If you haven't yet, set-up the appliance by following the instructions at:
https://manual.cs50.net/Appliance

Getting around at the command line:

- ls
- mkdir
- cd
- rm (-r)
- man

Introducing C


```
int
main(void)
{
  // code goes here
}
```

Data types in C

- int
- short
- float
- double
- char
- long long
- “string”

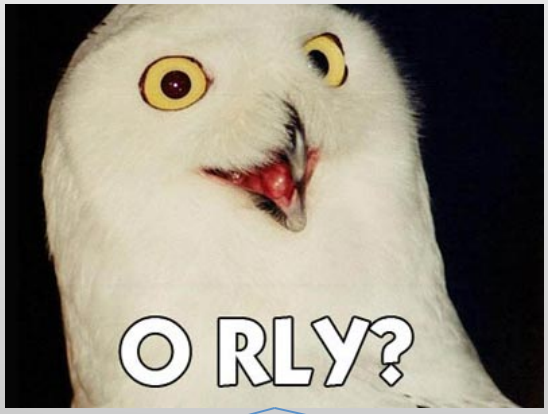
Remember!

It is impossible to store something like ".1" exactly in a float or a double

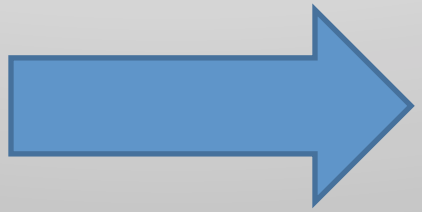
What's wrong with the following?

```
for(float f = 0; f != 1; f+=0.1)
{
   // code here
}
```

Conditions



Conditions



```
if (condition)
{
  // if “condition” is true
}
else
{
   // if “condition” is false
}
```

Conditions

```
while(condition)
{
   // code
}
```

The conditions for loops and if/else blocks must evaluate to a boolean value, i.e. true or false.

In C, there is no separate datatype for booleans. Instead, anything with a value of zero (e.g. “int i = 0”) evaluates to false, and any other value evaluates to true.

Conditions

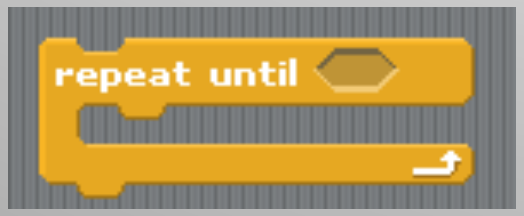
- <
- <=
- >
- >=
- ==
- !=
- !

You can use the operators to the left to compare values, e.g. “4 < 3” evaluates to false and “4 >= 3” evaluates to true. Note that “==“ is used to compare two numbers for equality (why?).

“!” means “not”, and negates the value of a boolean expression. So “!(4 < 3)” evaluates to true.

Loops


```
for(int i = 0; i < 10; i++)
{
   // code
}
```




```
while(condition)
{
   // code
}
```


```
while(1)
{
   // code
}
```

While Loops

```
while(condition)
{
   // code block which is repeated until
   // the condition is NOT true
}
```

For Loops

```
for(initialization; condition; update)
{
   // code block
}
```

For Loops

```
for(int i = 0; i < 10; i++)
{
   // code block which is repeated 10 times
}
```

while/for Comparison

```
int i = 0;
while(i < 10)
{
    printf("%d", i);
    i++;
}
```

```
for(int i = 0; i < 10; i++)
{
    printf("%d", i);
}
```

These two loops are equivalent.

Do-While Loops

```
do
{
   // code block
} while (condition);
```

Unlike in a while loop, we first check the condition *after* executing the code block!

Translating Scratch to C

Commenting

```
/**************************************************************************
 * math1.c
 *
 * Computer Science 50
 * David J. Malan
 *
 * Computes a total but does nothing with it.
 *
 * Demonstrates use of variables.
 **************************************************************************/
```

Always comment your code!