Day 13 Advanced Program Control

Ending Loops Early

break- placed only in body of for loop, while loop, do...while loop, switch, when encountered execution immediately exits the loop

continue- " when encountered goes back to beginning of loop to begin next iteration

goto- unconditional jump instruction. when encountered will jump the program to a location you identify with something like "location1:". Should never use. Target must be in same function

Infinite Loops

will run forever

have one of the loop functions evaluate an always true condition such as while (1) or for (; ;)

good for when many conditions need to be evaluated

avoid if other alternatives

```
void delay ( void) {
    long x;
    for ( x=0 ; x < 15000 ; x++)
        ;
}
```

that is good if you want a momentary pause in program and compiler does not contain something like sleep()

Switch Statement

execute different statements based on expression that can have more than 2 values

if template matches expression, then that statement and every statement below executed (**why typically use break after every statement**)

```
switch (expression)
{
    case template1:
    {
    statement1;
    break;
```

```
    }
    case template2:
    {
    statement2;
    break;
    }
    default:
    {
    defaultStatement;
    }
}
```

also can evaluate multiple templates to the same statement:

```
...
case 1:
case 2:
case 3:
{
    statement123;
    break;
}
```

Exiting the Program

can terminate a program at any time by using **exit()** can also specify one or more functions to be executed at termination

takes single int argument to indicate success or failure

`exit(status);`

0 means terminated normally, 1 means terminated with some error

OS System Commands

`system(command);`