CS31 Written Homework 8: Processes Name:

Question 1

For the code snippet shown below (assume that all the calls to `fork()` succeed), answer the following questions:

1. Draw a process-hierarchy diagram that results from the code's execution. Your diagram should be similar to Figure 2 in Chapter 9, where you draw a node for every process and arrows from parent to child processes. Also, in your picture:
   - Label each node with a letter to indicate the order in which it was spawned. In cases where the order is not determined, choose a possible order.
   - Next to each node write the output value(s) that process prints via `printf()`.

2. After this code's execution, how many zombie process are there? Explain your answer in a sentence or two.

```
int i, pid;

printf("-1");
pid = 0;
for(i = 0; i < 2; i++){
   pid = fork();
   printf("%d ", i);
}

if(pid != 0){
  wait(NULL);
} else {
  exit(0);
}
```

Question 2

Consider the code snippet shown below (and assume all calls to fork succeed).

(1) To the right of the code, draw the execution timeline corresponding to the code's execution showing a possible ordering of `fork()` and `wait()` calls from the proceses involved. Use Figure 7 in Chapter 9 of the textbook as an example.

```
int pid1, pid2;

printf("1");
pid1 = fork();
if (pid1 == 0){
   pid2 = fork();
   printf("2 ");
   if (pid2 == 0){
        printf("3 ");
        exit(0);
   } else {
        printf("4 ");
        wait(NULL);
        printf("5 ");
        exit(0);
   }
} else {
   printf("6 ");
   wait(NULL);
   printf("7 ");
}
```

(2) Which of the following outputs below are possible from executing the above code? For any that are not, describe in 1 sentence why not.

(*) 1 2 3 4 5 6 7

(*) 1 2 2 3 4 5 6 7

(*) 1 6 2 3 2 4 7 5

(*) 1 6 2 2 4 3 5 7