8.9)

AB: N
AC: Y
AD: Y
BC: Y
BD: Y
CD: Y

8.11)

The for loop gives us two forks, resulting in **four** runs of the function.

8.13)

First we fork. Whichever fork gets called first gets to access x. If the printf in the conditional gets called, its outcome would be:

X=4

X=3

8.14)

We run Fork once in the conditional, so the second fork does not replicate on the branch, giving us not four, but three forks. However, unlike the next problem, we are using exit(0) instead of return, terminating the whole process, not just exiting the function. The final hello is never called.

hello

hello

8.15)

We run Fork once in the conditional, so the second fork does not replicate on the branch, giving us not four, but three forks. The output then follows:

hello

hello

hello

8.16)

Nothing is printed due to our use of an exit().