OS Quiz 3

Q1) How many processes?

```
1  int main()
2  {
3          int i = 0;
4          for(i = 1; i < 10; i++){
5                  if((i%2) == 0)
6                  {fork();}
7          }
8  }
```

The program starts with the parent program running into the form look.

At i = 1, the parent enters the loop. i%2 = 1 so no fork.

At i = 2, the parent enters the loop, i%2 = 0 so fork()

Parent, and 1 child come up

At i = 3, two (2) processes enter loop, i%3 = 1 so no fork.

2 processes come up

At i = 4, two (2) processes enter loop, i%4 = 0 so fork()

4 processes come up

The pattern shows at i %2 (i.e. whenever i is even, there is a fork() on the number of processes. The number of times this

happens is at i = {2,4,6,8}. Note that i=10 never happens inside the loop.

So at i = 0, 1 process,

At i = 2, 2 processes,

At i = 4, 4 processes,

At i = 6, 8 processes,

At I = 8, 16 processes

A total of 16 processes is created including the parent.

Q2) What is the output of this program.

```
1  #define SIZE 5
2  int num[SIZE] = {2,4,6,8,10};
3  int main()
4  {
5        pid_t fpid;     int i; fpid = fork();
6        if(fpid == 0){
7              for( i = 0; i < SIZE; i++){
8                    if (fpid == 0)
9                    { num[i] = num[i]+i; }
10                   else { num[i] *=2;}
11             }
12       }
13       for(i = 0; i < SIZE;i++){
14             if(fpid == 0)
15             {printf("say:%d",num[i]);}
16             else {printf("speak:%d",num[i]);}
17       }
18 }
```

The program does a fork at line 5 (fpid=fork();)

The parent has an array with {2,4,6,8,10}

The child has an array with {2,4,6,8,10}

After the fork (line 6 and after), both parent and child have same data. However, only the child can etner the if(fid==0) to execute lines 7,8,9,and 10. The question looks tricky because it has line 8

ask if fid==0. The child enters this part so the child will simply add I to each number. i.e. {2+0,4+1,6+2,8+3,10+4} = {2,5,8,11,14}

At line 13, the parent still has {2,4,6,8,10}, but the child has {2,5,8,11,14}. The parent prints "Speak %d", and the child uses "Say %d"

The printout will include

Say 2

Say 5

Say 8

Say 11

Say 14

Speak 2

Speak 4

Speak 6

Speak 8

Speak 10