CSC209H Worksheet: Fork

Consider the program below that runs without errors. In the table to the right, indicate how many times the name of each fruit is printed.

```
int main(void) {
    printf("Mangoes\n");
    int r = fork();
    printf("Apples\n");
    if (r == 0) {
        printf("Oranges\n");
        int k = fork();
        if (k >= 0) {
            printf("Bananas\n");
        }
    } else if (r > 0) {
        printf("Peaches\n");
        for (int i = 0; i < 3; i++) {
            if ((r = fork()) == 0) {
                printf("Pears\n");
                exit(0);
                printf("Nectarines\n");
            } else if (r > 0) {
                printf("Plums\n");
            }
        }
    }

    return 0;
}
```

Fruit Name	**Times Printed**
`Mangoes`	1
`Apples`	2
`Oranges`	1
`Bananas`	2
`Peaches`	1
`Pears`	3
`Nectarines`	0
`Plums`	3

Several orderings of the fruit names are possible valid output. Some of these orderings even have the unix prompt displaying before the final fruit name (or names). Explain why this happens. SOLUTION: The parent process finishes and the child is still running.

Not all of the fruit names could appear after the prompt in a valid output. For example the word `Mangoes` will never appear after the prompt. List all the fruit names that could occur after the prompt.

SOLUTION: Apples, Oranges, Bananas, Pears