Practice Sheet – fork & exec ***(Try to solve in 15-20 mns before looking at key on next page)***

In the code below, assume that *(i)* all `fork` and `execvp` statements execute successfully, *(ii)* the program arguments of `execvp` do not spawn more processes or print out more characters, and *(iii)* all `pid` variables are initialized to `0`.

a. What is the total number of processes that will be created by the execution of this code?
b. How many of each character 'A' to 'J' will be printed out?

You must justify your answers with a clear and readable table or diagram.

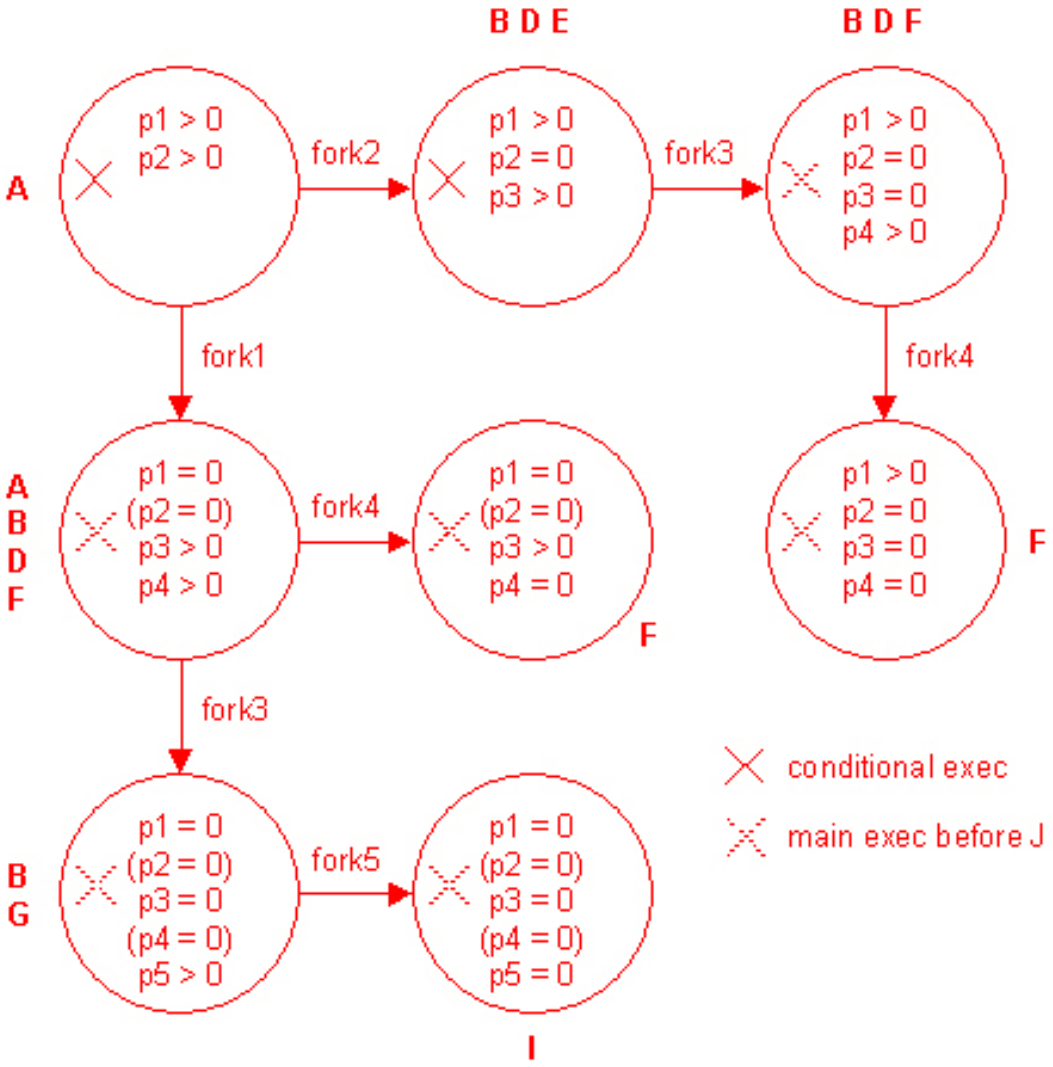
```
void main()
{
     ...
     pid1 = fork();
     printf("A");
     if (pid1) pid2 = fork();
     if (!pid2) {
         pid3 = fork();
         printf("B");
     }
     else {
         execvp(...);
         printf("C");
     }
     if (pid1 != pid3) {
         printf("D");
         if (pid1 && pid3) {
             printf("E");
             execvp(...);
         }
         pid4 = fork();
         printf("F");
     }
     else {
         printf("G");
         pid5 = fork();
     }
     if (pid2) {
         pid6 = fork();
         pid7 = fork();
         if (pid6) pid8 = fork();
         printf("H");
     }
     if (!pid1 && !pid2 && !pid3 && !pid4 && !pid5) printf("I");
     execvp(...);
     printf("J");
}
```

Key:

a. **8 processes total**

b. **A = 2** **F = 4**
B = 4 **G = 1**
C = 0 **H = 0**
D = 3 **I = 1**
E = 1 **J = 0**

DIAGRAM:



In this diagram, "p1" stands for pid1, "fork1" stands for pid1 = fork(), etc.

(ALL KINDS AND SHAPES OF DIAGRAMS AND TABLES ARE ACCEPTED, AS LONG AS THE FOLLOWING ITEMS ARE CLEARLY VISIBLE FOR EACH PROCESS: fork transitions, pid variables, printed characters; EXEC MARKINGS ARE OPTIONAL)