problem set 0

https://www.cs50.net/psets/0/
office hours
walkthrough
scribe notes
sectioning
when clicked
say O hai, world!
```c
int main()
{
    printf("O hai, world!\n");
}
```
statements

say O hai, world!
statements

printf("O hai, world!\n");
loops
while (true)
{
    printf("O hai!\n");
}
loops
for (int i = 0; i < 10; i++)
{
    printf("O hai!\n");
}
variables
variables

```c
int counter = 0;
while (true)
{
    printf("%d\n", counter);
    counter++;
}
```
Boolean expressions
Boolean expressions

\[(x < y) \land ((x < y) \land (y < z))\]
conditions
conditions

if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
arrays

string inventory[1];
inventory[0] = "Orange";
#include <stdio.h>

int main(void) {
    printf("O hai, world!\n");
}

how to write a program
how to compile a program

gcc hello.c
how to run a program

./a.out
how to compile a program

gcc -o hello hello.c
how to run a program

./hello
how to compile a program

make hello
functions
main
Standard Library

stdio.h

printf

...
CS50 Library

cs50.h

GetChar
GetDouble
GetFloat
GetInt
GetLongLong
GetString
printf
% c  % d  % f  %lld  % s  ...

escape sequences

\n \r \t \' \" \\ \0 ...
math
+
-
*
/
%

primitive types

char  double  float  int  long  long  long  ...

CS50 types

bool    string    ...
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>Parentheses (grouping)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>[</td>
<td>Brackets (array subscript)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>.</td>
<td>Member selection via object name</td>
<td>left-to-right</td>
</tr>
<tr>
<td>-&gt;</td>
<td>Member selection via pointer</td>
<td>left-to-right</td>
</tr>
<tr>
<td>++</td>
<td>Prefix increment/decrement</td>
<td>right-to-left</td>
</tr>
<tr>
<td>--</td>
<td>Prefix increment/decrement</td>
<td>right-to-left</td>
</tr>
<tr>
<td>+ -</td>
<td>Unary plus/minus</td>
<td>left-to-right</td>
</tr>
<tr>
<td>~</td>
<td>Logical negation/bitwise complement</td>
<td>left-to-right</td>
</tr>
<tr>
<td>(type)</td>
<td>Cast (change type)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>*</td>
<td>Dereference</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&amp;</td>
<td>Address</td>
<td>left-to-right</td>
</tr>
<tr>
<td>sizeof</td>
<td>Determine size in bytes</td>
<td>left-to-right</td>
</tr>
<tr>
<td>* / %</td>
<td>Multiplication/division/modulus</td>
<td>left-to-right</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition/subtraction</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&lt;&lt;=</td>
<td>Bitwise shift left, Bitwise shift right</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&lt; &lt;= &gt;=</td>
<td>Relational less than/less than or equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&gt; &gt; =</td>
<td>Relational greater than/greater than or equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>== !=</td>
<td>Relational is equal to/not equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&amp; &amp;</td>
<td>Bitwise AND</td>
<td>left-to-right</td>
</tr>
<tr>
<td>^</td>
<td>Bitwise exclusive OR</td>
<td>left-to-right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bitwise inclusive OR</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Logical AND</td>
<td>left-to-right</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?:</td>
<td>Ternary conditional</td>
<td>right-to-left</td>
</tr>
<tr>
<td>=</td>
<td>Assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>+= -=</td>
<td>Addition/subtraction assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>*= /=</td>
<td>Multiplication/division assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>%=</td>
<td>Modulus/bitwise AND assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>^=</td>
<td>Bitwise exclusive/exclusive OR assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>&lt;&lt;= &gt;&gt;=</td>
<td>Bitwise shift left/right assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>,</td>
<td>Comma (separate expressions)</td>
<td>left-to-right</td>
</tr>
</tbody>
</table>
how to compile a program

gcc -o hello hello.c -lcs50
how to compile a program

make hello
to be continued...