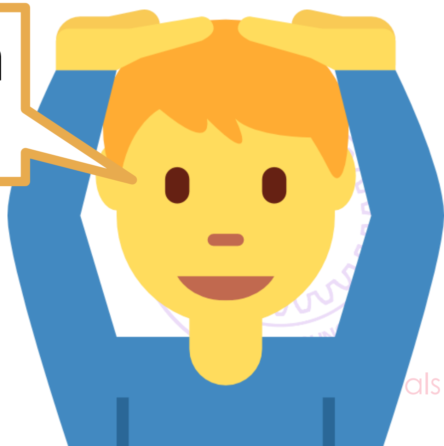
<, <=, >, >=, ==, != 6 new operators for Mr C!

WORDS OF CAUTION

Do not forget brackets

Note: it is p == q, not p = q

I can do so much with these!



Freedom of Choice

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

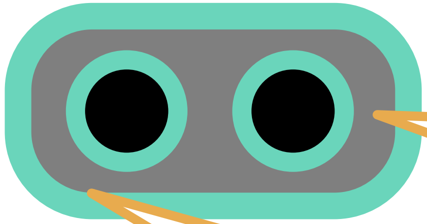
If p is equal to q

```
if(p == q){ ... }
```

If m is not equal to 15

```
if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!



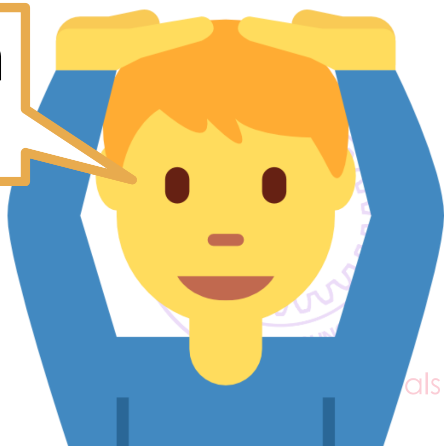
Yes, but calm down a bit first

WORDS OF CAUTION

Do not forget brackets

Note: it is p == q, not p = q

Risky to use p == q and p != q with float, double



I can do so much with these!

Freedom of Choice

If m is less than 10

```
    if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
    if(a >= b+c){ ... }
```

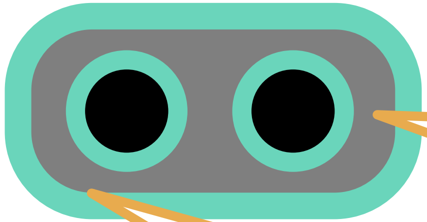
If p is equal to q

```
    if(p == q){ ... }
```

If m is not equal to 15

```
    if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!



Yes, but calm down a bit first

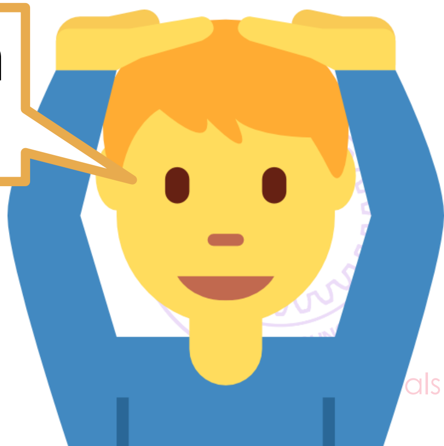
WORDS OF CAUTION

Do not forget brackets

Note: it is p == q, not p = q

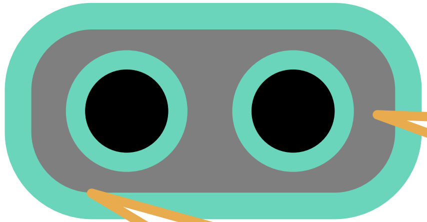
Risky to use p == q and p != q with float, double

Use these freely with int, long



I can do so much with these!

Freedom of Choice



Yes, but calm down a bit first

If m is less than 10

```
if(m < 10){ ... }
```

If a is greater than or equal to b + c

```
if(a >= b+c){ ... }
```

If p is equal to q

```
if(p == q){ ... }
```

If m is not equal to 15

```
if(m != 15){ ... }
```

<, <=, >, >=, ==, != 6 new operators for Mr C!

WORDS OF CAUTION

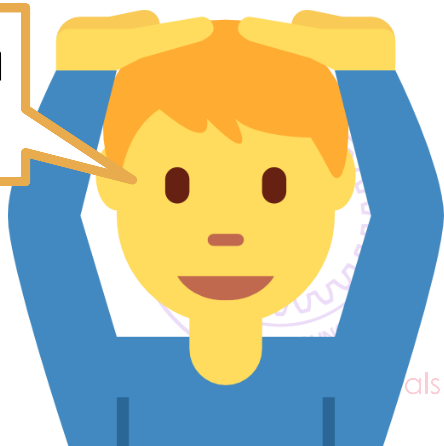
Do not forget brackets

Note: it is p == q, not p = q

Risky to use p == q and p != q with float, double

Use these freely with int, long

<, <=, >, >= fine with int, long, float, and double



I can do so much with these!

More powerful conditionals

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

  Say Small to me. Then increment c by 1

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

More powerful conditionals

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf("Small");
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf("Small");
    c++;
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf("Small");
    c++;
}else{
    printf("Big");
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf("Small");
    c++;
}else{
    printf("Big");
    c--;
}
printf("Goodbye");
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

More powerful conditionals

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to me

Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf("Small");
    c++;
}else{
    printf("Big");
    c--;
}
printf("Goodbye");
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

  Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

  Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye

Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
 if((a + b)/2.0 < 0.5){printf(“Small”);
c++;}else{printf(“Big”);c--;}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye

Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
 if((a + b)/2.0 < 0.5){printf(“Small”);
c++;}else{printf(“Big”);c--;}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye to

Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5){
    printf(“Small”);
    c++;
}else{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

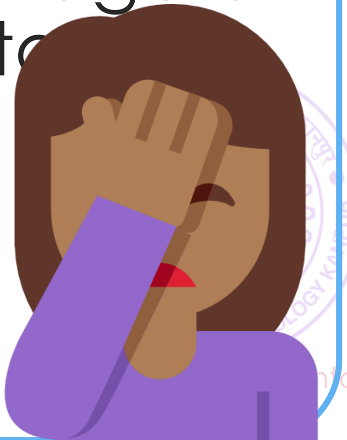
If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye t



Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5)
{
    printf(“Small”);
    c++;
}
else
{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

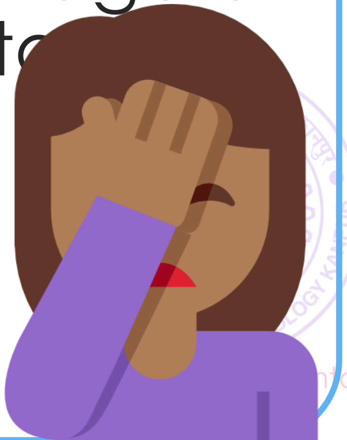
If the average of a and b is less than 0.5, do the following

  Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

  Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye t



Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5)
{
    printf(“Small”);
    c++;
}
else
{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

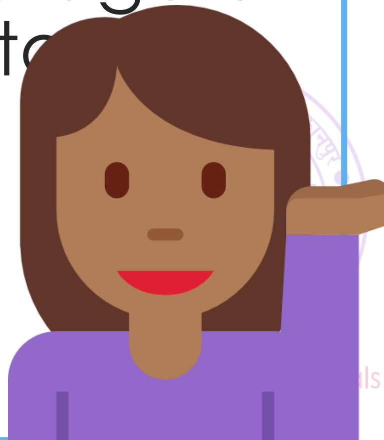
If the average of a and b is less than 0.5, do the following

Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

No matter what the average of a and b, say Goodbye t



Writing pretty code is an art

HOW WE MUST SPEAK TO MR. COMPILER

```
if((a + b)/2.0 < 0.5)
{
    printf(“Small”);
    c++;
}
else
{
    printf(“Big”);
    c--;
}
printf(“Goodbye”);
```

HOW WE USUALLY SPEAK TO A HUMAN

If the average of a and b is less than 0.5, do the following

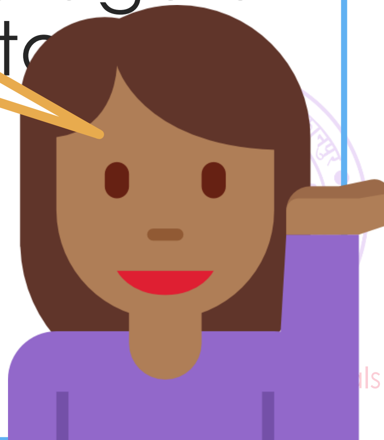
Say Small to me. Then increment c by 1

Else if average of a and b is not less than 0.5, do the following

Say Big to me. Then decrement c by 1

N verage of

a o t



Artists sometimes differ on what is prettier

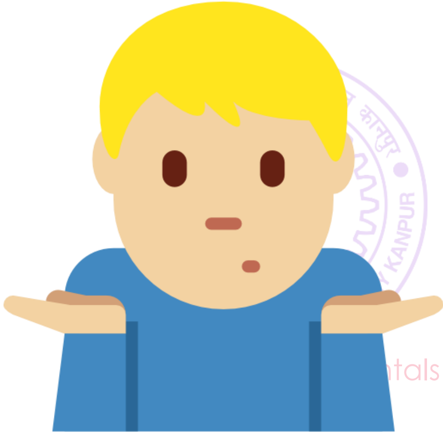
Even more powerful conditionals

Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){



}else{



}
```

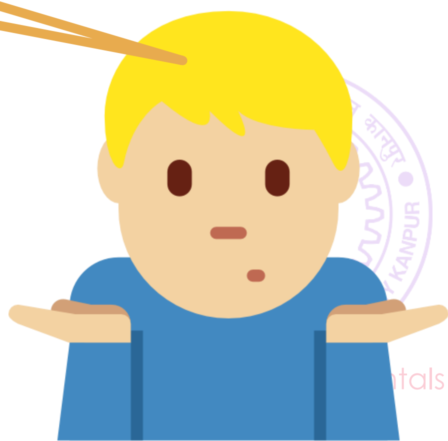
Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){



}else{



}
```



Even more powerful conditionals

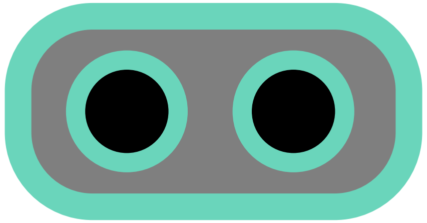
```
if((a + b)/2.0 < 0.5){



}else{



}
```

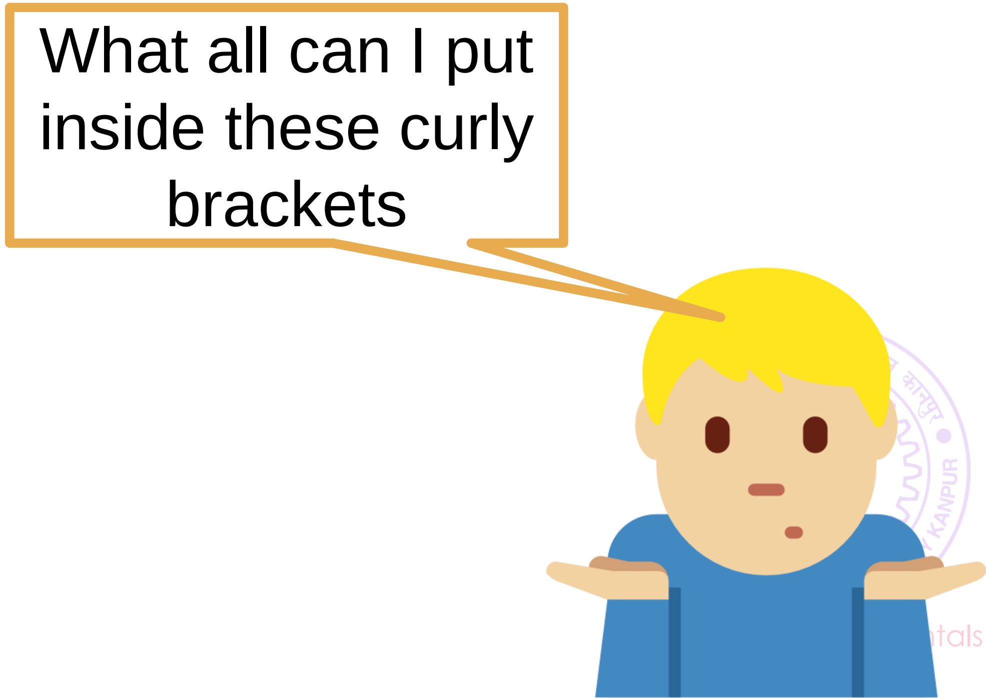


What all can I put inside these curly brackets

Even more powerful conditionals

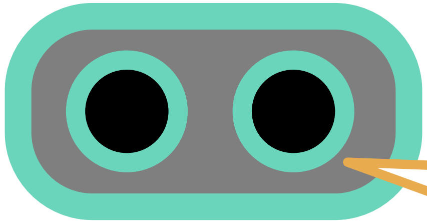
```
if((a + b)/2.0 < 0.5){



}else{



}
```





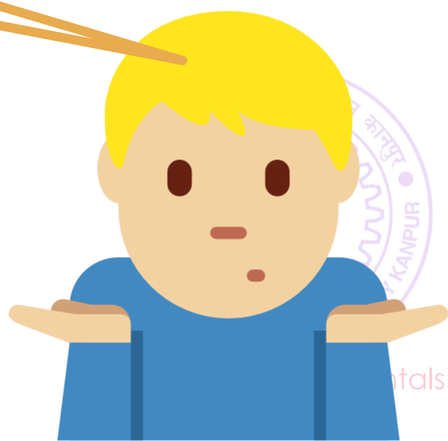
Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){



}else{



}
```



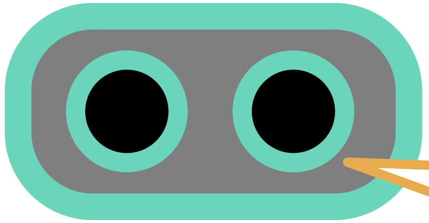
Any number of statements!

What all can I put inside these curly brackets



Even more powerful conditionals

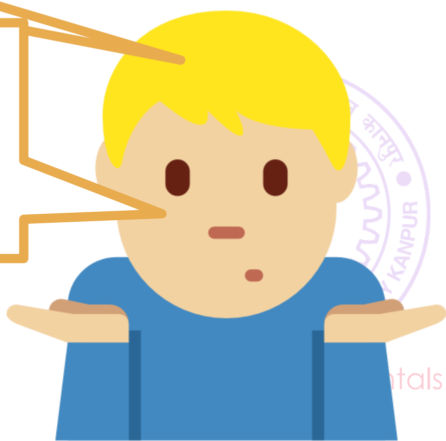
```
if((a + b)/2.0 < 0.5){



}else{



}
```



Any number of statements!

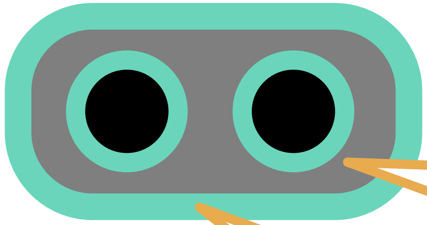
What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){



}else{



}
```

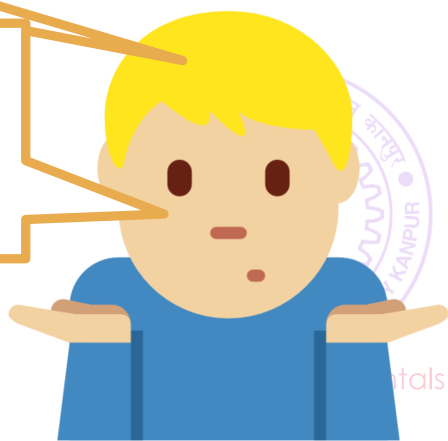


Any number of statements!

Of course! Called a *nested* if-else statement

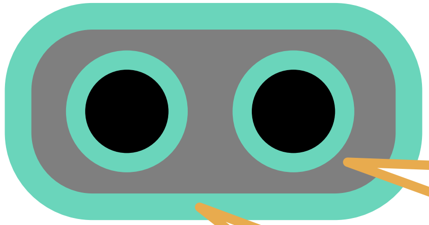
What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
        if((a + b)/2.0 < 0.25){
            printf("Very small");
        }else{
            printf("Small");
        }
}else{


}
```

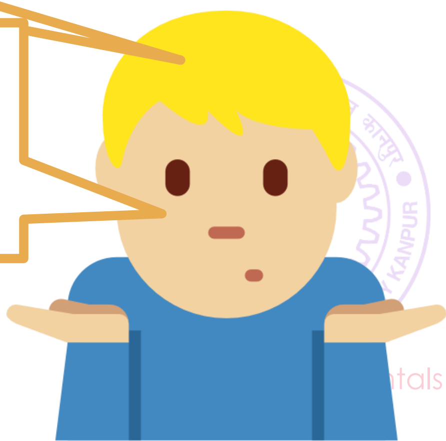


Any number of statements!

Of course! Called a *nested* if-else statement

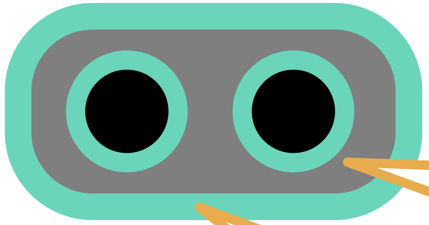
What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf("Very small");
    }else{
        printf("Small");
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf("Very Big");
    }else{
        printf("Big");
    }
}
```

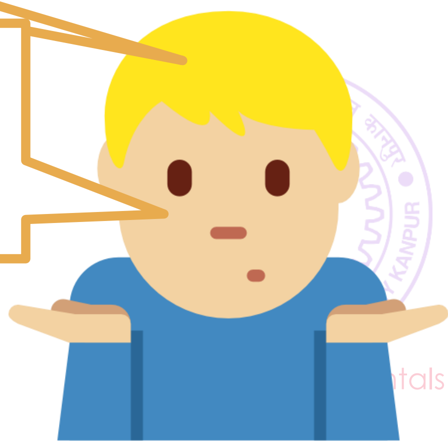


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

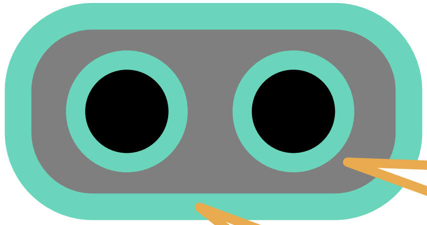
Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf("Very small");
    }else{
        printf("Small");
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf("Very Big");
    }else{
        printf("Big");
    }
}
```

Only < 0.5

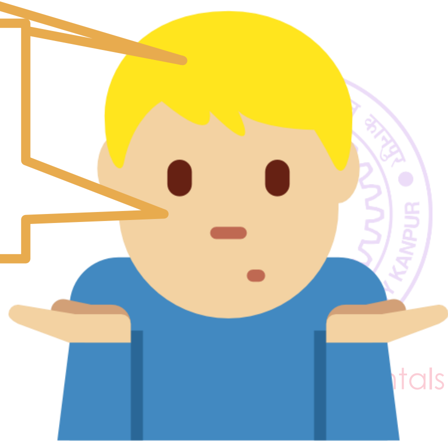


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?

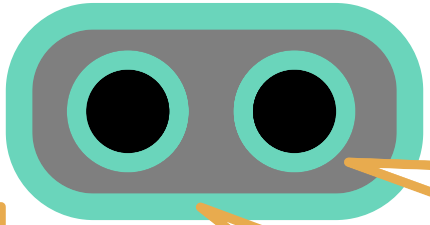


Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf("Very small");
    }else{
        printf("Small");
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf("Very Big");
    }else{
        printf("Big");
    }
}
```

Only < 0.5

< 0.5 as well as < 0.25

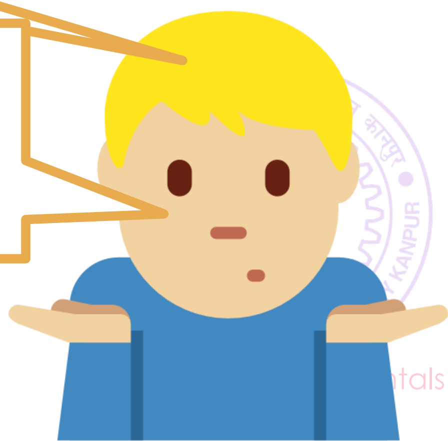


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?



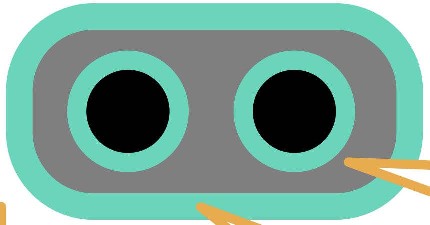
Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf(“Very small”);
    }else{
        printf(“Sm
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf(“Very Big”);
    }else{
        printf(“Big”);
    }
}
```

Only < 0.5

< 0.5 as well as < 0.25

< 0.5 but not < 0.25

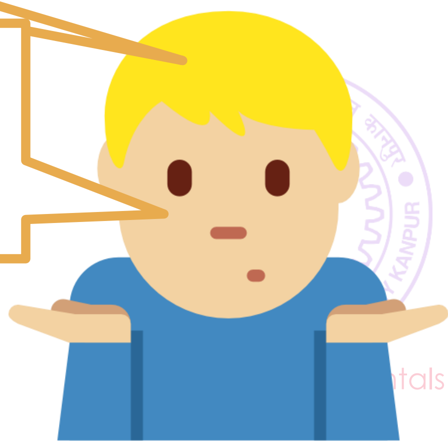


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

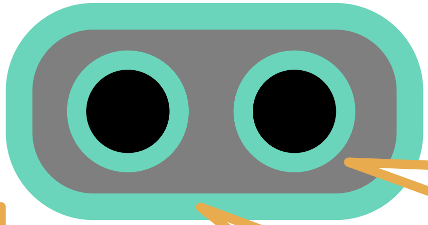
```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf(“Very small”);
    }else{
        printf(“Sm
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf(“Very Big”);
    }else{
        printf(“Big”);
    }
}
```

Only < 0.5

< 0.5 as well as < 0.25

< 0.5 but not < 0.25

Only >= 0.5

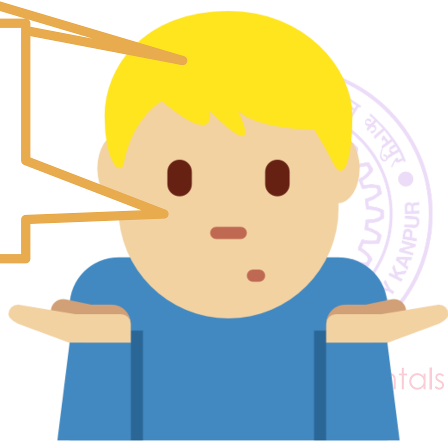


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf(“Very small”);
    }else{
        printf(“Sm
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf(“Very Big”);
    }else{
        printf(“Big”);
    }
}
```

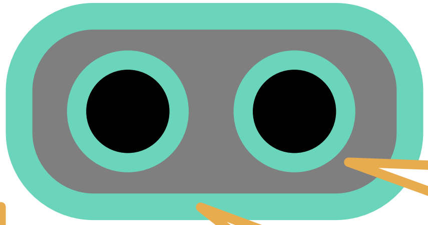
Only < 0.5

< 0.5 as well as < 0.25

< 0.5 but not < 0.25

Only >= 0.5

>= 0.5 as well as > 0.75

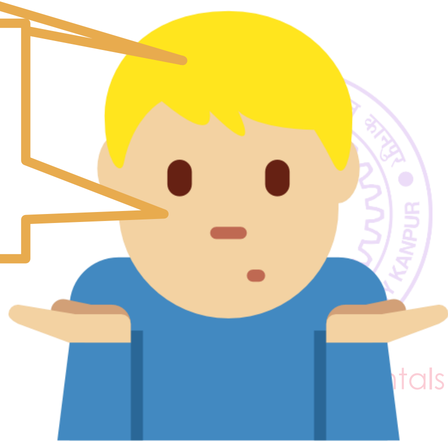


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?



Even more powerful conditionals

```
if((a + b)/2.0 < 0.5){
    if((a + b)/2.0 < 0.25){
        printf(“Very small”);
    }else{
        printf(“Sm
    }
}else{
    if((a + b)/2.0 > 0.75){
        printf(“Very Big”);
    }else{
        printf(“Big
    }
}
```

Only < 0.5

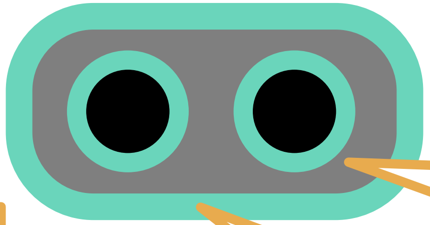
< 0.5 as well as < 0.25

< 0.5 but not < 0.25

Only >= 0.5

>= 0.5 as well as > 0.75

>= 0.5 but not > 0.75

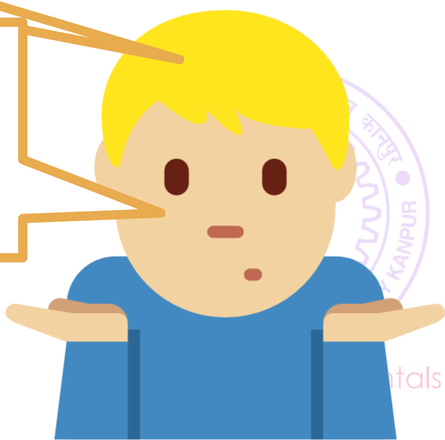


Any number of statements!

Of course! Called a *nested* if-else statement

What all can I put inside these curly brackets

Can I put another if-else condition inside?



The Goldilocks Challenge

The Goldilocks Challenge

Write a program to take a temperature and print

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
```

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

```
if(temp < 22){


}else{
```

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

```
if(temp < 22){


}else{








}
```

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{






}
```

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){




}
```

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);




}
```

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

```
if(temp < 22){
    printf("Too Cold");
}else{
    if (22 <= temp <= 27){
        printf("Just Right");
    }else{



}
```

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22
Just Right if between 22 and 27
Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);


}
```

The Goldilocks Challenge

Write a program to take a temperature and print

- Too Cold if temperature is below 22
- Just Right if between 22 and 27
- Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);
    }
}
```

The Goldilocks Challenge

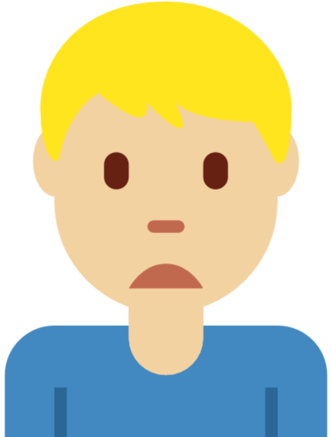
Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf("Too Cold");
}else{
    if (22 <= temp <= 27){
        printf("Just Right");
    }else{
        printf("Too Hot");
    }
}
```



The Goldilocks Challenge

Write a program to take a temperature and print

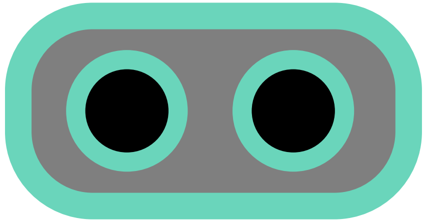
Too Cold if temperature is below 22
Just Right if between 22 and 27
Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);
    }
}
```

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22
Just Right if between 22 and 27
Too Hot if temperature is above 27



```
if(temp < 22){
    printf("Too Cold");
}else{
    if (22 <= temp <= 27){
        printf("Just Right");
    }else{
        printf("Too Hot");
    }
}
```

The Goldilocks Challenge

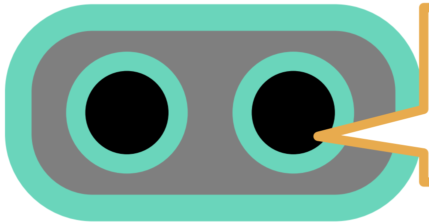
Write a program to take a temperature and print

Too Cold if temperature is below 22

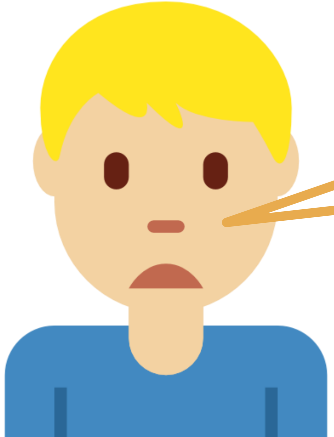
Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf("Too Cold");
}else{
    if (22 <= temp <= 27){
        printf("Just Right");
    }else{
        printf("Too Hot");
    }
}
```



Remember, expressions generate values



What just happened?

The Goldilocks Challenge

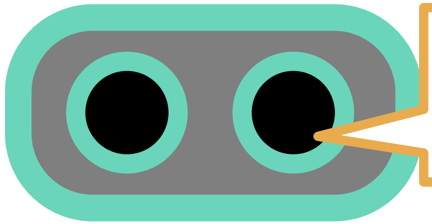
Write a program to take a temperature and print

Too Cold if temperature is below 22

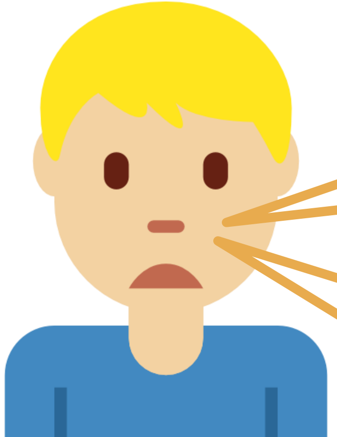
Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf("Too Cold");
}else{
    if (22 <= temp <= 27){
        printf("Just Right");
    }else{
        printf("Too Hot");
    }
}
```



Remember, expressions generate values



What just happened?

What value does temp < 22 generate?

The Goldilocks Challenge

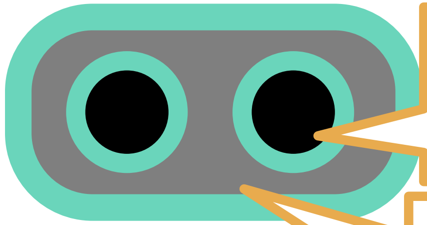
Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

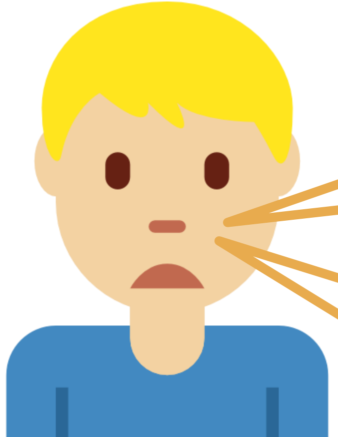
Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);
    }
}
```



Remember, expressions generate values

If temp is less than 22, it generates value 1. If temp is greater than or equal to 22, it generates value 0



What just happened?

What value does temp < 22 generate?

The Goldilocks Challenge

Write a program to take a temperature and print

Too Cold if temperature is below 22

Just Right if between 22 and 27

Too Hot if temperature is above 27

```
if(temp < 22){
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);
    }
}
```

Remember, expressions generate values

If temp is less than 22, it generates value 1. If temp is greater than or equal to 22, it generates value 0

What just happened?

What value does temp < 22 generate?

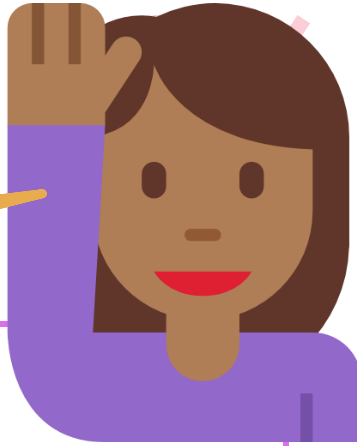
The Goldilocks Challenge

Write a program to take

Too Cold if temperature is bel

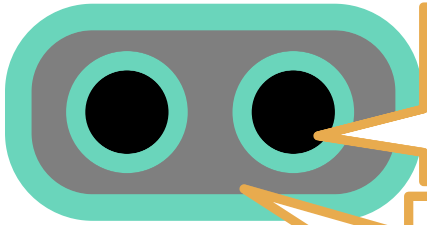
Just Right if between 22 and 27

Too Hot if temperature is above 27



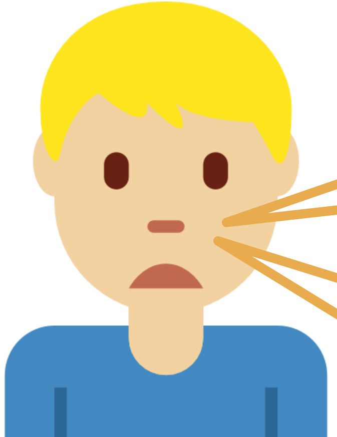
Mr C considers 0 to be FALSE and 1 (or anything non-zero) to be TRUE

```
    printf(“Too Cold”);
}else{
    if (22 <= temp <= 27){
        printf(“Just Right”);
    }else{
        printf(“Too Hot”);
    }
}
```



Remember, expressions generate values

If temp is less than 22, it generates value 1. If temp is greater than or equal to 22, it generates value 0



What just happened?

What value does temp < 22 generate?

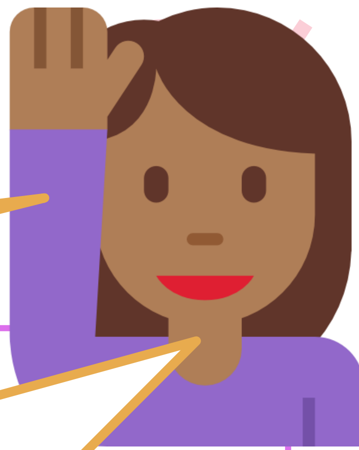
The Goldilocks Challenge

Write a program to take

Too Cold if temperature is bel

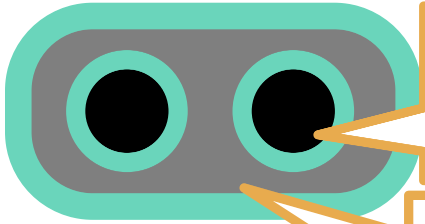
Just Right if between 22 and 27

Too Hot if temperature is above 27



Mr C considers 0 to be FALSE and 1 (or anything non-zero) to be TRUE

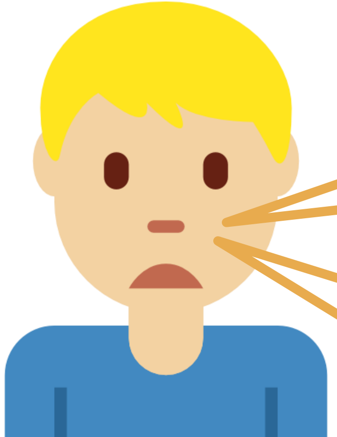
If the expression inside if(…) evaluates to 1 or something non-zero, Mr C executes the if part. If the expression evaluates to 0, Mr C executes the else part



Remember, expressions generate values

If temp is less than 22, it generates value 1. If temp is greater than or equal to 22, it generates value 0

```
if (22 < temp < 27){
    printf(“Just Right”);
}else{
    printf(“Too Hot”);
  }
}
```



What just happened?

What value does temp < 22 generate?

Complex Relational Expressions

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

When we entered 30, Mr C evaluated ((22 <= 30) <= 27)

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

When we entered 30, Mr C evaluated ((22 <= 30) <= 27)

This became (1 <= 27) which is true so the final result is 1

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

When we entered 30, Mr C evaluated ((22 <= 30) <= 27)

This became (1 <= 27) which is true so the final result is 1

This is why Mr C printed Just Right even when temp = 30

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

When we entered 30, Mr C evaluated ((22 <= 30) <= 27)

This became (1 <= 27) which is true so the final result is 1

This is why Mr C printed Just Right even when temp = 30

Complex Relational Expressions

<, <=, ==, >, >=, != are called *relational operators*

Expressions containing these operators generate 0 or 1

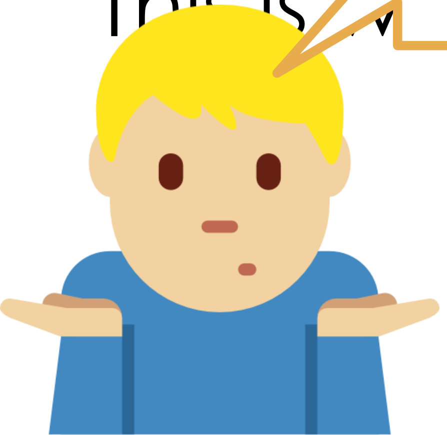
All have left to right associativity (just like +, -, *, /)

22 <= temp <= 27 became ((22 <= temp) <= 27)

When we entered 30, Mr C evaluated ((22 <= 30) <= 27)

This bec is true so the final result is 1

This is w Right even when temp = 30



So how do I get Mr C to do something when temperature is between 22 and 27?

Logical Operators

Logical Operators

Used to create powerful conditions and choices

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
    if((22 <= temp) && (temp <= 27)){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
    if((22 <= temp) && (temp <= 27)){ ... }
    if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
    if((temp >= 22) || (temp <= 27)){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

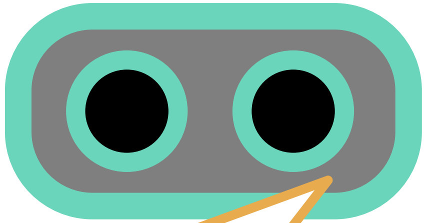
```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```



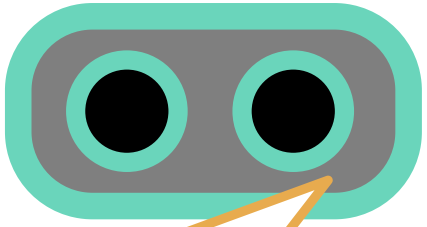
Bracket yourself to avoid errors and confusion

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```



Bracket yourself to avoid errors and confusion

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

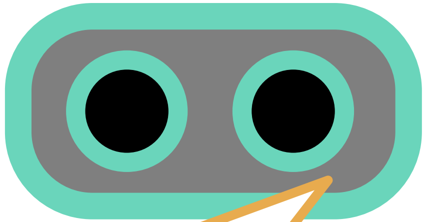
```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```



Bracket yourself to avoid errors and confusion

If you don't put brackets, Mr C will put brackets according to his table

Logical Operators

Used to create powerful conditions and choices

If we want temp >= 22 AND temp <= 27

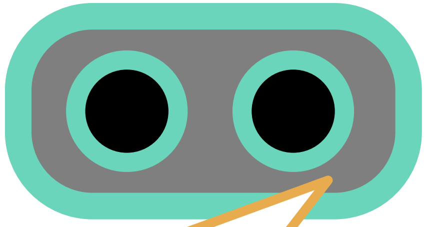
```
if((22 <= temp) && (temp <= 27)){ ... }
if((temp >= 22) && (temp <= 27)){ ... }
```

If we want temp >= 22 OR temp <= 27

```
if((temp >= 22) || (temp <= 27)){ ... }
```

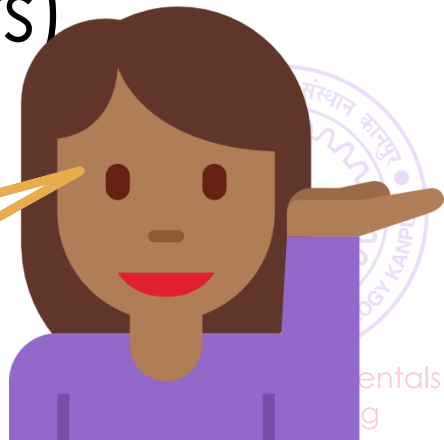
If we want NOT a % 2 == 0 (to select odd numbers)

```
if(!(a %2 == 0) ){ ... }
if(a %2 != 0){ ... }
```



Bracket yourself to avoid errors and confusion

Lets show them the new table



If you don’t put brackets, Mr C will put brackets according to his table