NAME:_____

COLLABORATOR(S):_____

1. What is a signal and what makes them *asynchronous*?
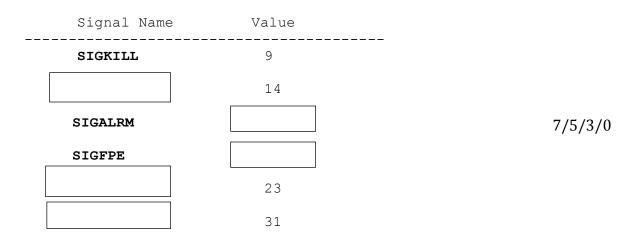
```



```
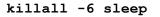
5/3/1/0

2. Match the terminal signal to the keyboard shortcut/shell command

Ctrl-c ____

Ctrl-Z ____

fg/bg   ____

Ctrl-\ ____

(a) **SIGSTOP**

(b) **SIGCONT**

(c) **SIGINT**

(d) **SIGTERM**

(e) **SIGQUIT**

8/5/3/0

3. Open up section 7 of the man page for signal (**man 7 signal**) and find the signal value or the signal name for the following signals:

```
          Signal Name          Value
         ------------------------------------------
           SIGKILL               9

           [        ]            14

           SIGALRM            [        ]

           SIGFPE             [        ]

           [        ]            23

           [        ]            31
```

7/5/3/0

4. Describe what the following command does with respect to signaling and the expected result.

killall -6 sleep

```



```

5/3/1/0

_/25

5. On a lab machine in MI302, run the following program

~aviv/hw9-bin/ic221-signaler

From another terminal on the same machine, using **killall**, send the
program a SIGUSR1 and SIGUSR2 signal, describe the result.

SIGUSR1

3/1/0

SIGUSR2

3/1/0

6. For the following program

a) What is the output of running the
program after typing *one* Ctrl-c? Why?

5/3/0

b) What is the output of running the
program after typing *four* Ctrl-C? Why?

5/3/0

c) What is the output of running the
program after typing *three* Ctrl-C's and
one Ctrl-Z? Why?

5/3/0

```c
int count = 0;
void hand1(int signum){
    printf("You Shot Me!\n_");
    count++;
    if(count > 3){
        printf("I'm dead.\n");
        _exit(1);
    }
}

int main(){
    //set up handler
    //for SIGINT and SIGSTOP
    signal(SIGINT,handler);
    signal(SIGSTOP,handler);

    //loop forever
    while(1);
}
```

7. What two signals can never be handled, ignored, or blocked and
why not?

4/3/1/0

8. What is the difference between the two code snippets in terms of CPU resources and signal handling:

5/3/1/0

while(1);   and   **while(1) pause();**

---

9. For the following code snippet, in what interval and how many alarm messages print:

8/5/3/0

```c
int len = 1;
void handler(int signum){
    printf("Alarm\n");
    len = (len * 2) % 4;
    alarm(len);
}
int main(){
    signal(SIGALRM,handler);
    alarm(len);
    while(1) pause();
}
```

---

10. For the following program, how long does it take SIGALRM to be delivered and what happens after SIGALRM is delivered? Explain.

7/5/3/0

```c
int main(){
    alarm(1);
    alarm(2);
    alarm(3);
    alarm(2);

    pause(1);
}
```

---

11. Convert the use of **signal()** system call to a **sigaction()**:

signal(SIGALRM, handler);

7/5/3/0

```c
struct sigaction;
```

_/27

12. What does the error code EINTR stand for and when does such an error occur?

5/3/1/0

13. What is the necessary **sigaction** flag to ensure that a system call will restart if it is interrupted?

5/3/1/0

14. According to section 7 of the man page for signal (man 7 signal), which system calls will be atomically restarted after the signal handler returns if the right flag is used?

8/5/3/0

15. Draw a picture of your favorite signal from section 7 of the man page for signal. Be sure to identify the signal.

5/3/1/0

___/23