More about Linux Processes

Name: _____

Create a new program named "delay" using this source code:

```
#include <iostream>
using namespace std;
int main ()
{
  int cnt;
  for (cnt=0; cnt<30; cnt++)
     {
      cout << cnt << endl;
      sleep (1);
     }
  return 1;
}
```

This new delay program will create some simple, yet annoying, output that we can use to experiment with process control commands.

Run delay, but before it can finish (it takes 30 seconds to finish running), type ^C (hold down Ctrl and C at the same time).
Question 1: What does ^C do?

Run delay again, but this time give it a ^Z.
Question 2: What does ^Z do?

Type these commands

```
jobs
fg %1
```

Question 3: What did the fg command do?

After delay is finally finished, let's start it again and try something else. Do each of these commands within just a few seconds of each other (don't let delay finish before you type ls):

```
delay
^Z
jobs
bg %1
ls
```

Note that both ls and delay are running at the same time. bg makes a job run in the background.

Finally, try this

```
delay &
^C
```

Note that delay did not die. It is running in the background and so the ^C went to the foreground process, which is the bash shell.

Question 4: Suppose I typed `delay &`, yet my code had a bug and so delay never dies. Give me the commands to track down and kill this runaway process.