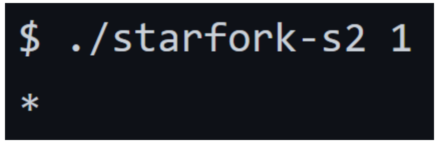
Starfork

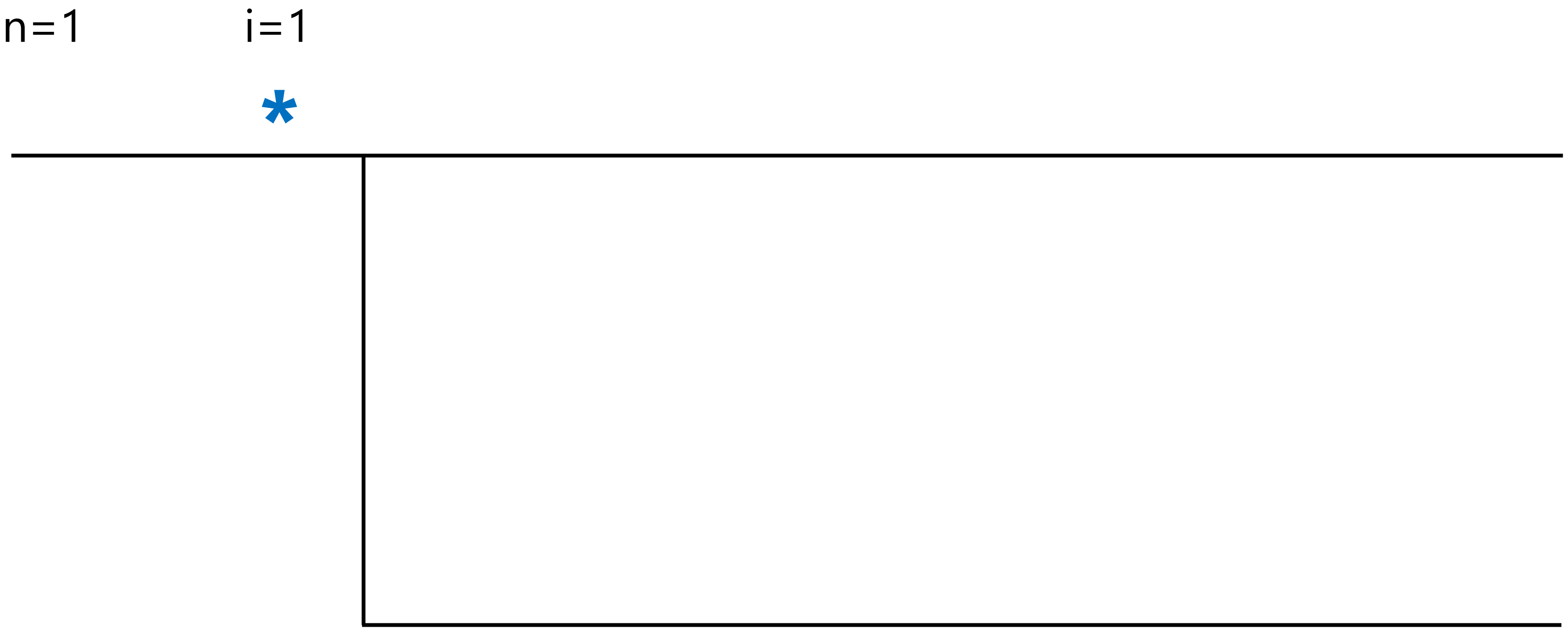
Stanley Lin and Alex Xu

COMS 3157 SP23
Recitation 7
3/24/2023

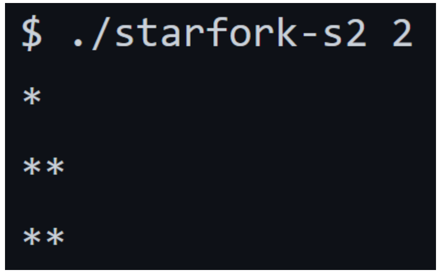
./starfork-s2 1



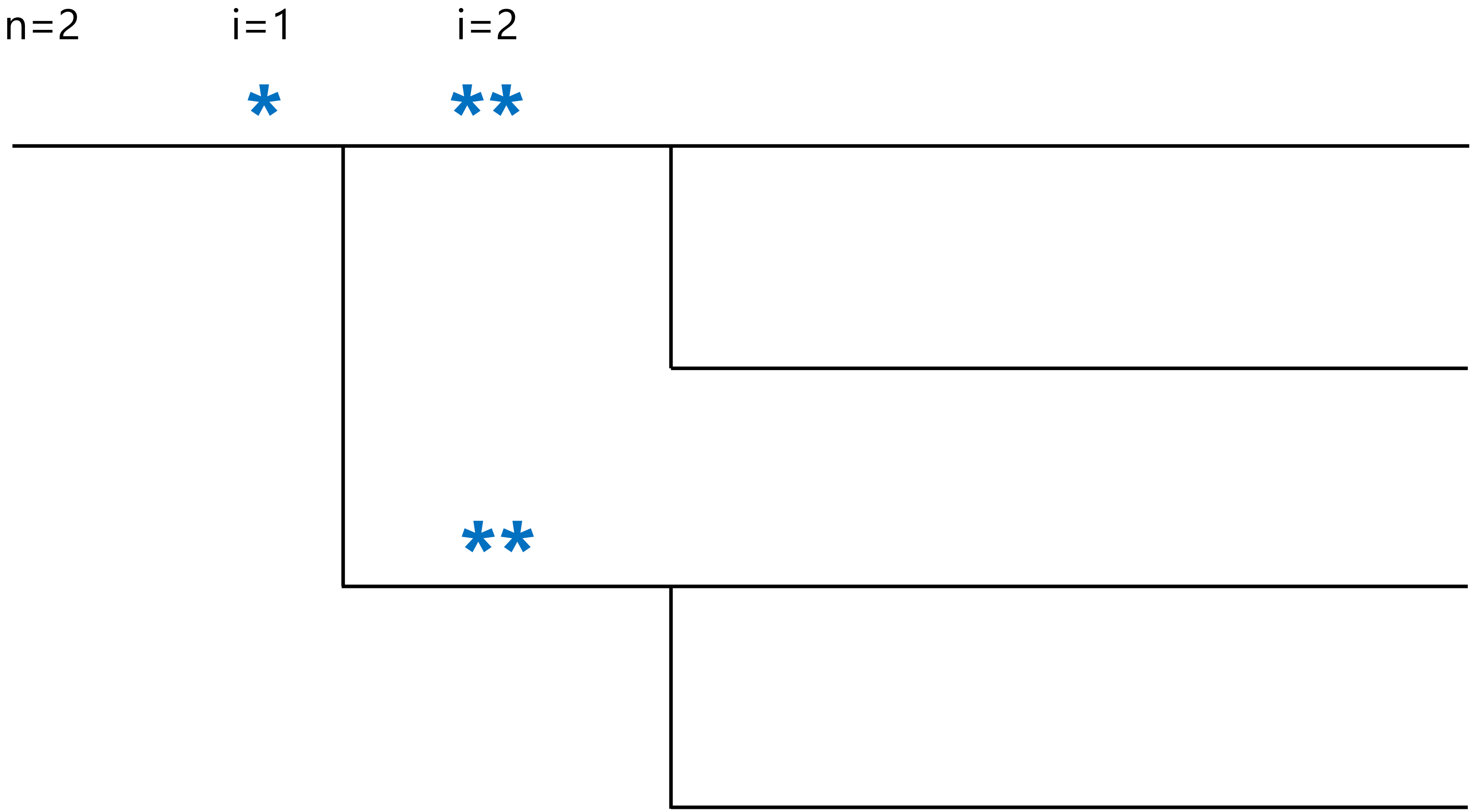
```
$ ./starfork-s2 1
*
```



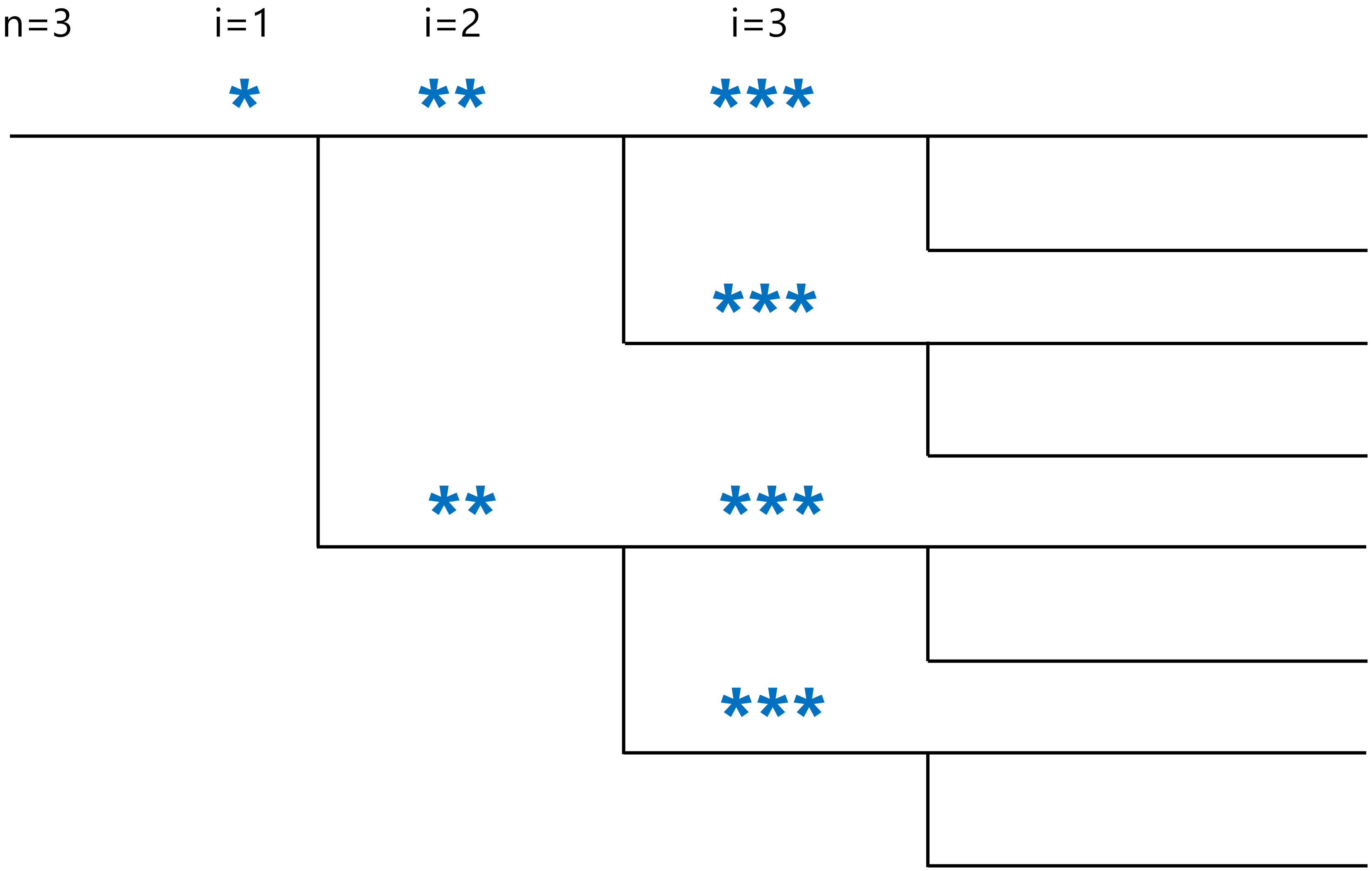
./starfork-s2 2



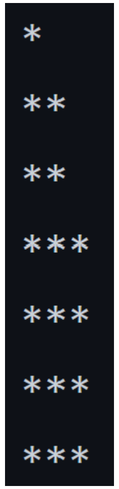
```
$ ./starfork-s2 2
*
**
**
```



./starfork-s2 3

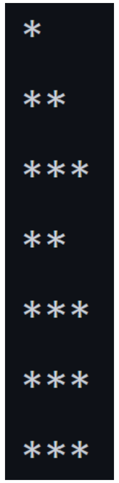


✓



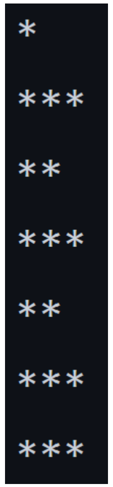
```
*
**
**
***
***
***
***
```

✓



```
*
**
***
**
***
***
***
```

✗

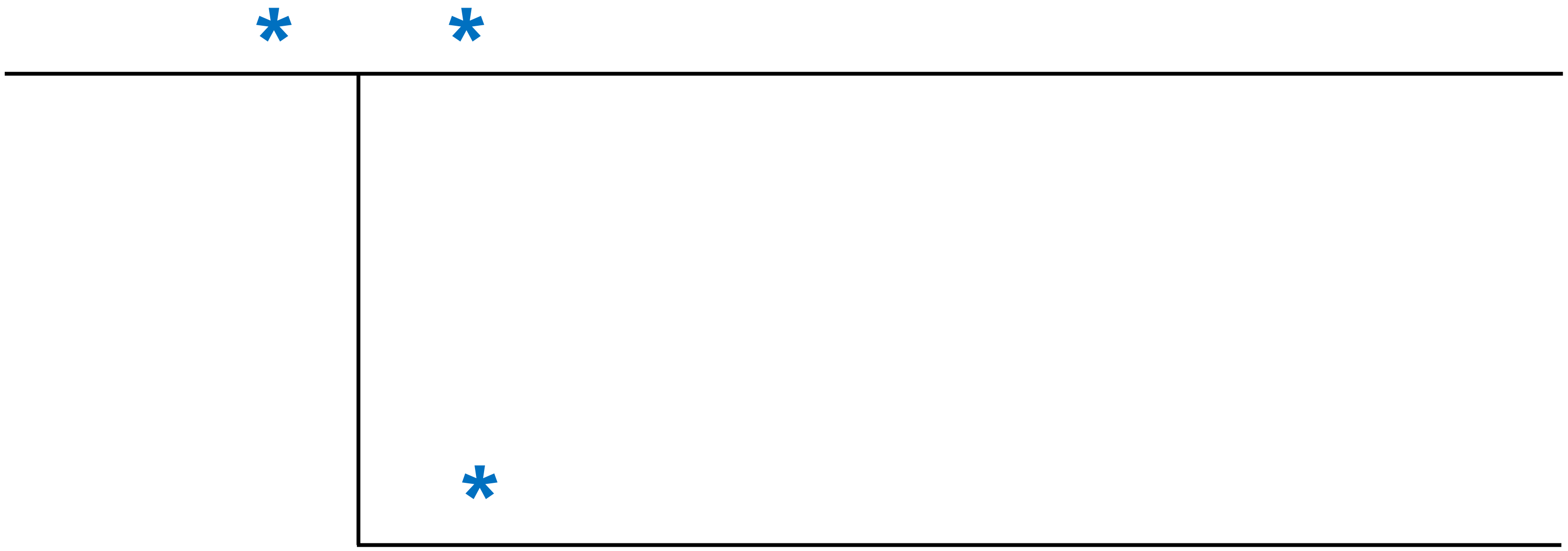


```
*
***
**
***
**
***
***
```

Unpredictable, but not totally random!

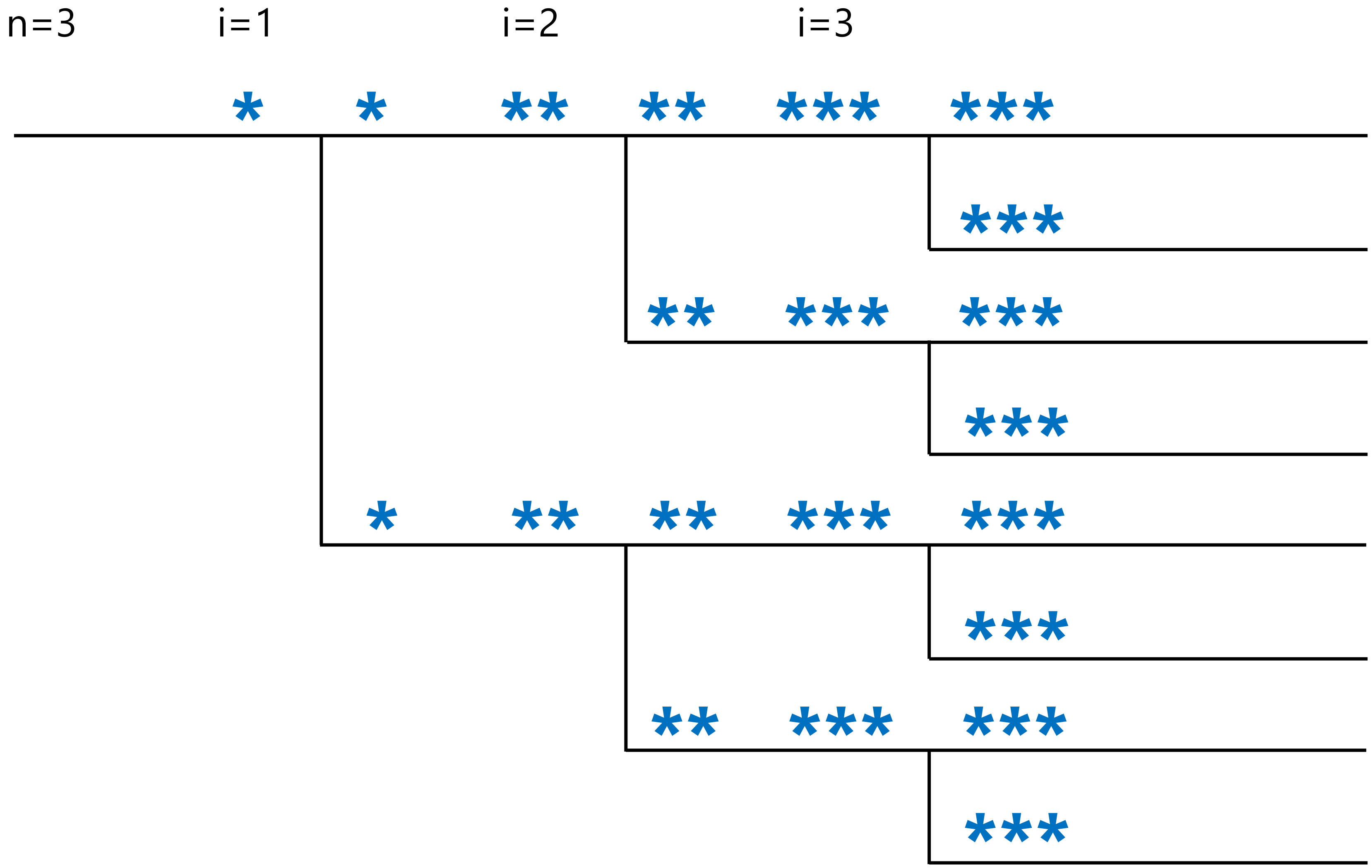
./starfork-s3 1

n=1 i=1

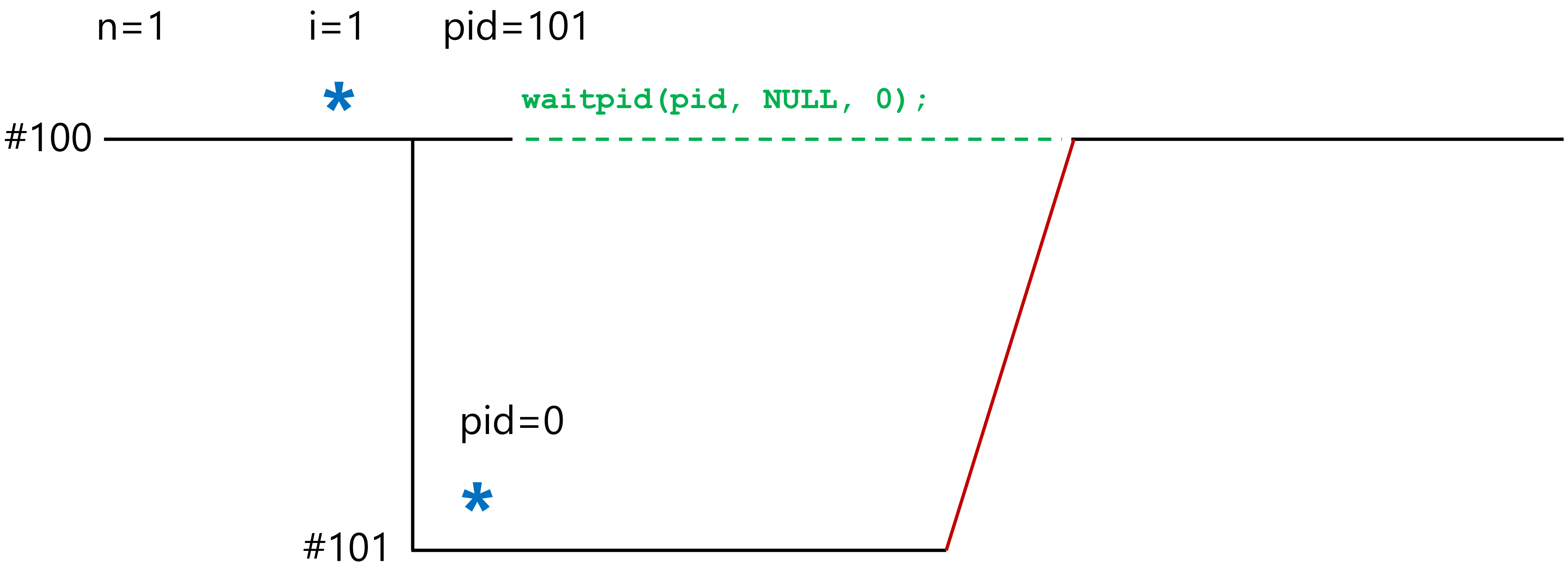


./starfork-s3 3

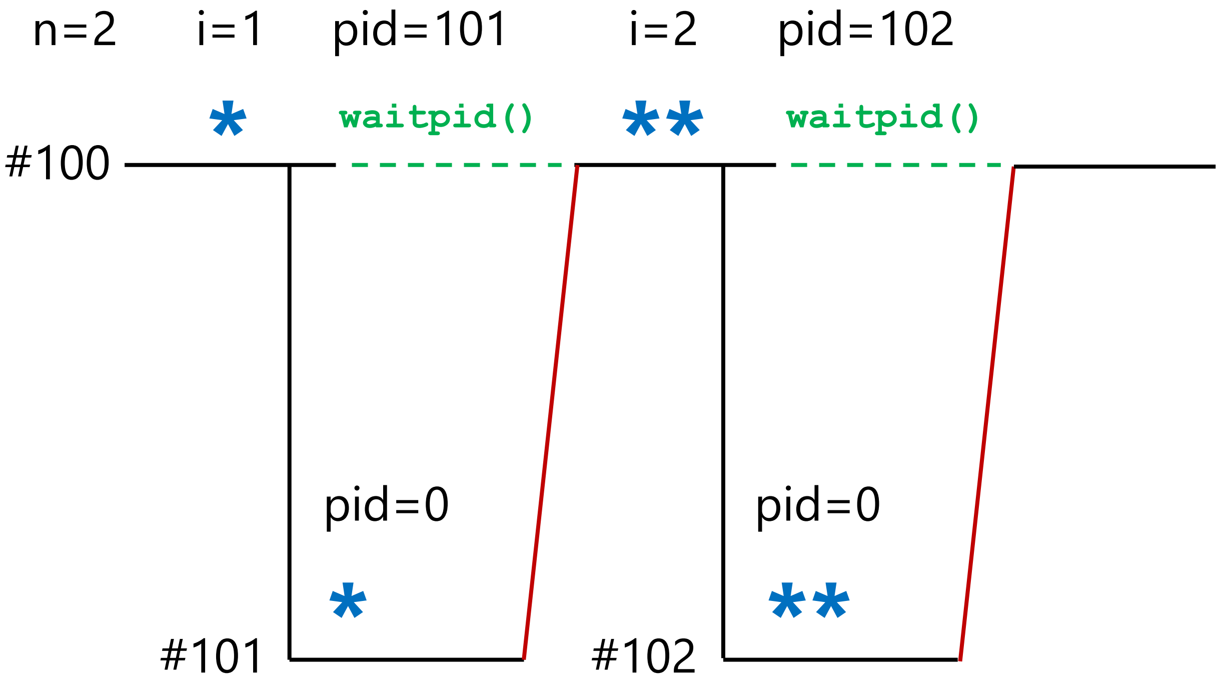
Unpredictable when n > 1



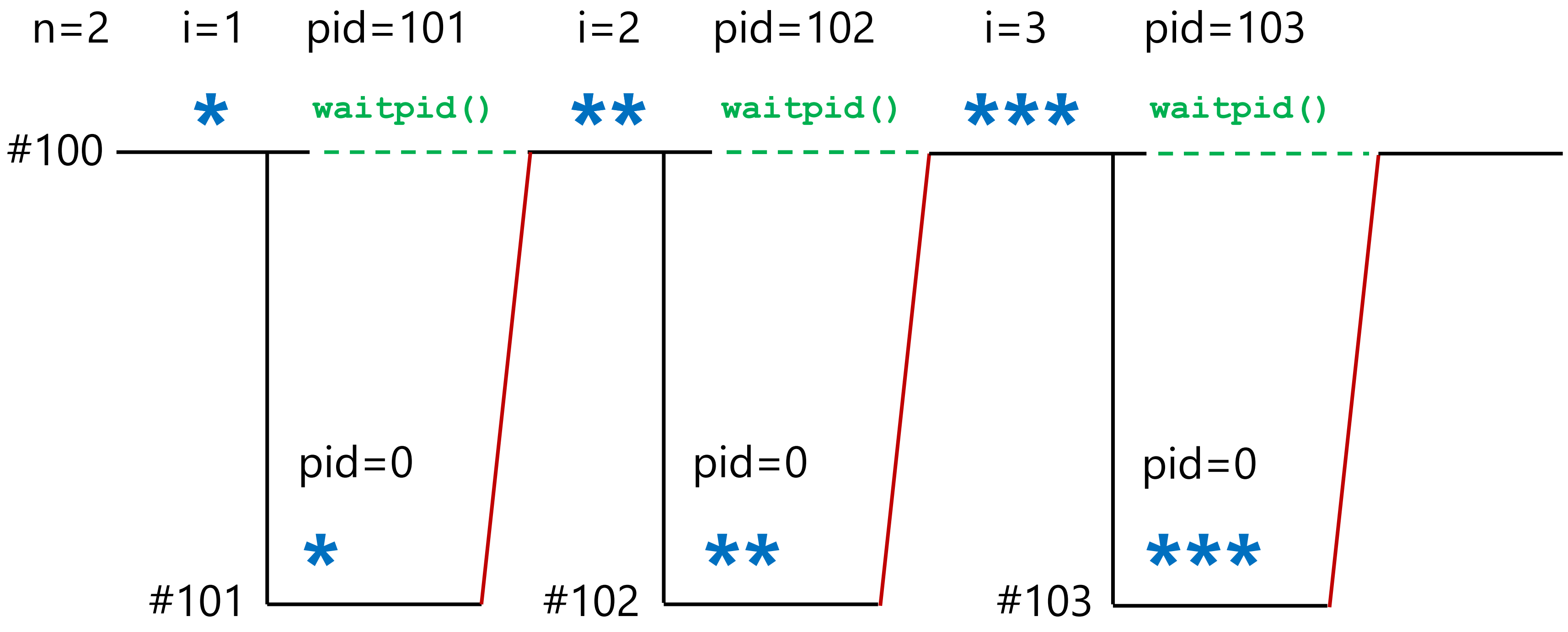
./starfork-s4 1



./starfork-s4 2



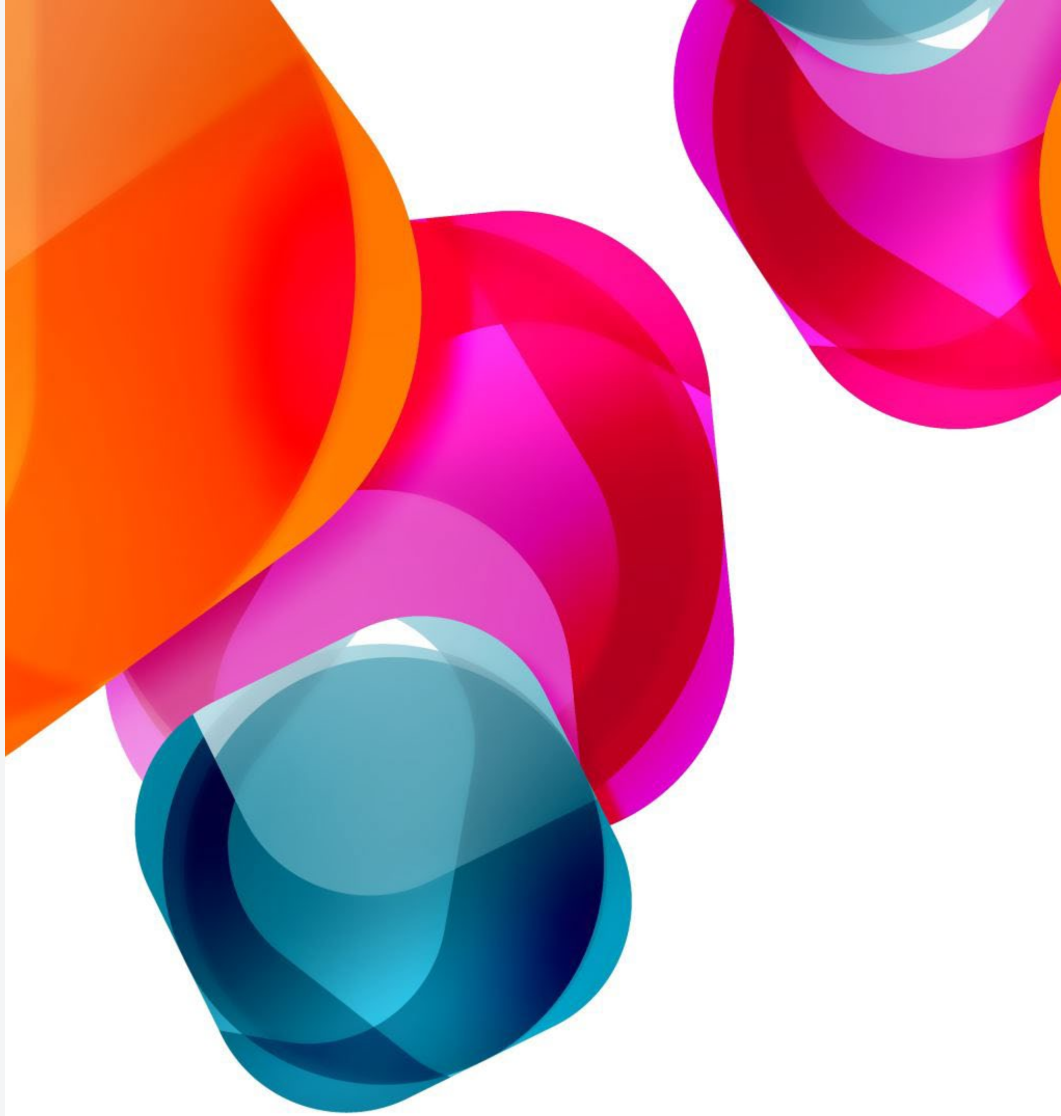
./starfork-s4 3



Recitation 7
fork, waitpid & exec

March 24, 2023

Alex XU



Part-5

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

How many lines are in the stdout outputs when running with arguments 1, 2 or 3?

Part-5

```
$ ./starfork-s5 1 2>/dev/null
```

n=1 i=1

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

stdout

Part-5

```
$ ./starfork-s5 1 2>/dev/null
```

n=1 i=1 *

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

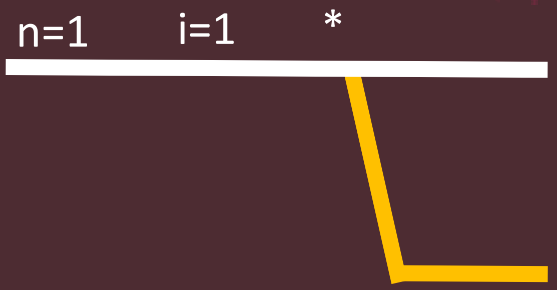
stdout

```
*
```

Part-5

```
$ ./starfork-s5 1 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

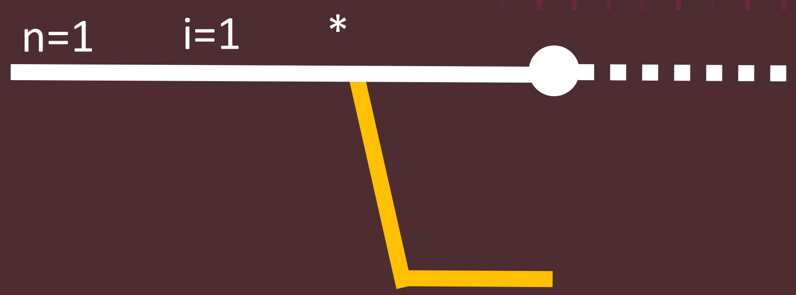


stdout

```
*
```

Part-5

$ ./starfork-s5 1 2>/dev/null



```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

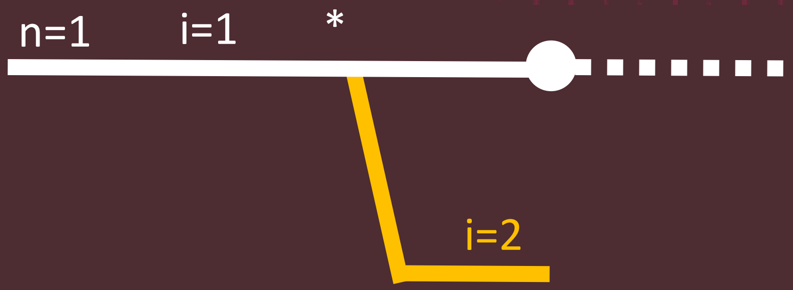
stdout

```
*
```

Part-5

```
$ ./starfork-s5 1 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



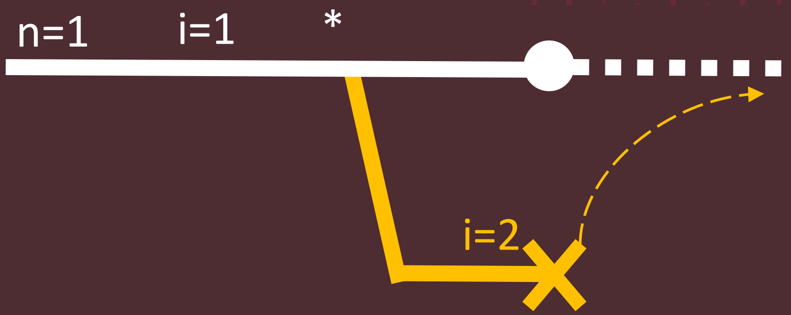
stdout

```
*
```

Part-5

$ ./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



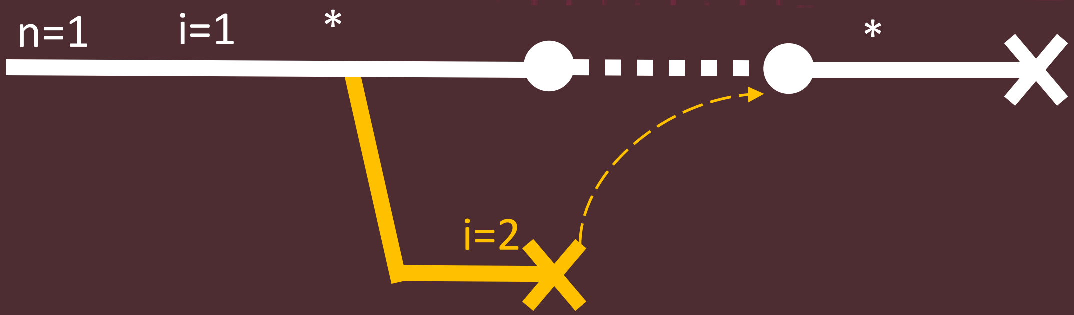
stdout

```
*
```

Part-5

```
$ ./starfork-s5 1 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



stdout

```
*
*
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

n=2 i=1

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

stdout

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

n=2 i=1 *

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

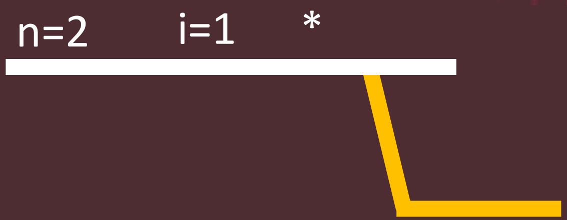
stdout

```
*
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



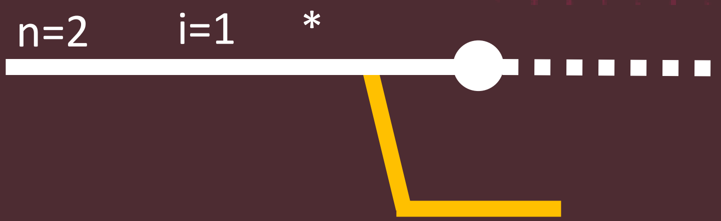
stdout

```
*
```

Part-5

$ ./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

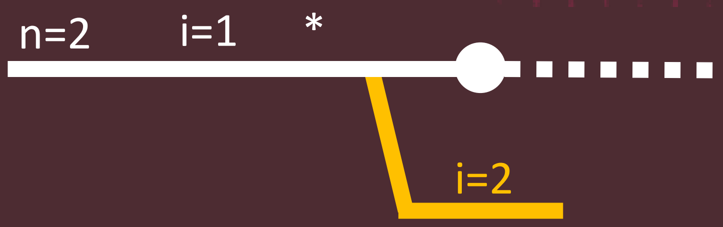


stdout

```
*
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```



```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

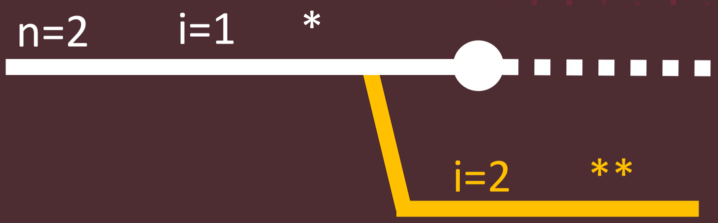
stdout

```
*
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



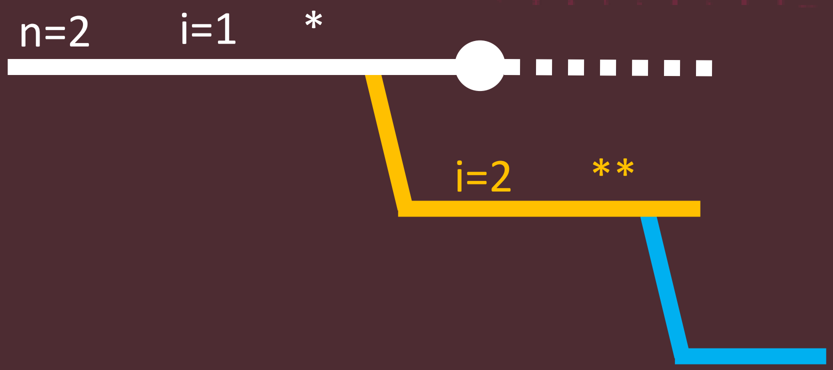
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



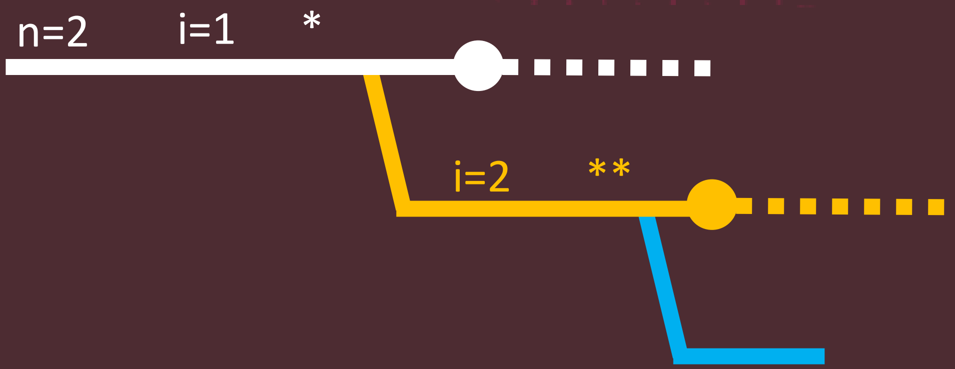
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



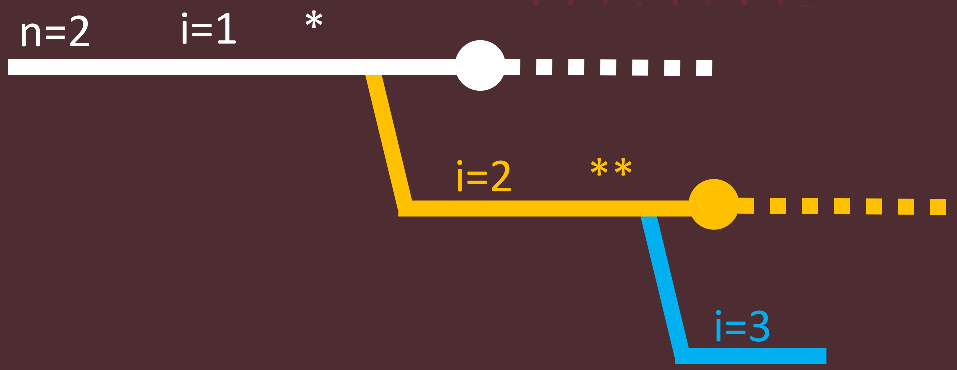
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



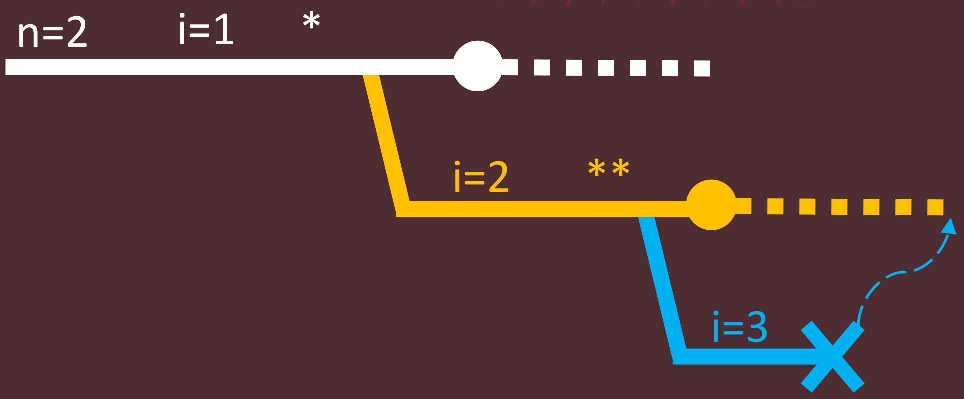
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



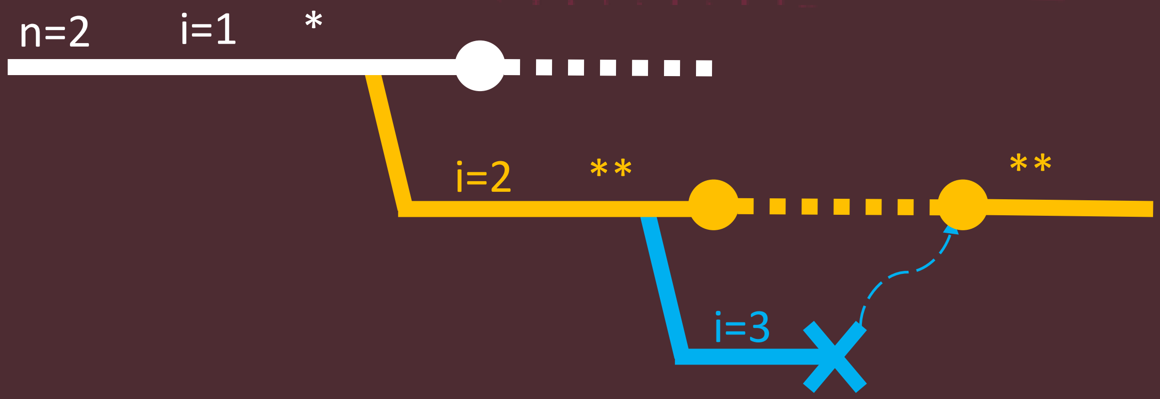
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



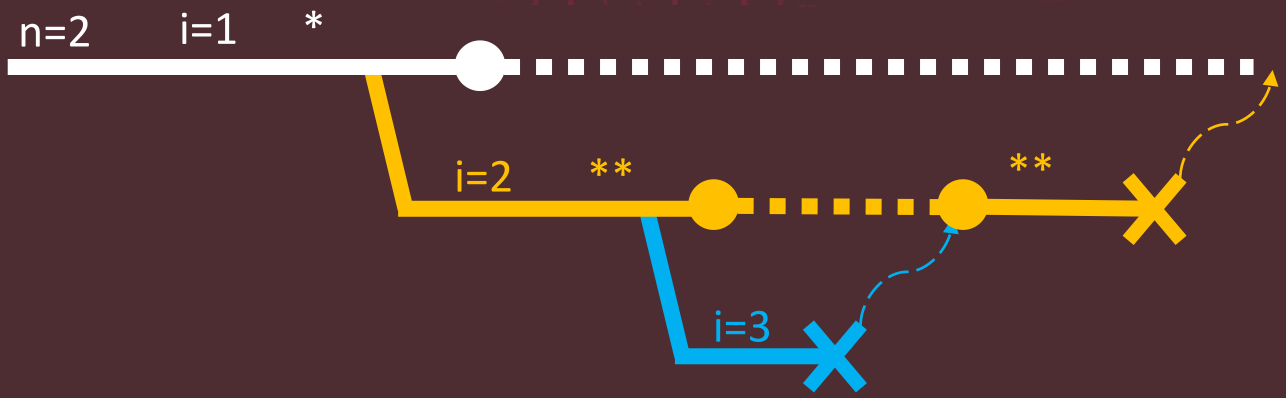
stdout

```
*
**
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



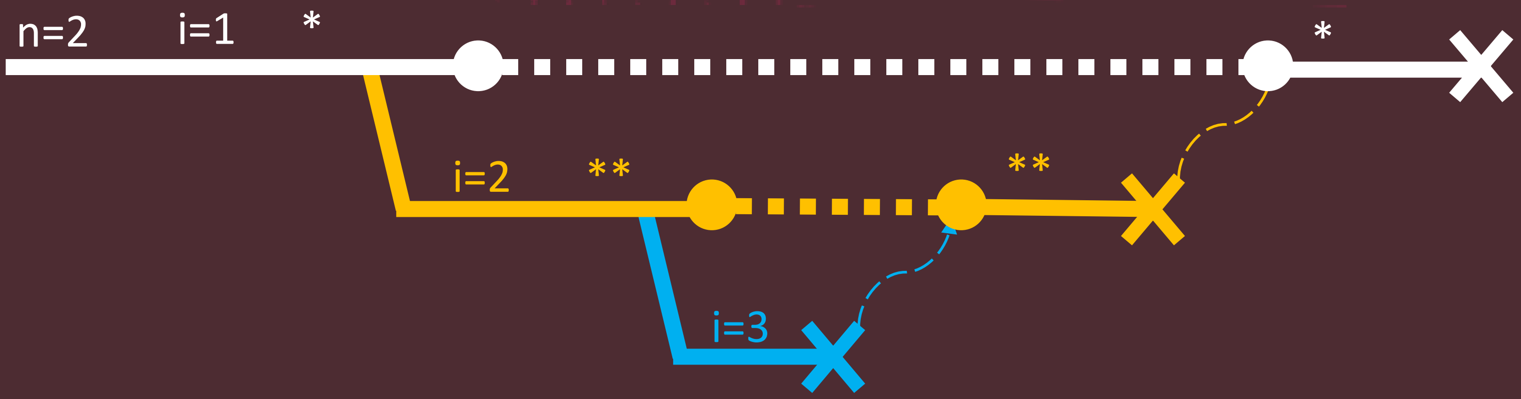
stdout

```
*
**
**
```

Part-5

```
$ ./starfork-s5 2 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



stdout

```
*
**
**
*
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

n=3 i=1

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

stdout

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

n=3 i=1 *

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

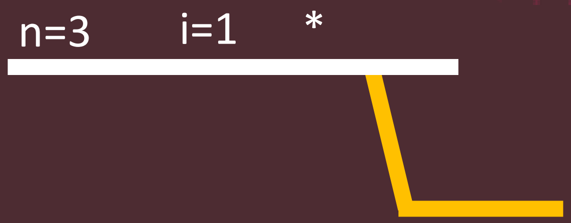
stdout

```
*
```

Part-5

$ ./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



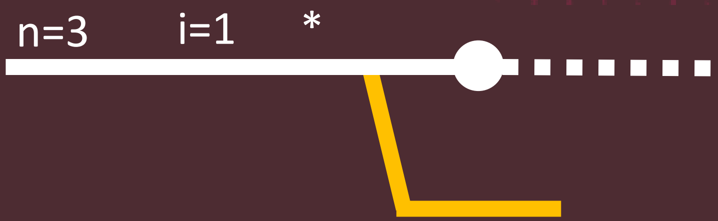
stdout

```
*
```

Part-5

$ ./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



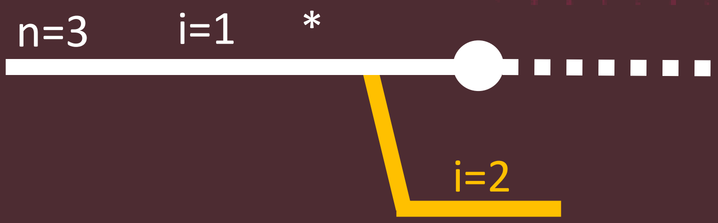
stdout

```
*
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



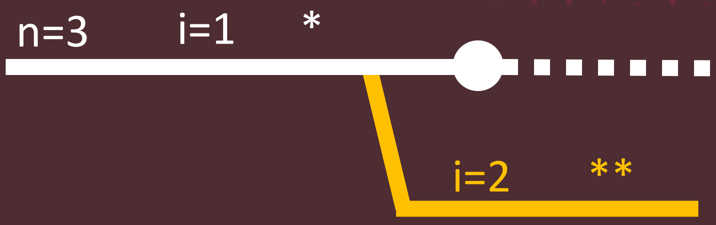
stdout

```
*
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



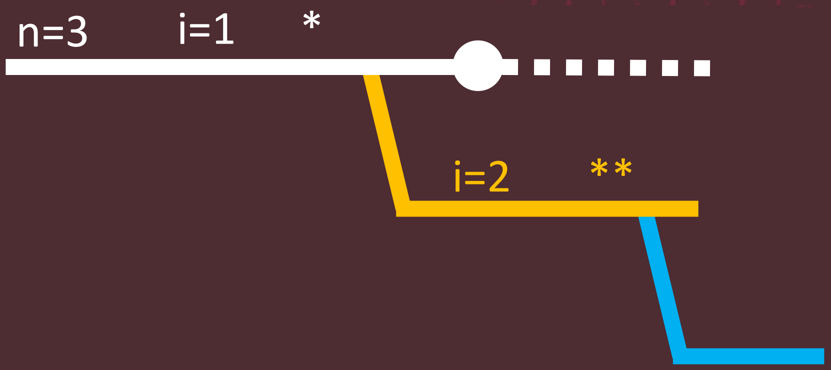
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



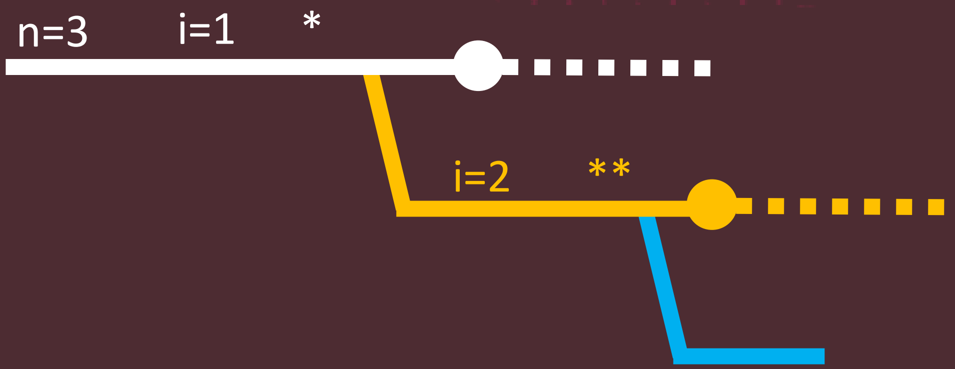
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



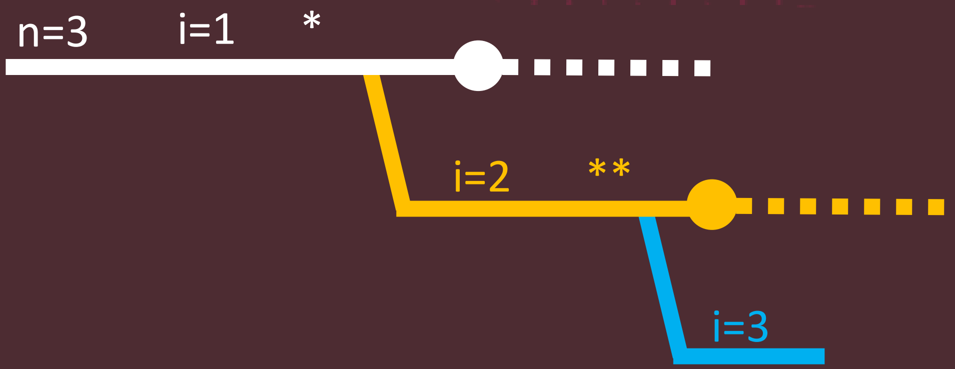
stdout

```
*
**
```

Part-5

$ ./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



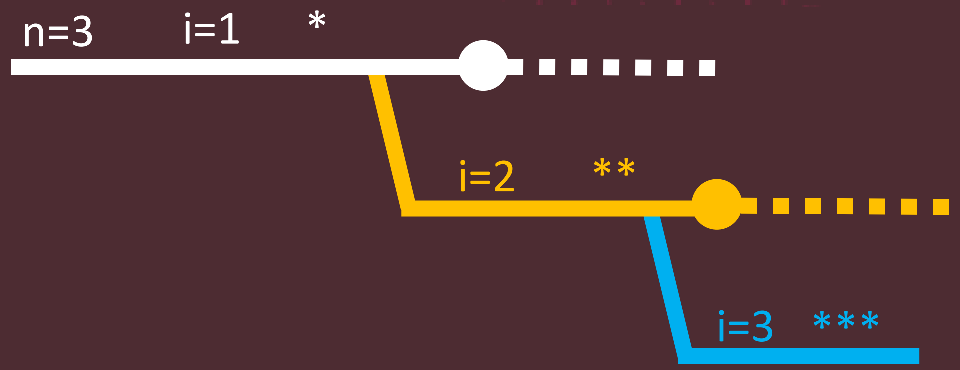
stdout

```
*
**
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



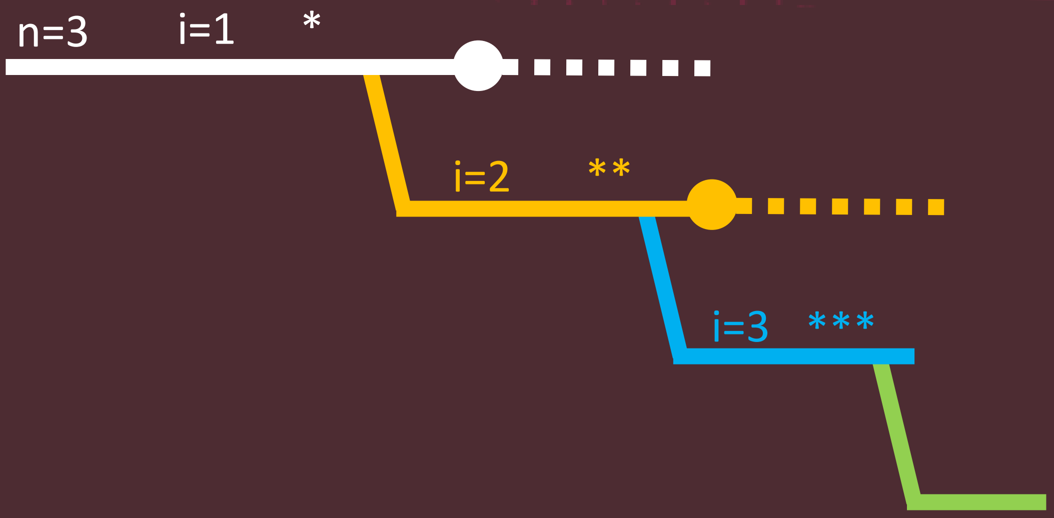
stdout

```
*
**
***
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



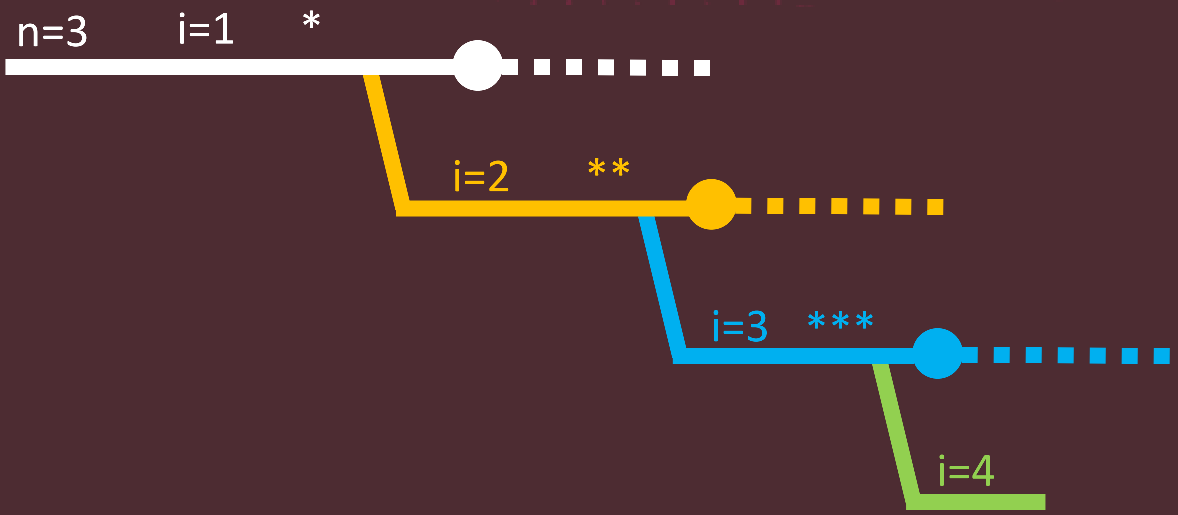
stdout

```
*
**
***
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



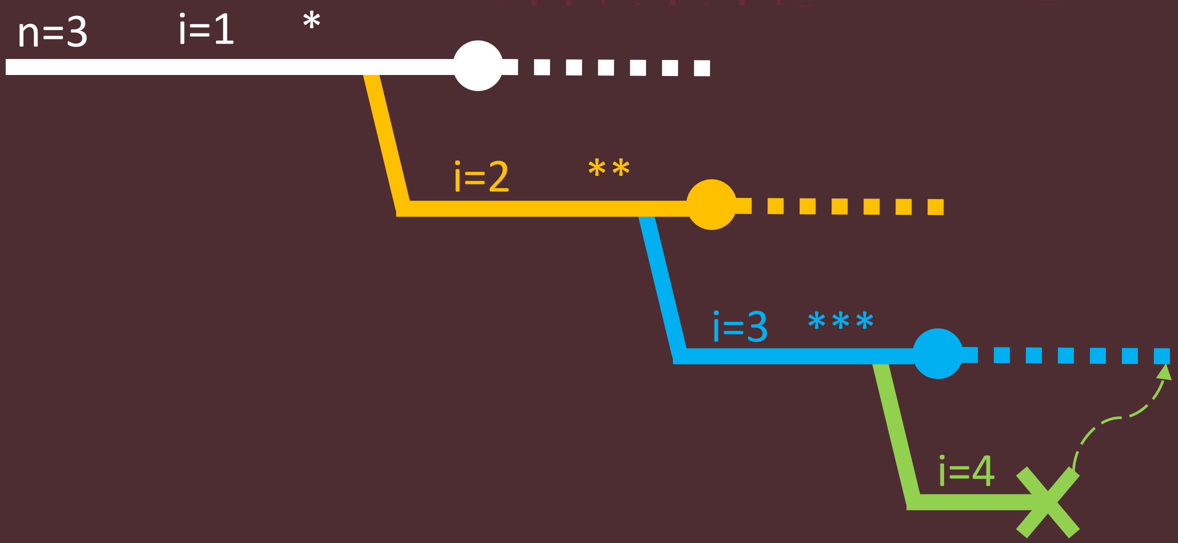
stdout

```
*
**
***
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



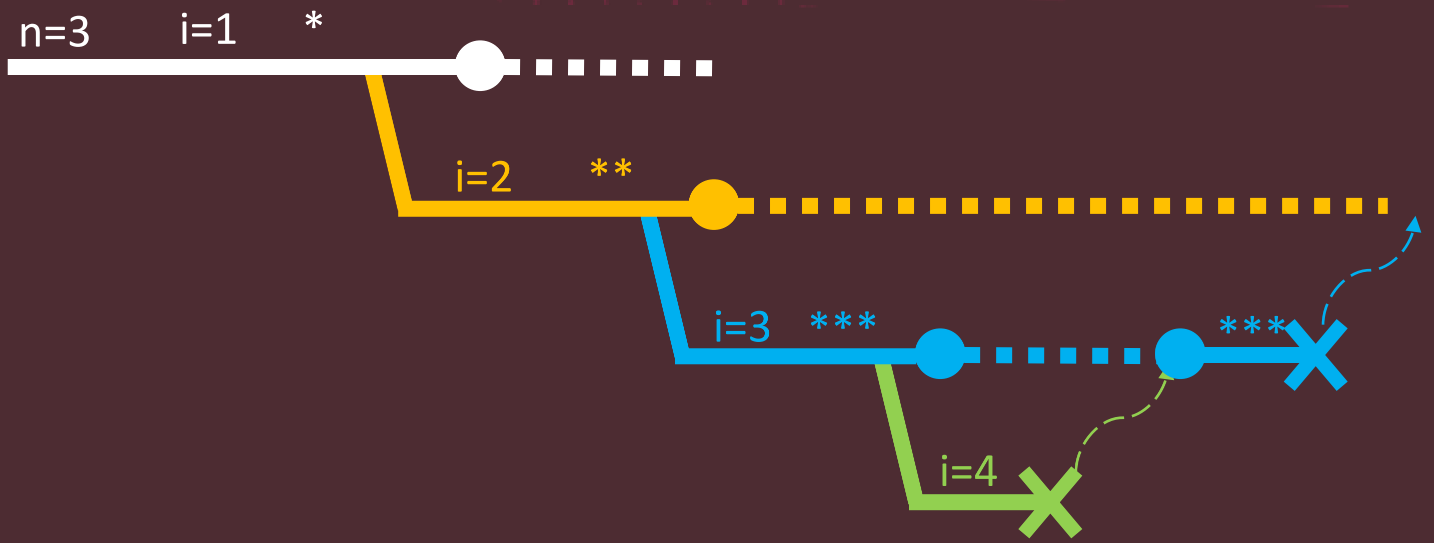
stdout

```
*
**
***
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



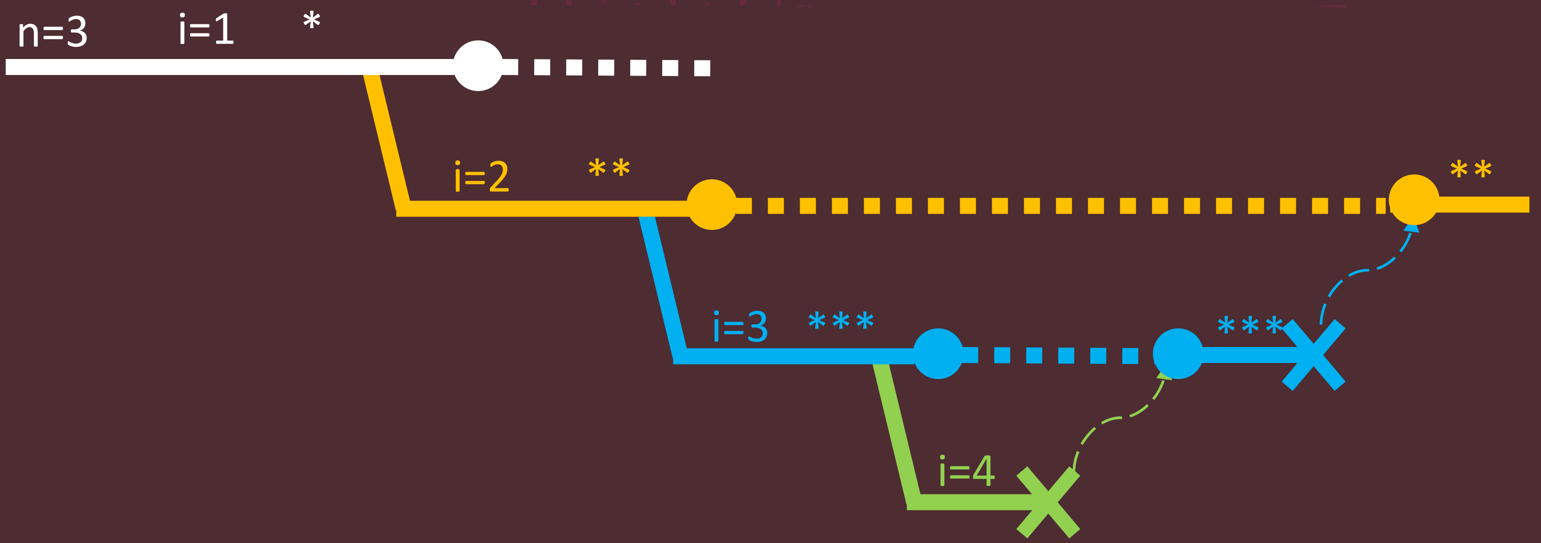
stdout

```
*
**
***
***
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



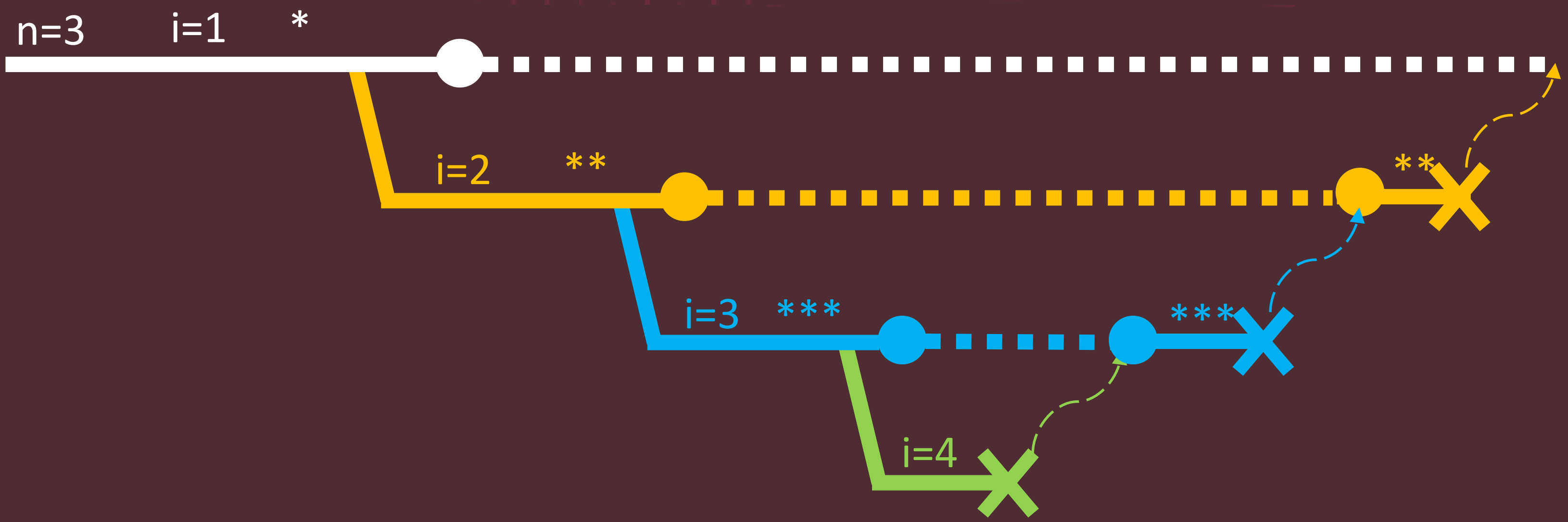
stdout

```
*
**
***
***
**
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



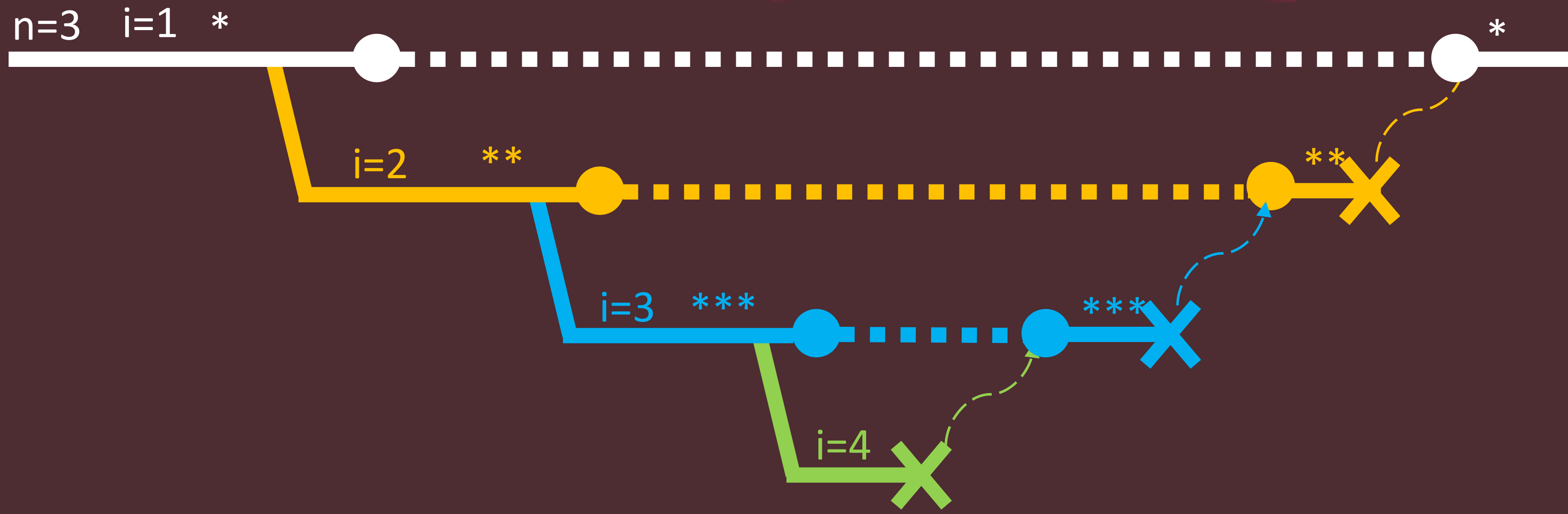
stdout

```
*
**
***
***
**
```

Part-5

```
$ ./starfork-s5 3 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```



stdout

```
*
**
***
***
**
*
```

Part-6

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

```
int execv (const char *path, char *const argv[]);
```

Execute a file/program

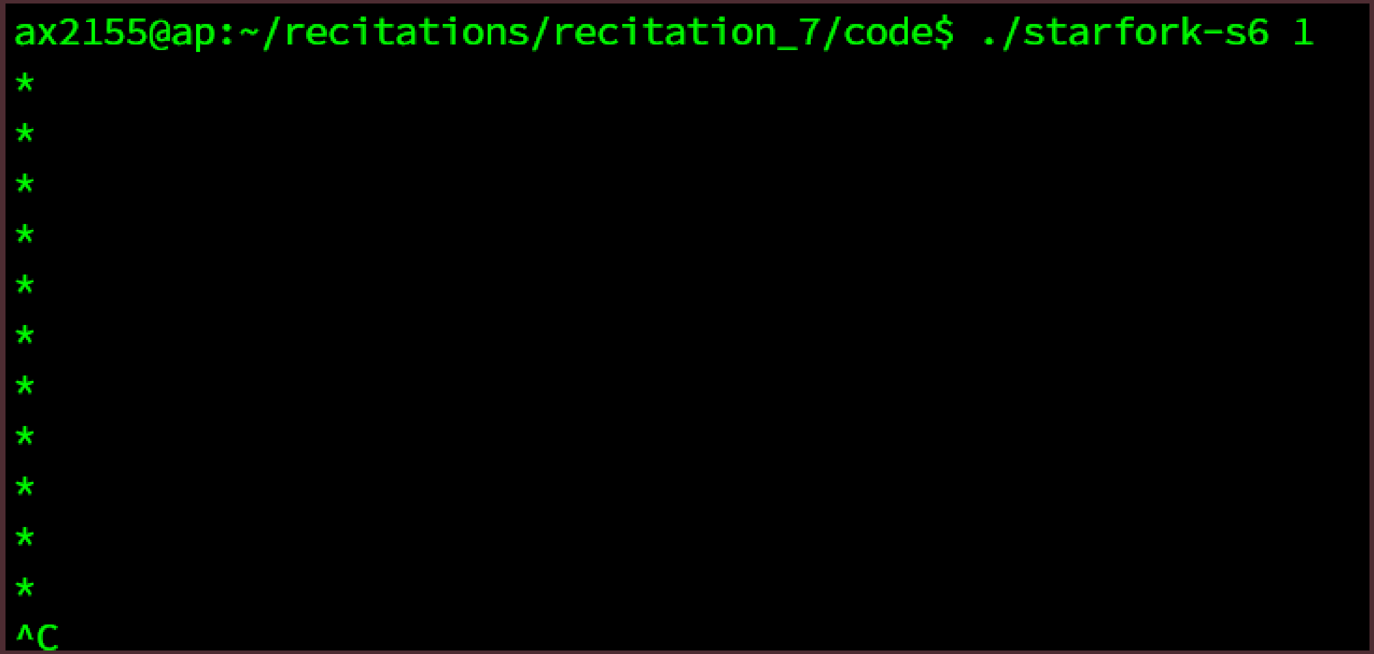
What is the output when running with arguments 1, 2 or 3?

Is “A STAR IS BORN” ever printed?

Are any new processes ever created?

Part-6

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NUL
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```



```
ax2155@ap:~/recitations/recitation_7/code$ ./starfork-s6 1
*
*
*
*
*
*
*
*
*
*
*
^C
```

Part-6

```
$ ./starfork-s6 1
```

n=1 i=1

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

Part-6

```
$ ./starfork-s6 1
```

n=1 i=1 *

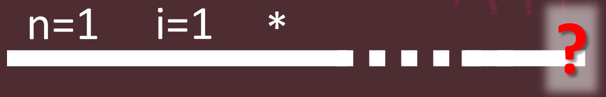
```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*
```

Part-6

```
$ ./starfork-s6 1
```



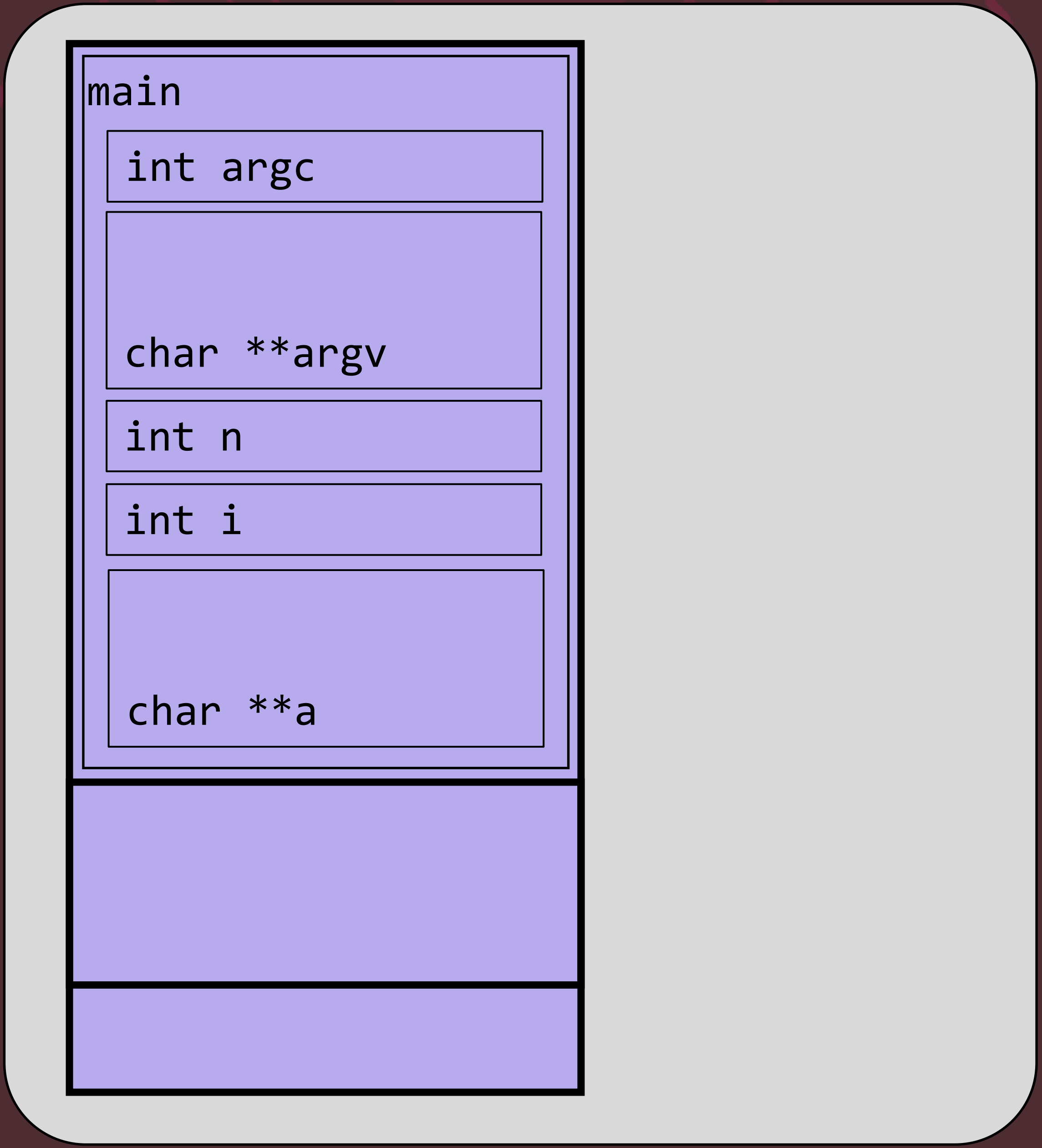
```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*
```

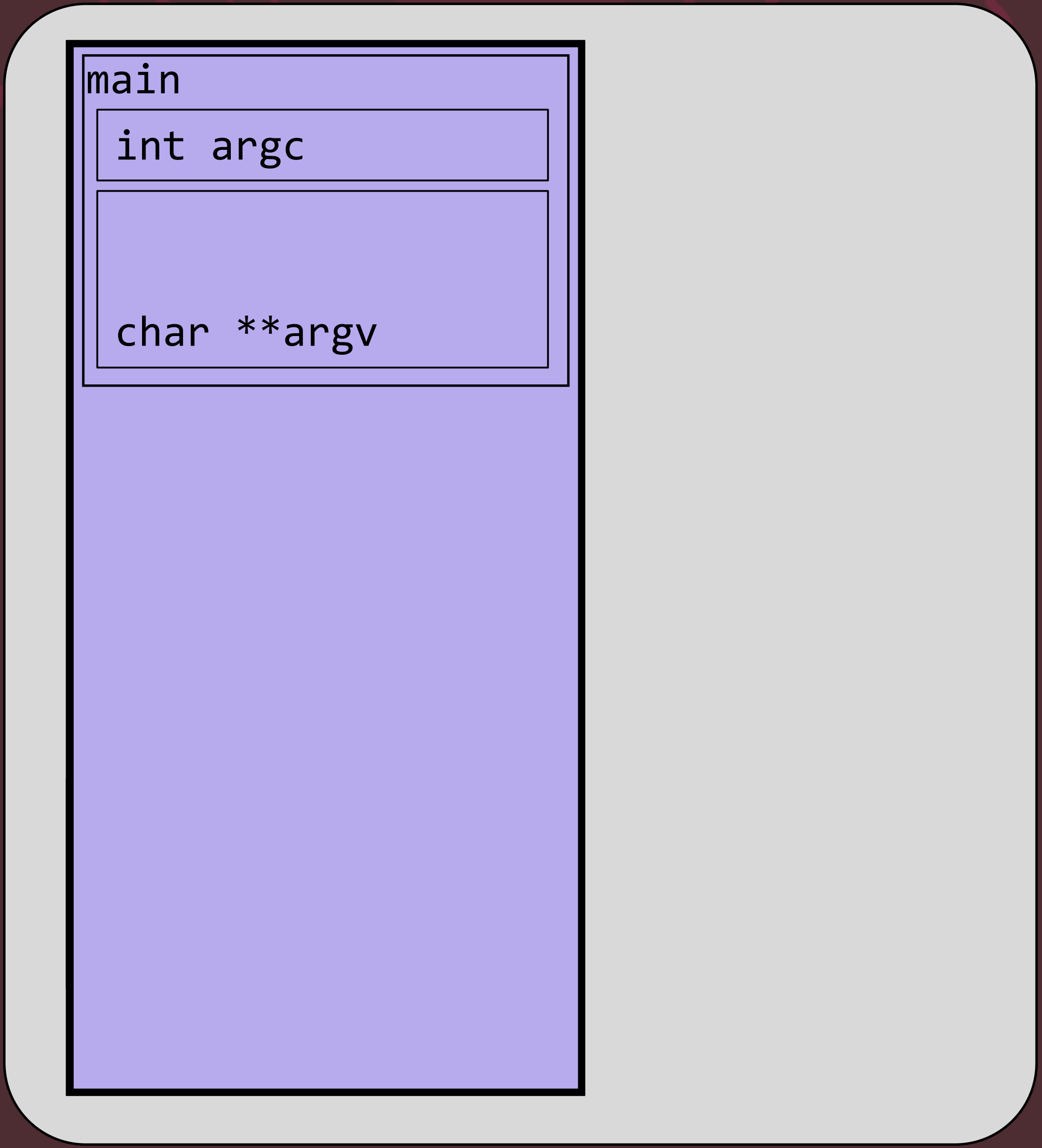
Part-6

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```



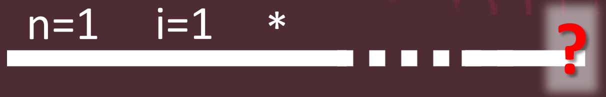
Part-6

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```



Part-6

```
$ ./starfork-s6 1
```



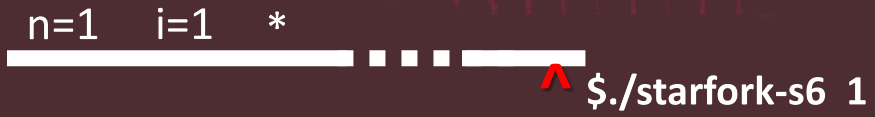
```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*
```

Part-6

```
$ ./starfork-s6 1
```



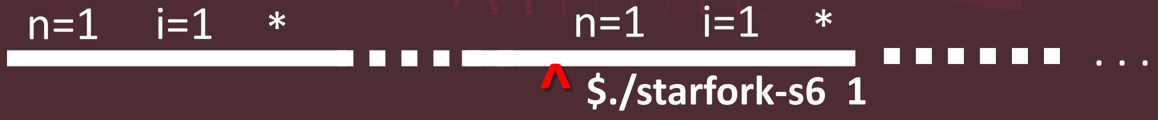
```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*
```

Part-6

$ ./starfork-s6 1


n=1 i=1 * ... n=1 i=1 * ... ...

^ $./starfork-s6 1

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);

    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*
*
```

Part-7

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

```
void star(int numstar) {
    if (numstar >= 100)
        exit(EXIT_FAILURE);
    . . .
}
```

`int sprintf(char *str, const char *format, ...);`

—— Produce output according to a *format* and write to the character string *str*.

What is the output when running with arguments 2, 10 or 50?

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

n=10

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

stdout

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

n=10 10x*

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

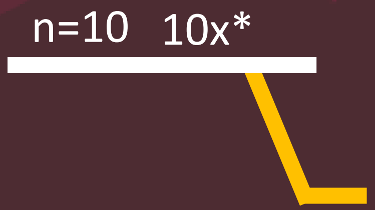
stdout

```
**********
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



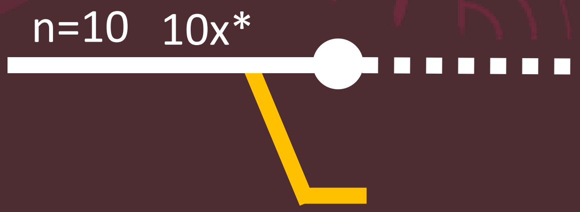
stdout

```
**********
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
```



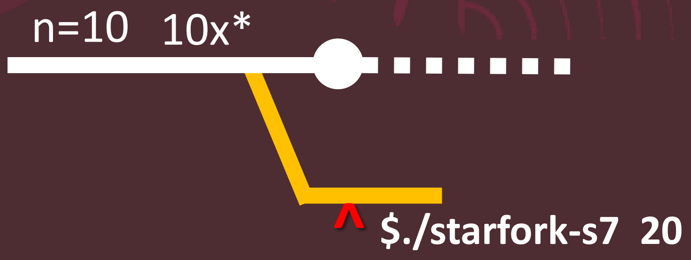
stdout

```
**********
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



$./starfork-s7 20

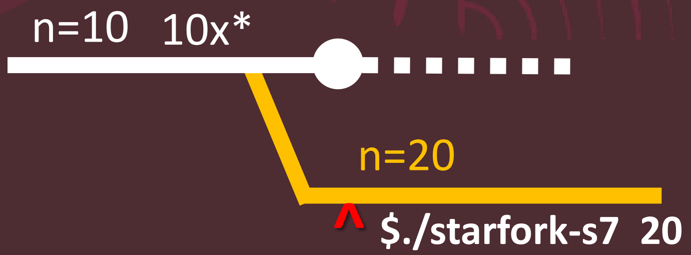
stdout

```
**********
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



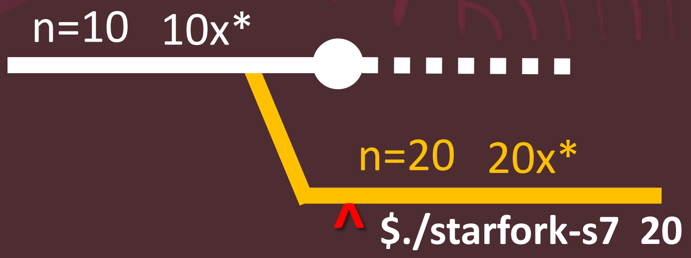
stdout

```
**********
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



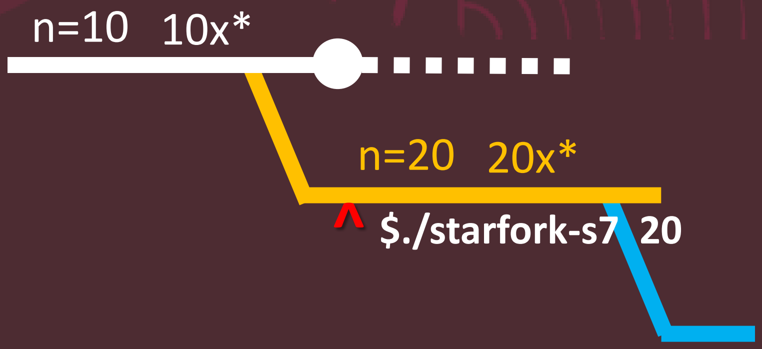
stdout

```
**********
********************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



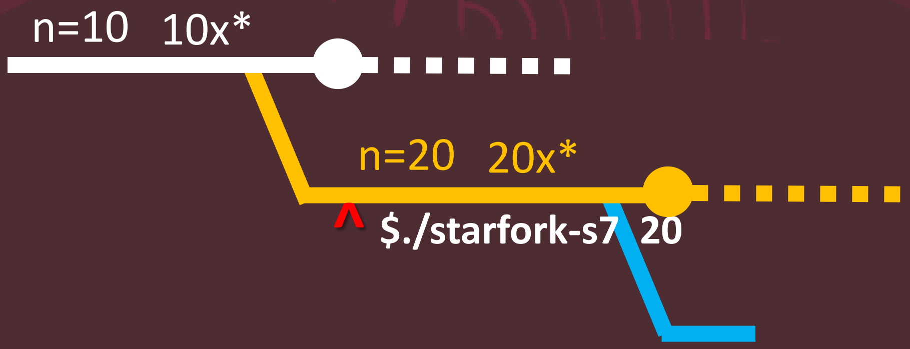
stdout

```
**********
********************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



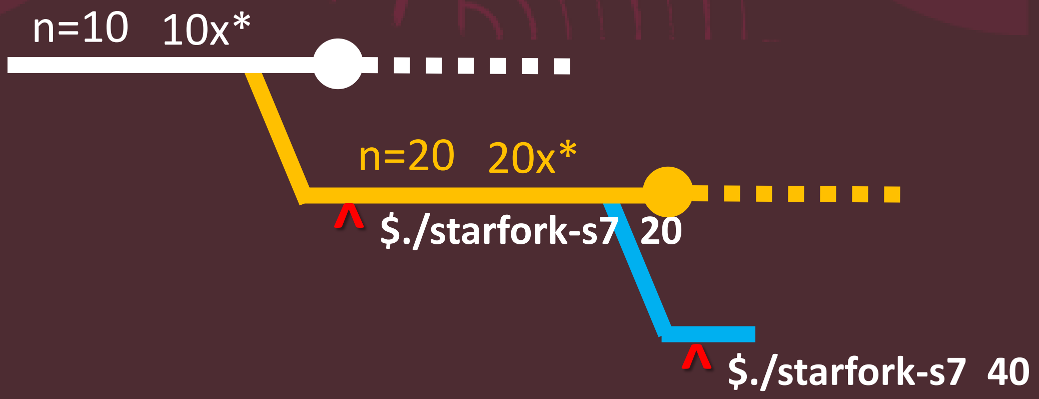
stdout

```
**********
********************
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



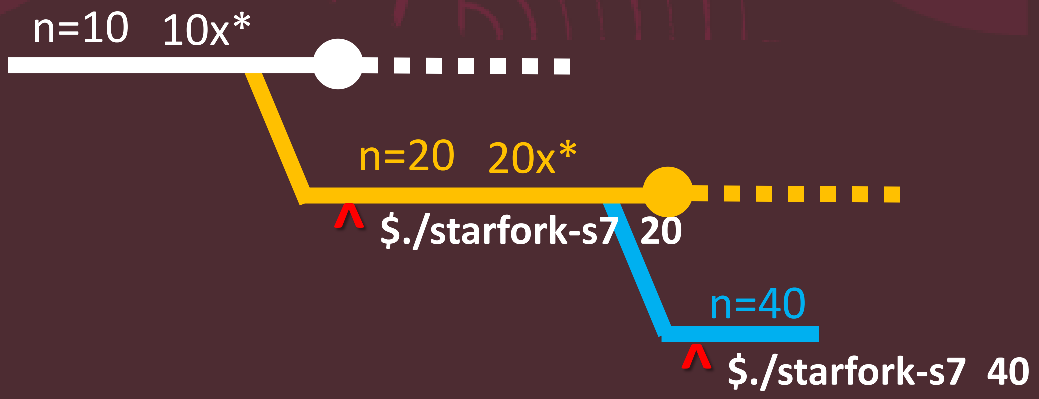
stdout

```
**********
********************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



```
$./starfork-s7 20
$./starfork-s7 40
```

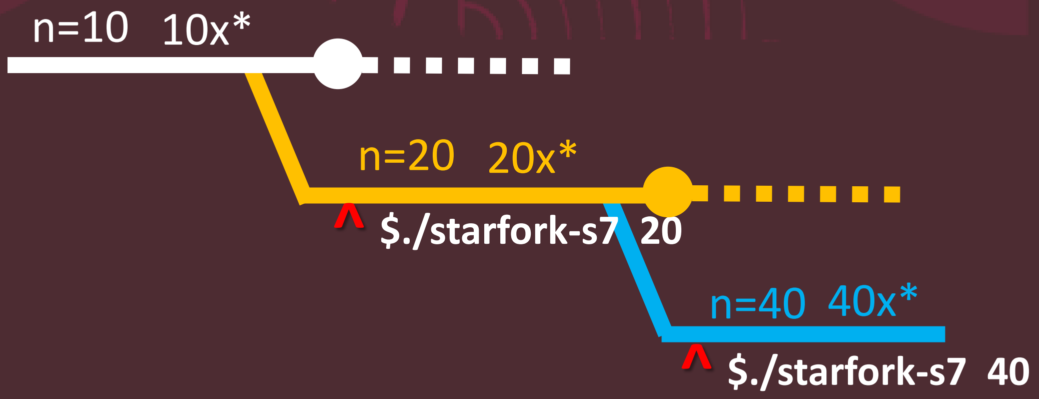
stdout

```
**********
********************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```


n=10 10x*

n=20 20x*

^ $./starfork-s7 20

n=40 40x*

^ $./starfork-s7 40

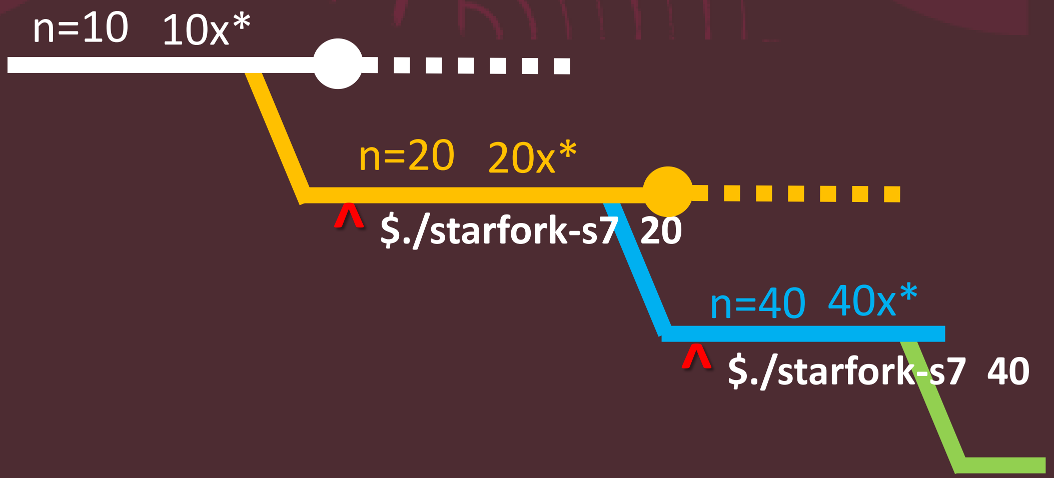
stdout

```
**********
********************
****************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



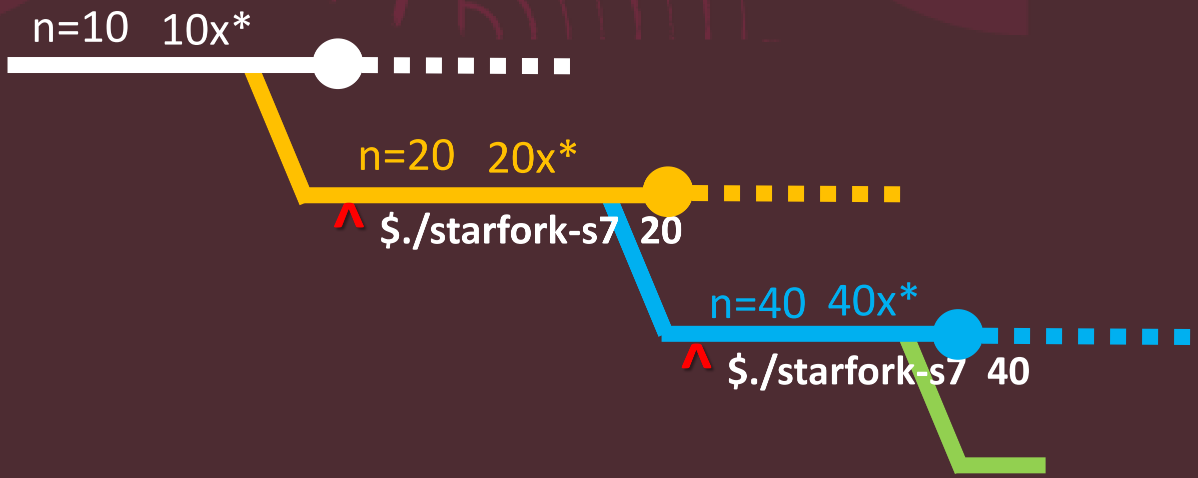
stdout

```
**********
********************
****************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



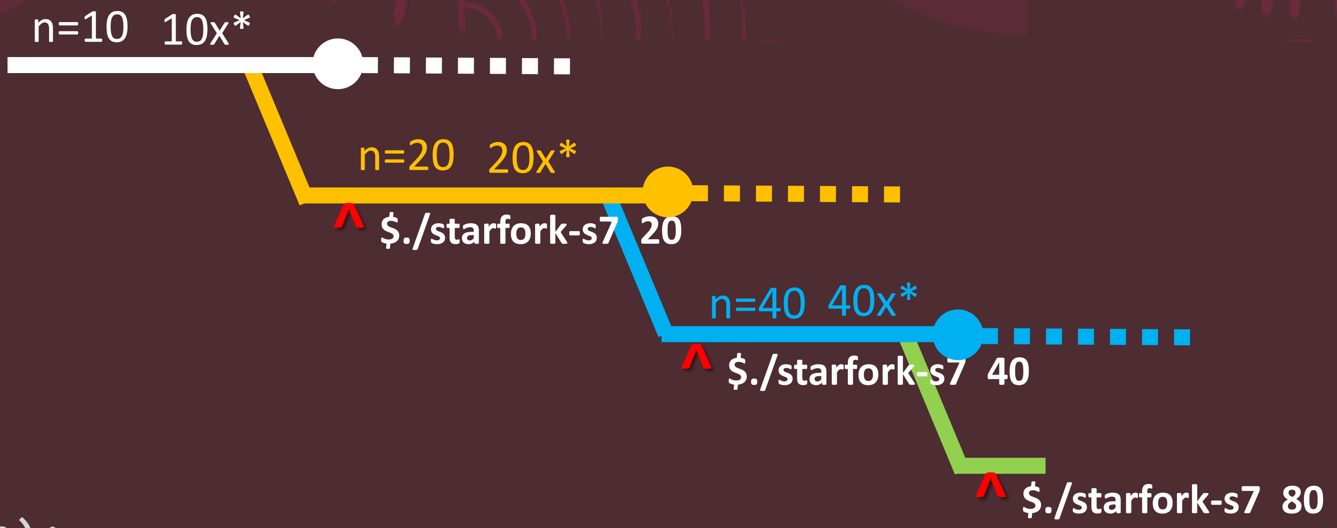
stdout

```
**********
********************
****************************************
```

Part-7

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

```
$ ./starfork-s7 10 2>/dev/null
```



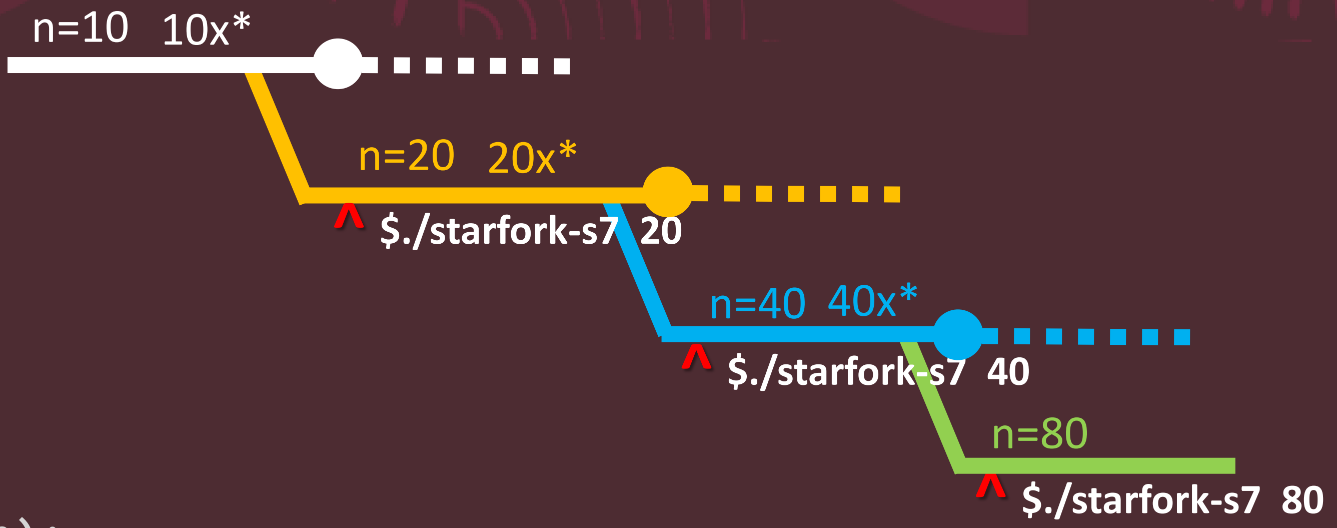
stdout

```
**********
********************
****************************************
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



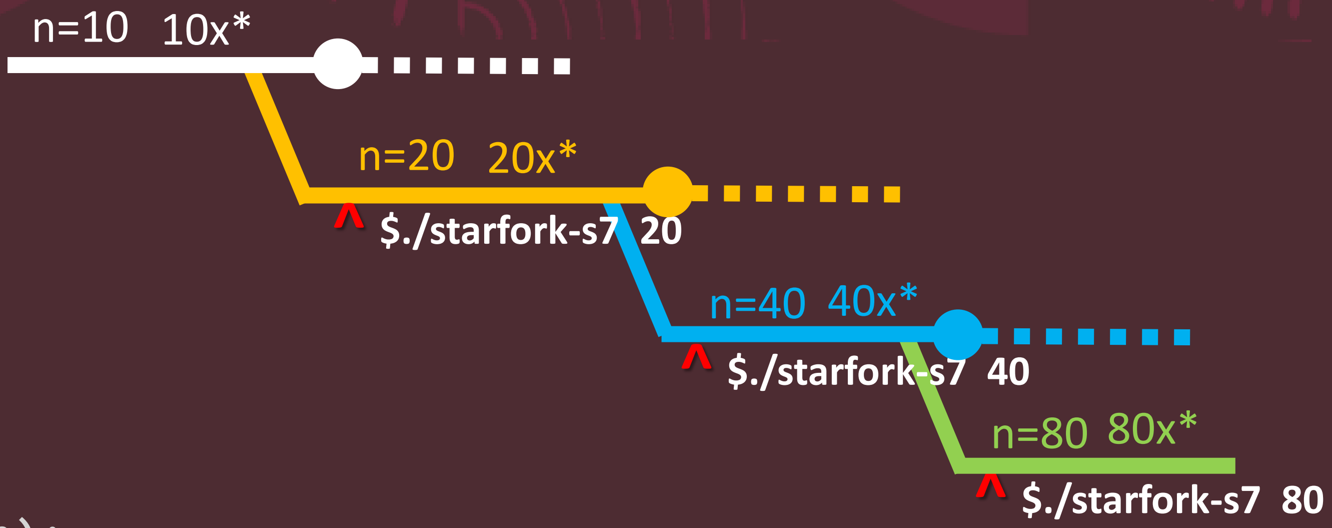
stdout

```
**********
********************
****************************************
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



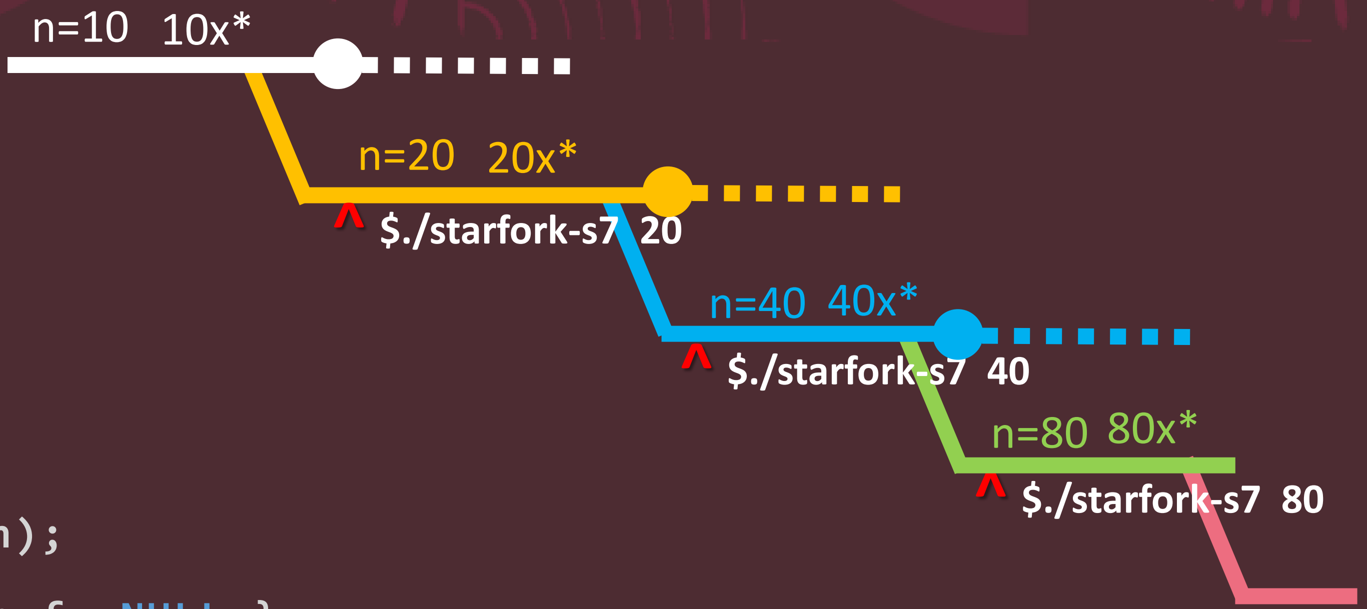
stdout

```
**********
********************
****************************************
********************************************************************************
```

Part-7

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

```
$ ./starfork-s7 10 2>/dev/null
```



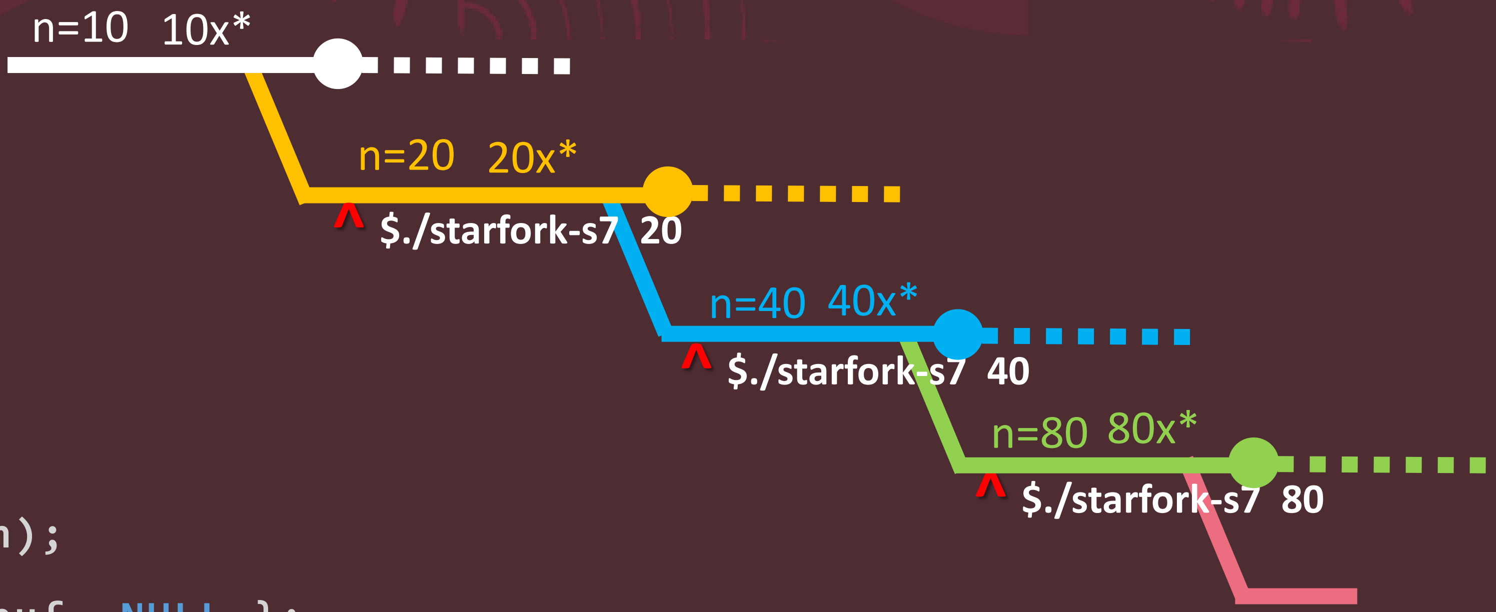
stdout

```
**********
********************
****************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



n=10 10x*

n=20 20x*

^ $./starfork-s7 20

n=40 40x*

^ $./starfork-s7 40

n=80 80x*

^ $./starfork-s7 80

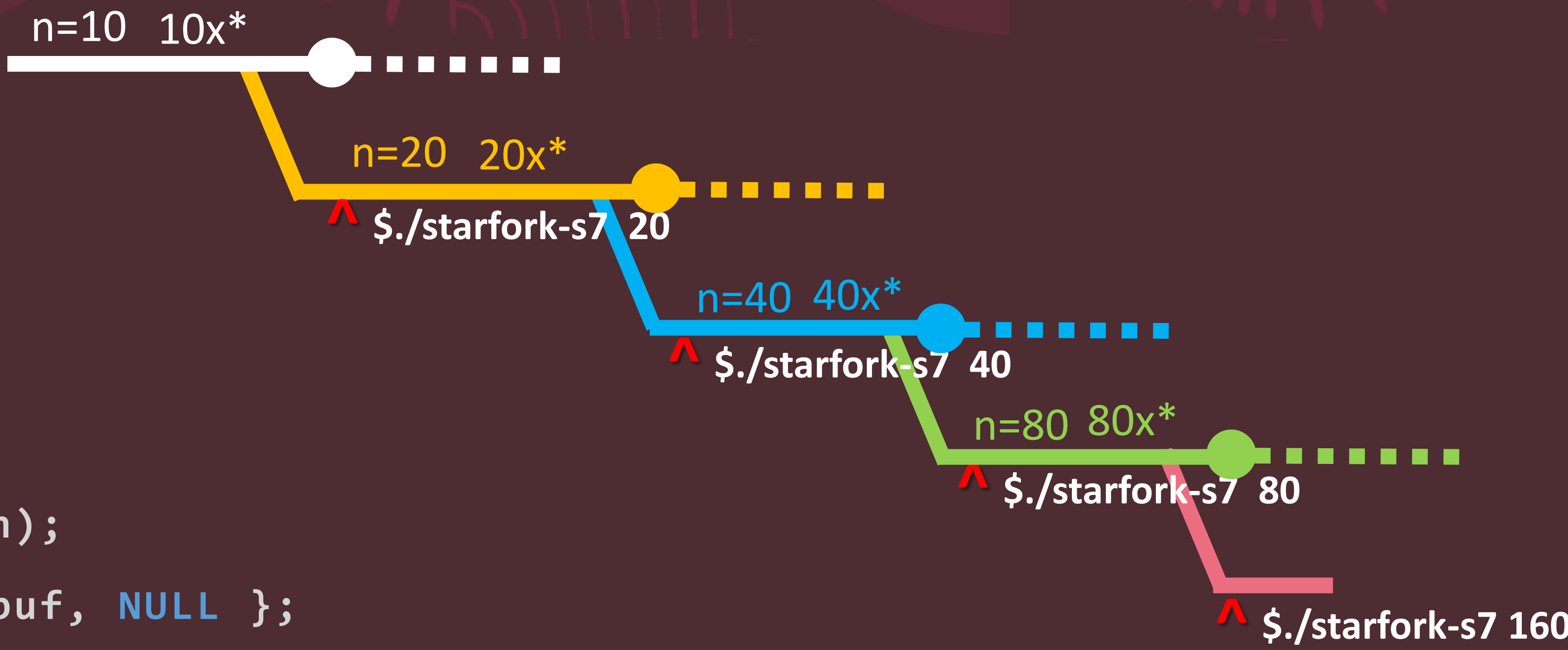
stdout

```
**********
********************
****************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



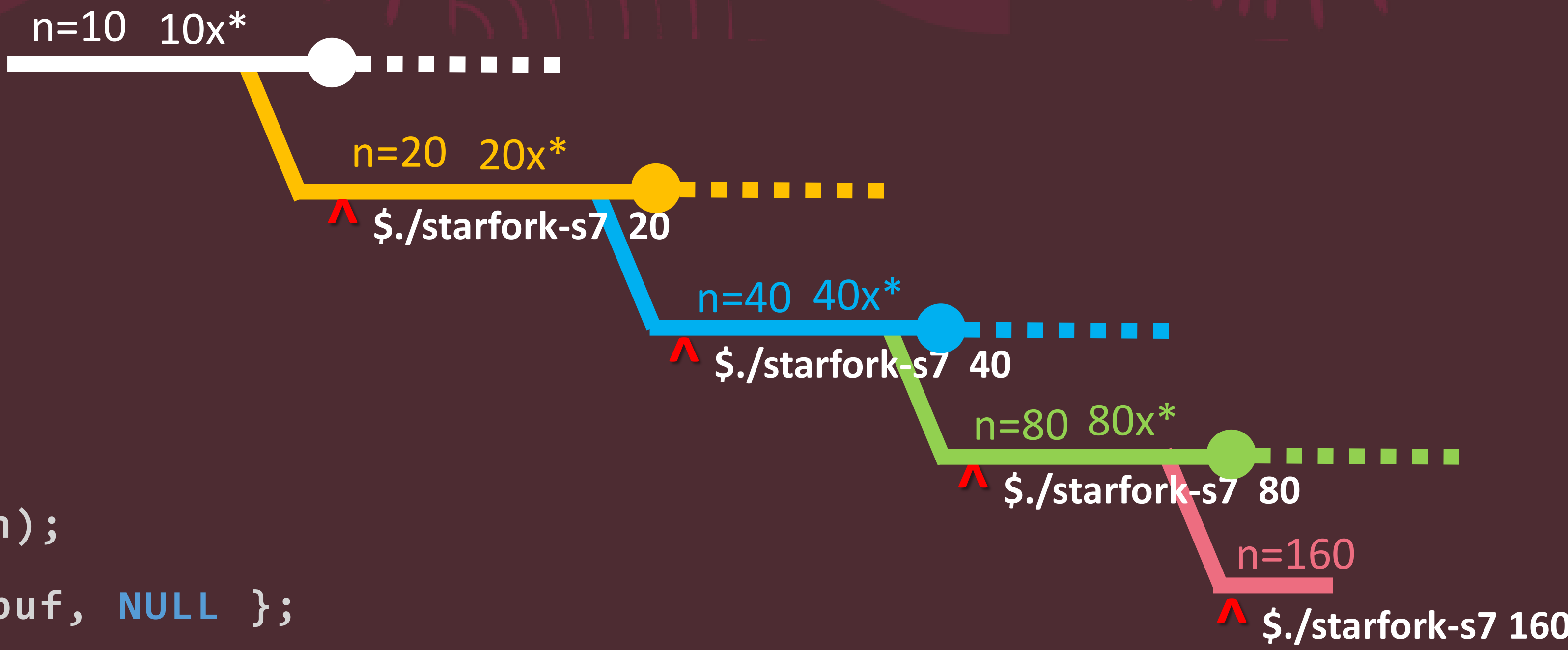
stdout

```
**********
********************
****************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7  10  2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



```
^ $./starfork-s7  20
^ $./starfork-s7  40
^ $./starfork-s7  80
^ $./starfork-s7 160
```

stdout

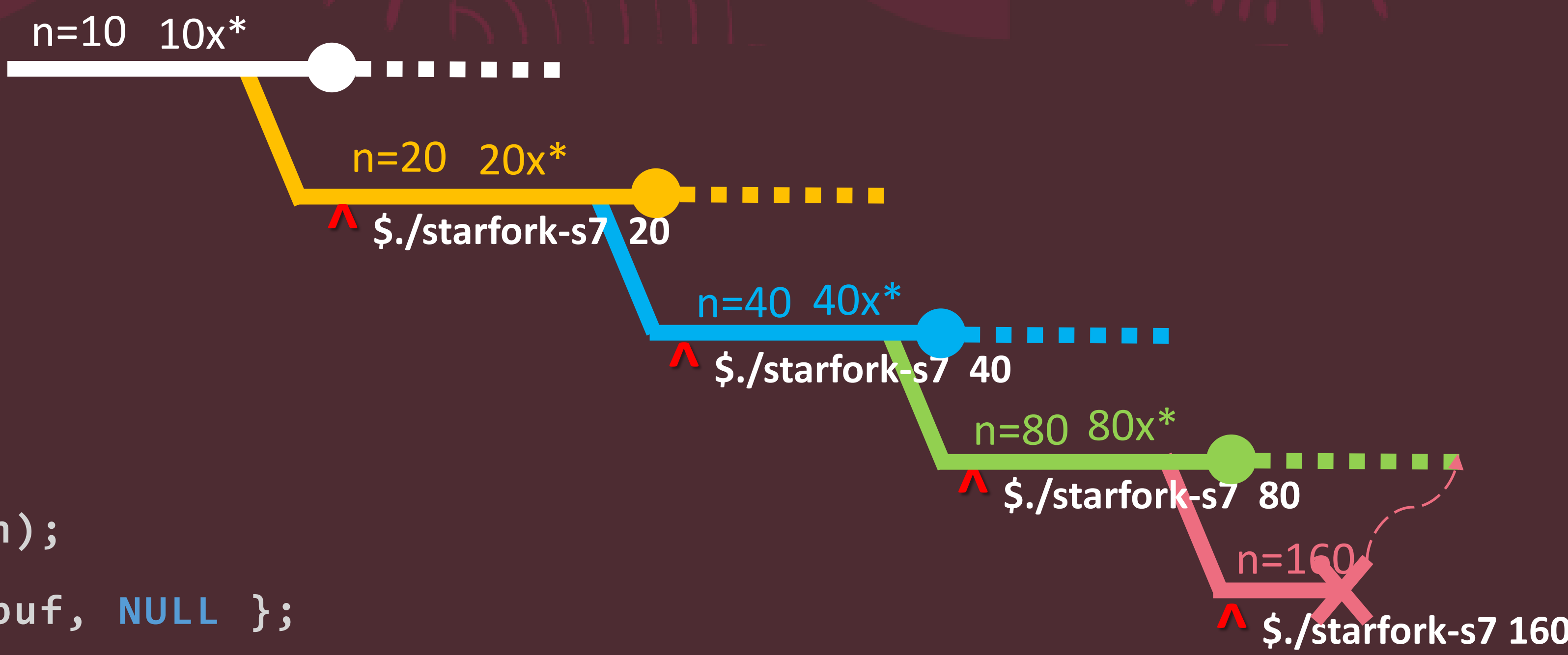
```
**********
********************
****************************************
********************************************************************************
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {

        star(n);

        pid_t pid = fork();
                                  hild

                                 ", 2 * n);

        char *a[] = { argv[0], buf, NULL };

            execv(*a, a);

        }

        waitpid(pid, NULL, 0);

        star(n);

        exit(EXIT_SUCCESS);

    }
}
```

```
void star(int numstar) {
    if (numstar >= 100)
        exit(EXIT_FAILURE);
    . . .
}
```



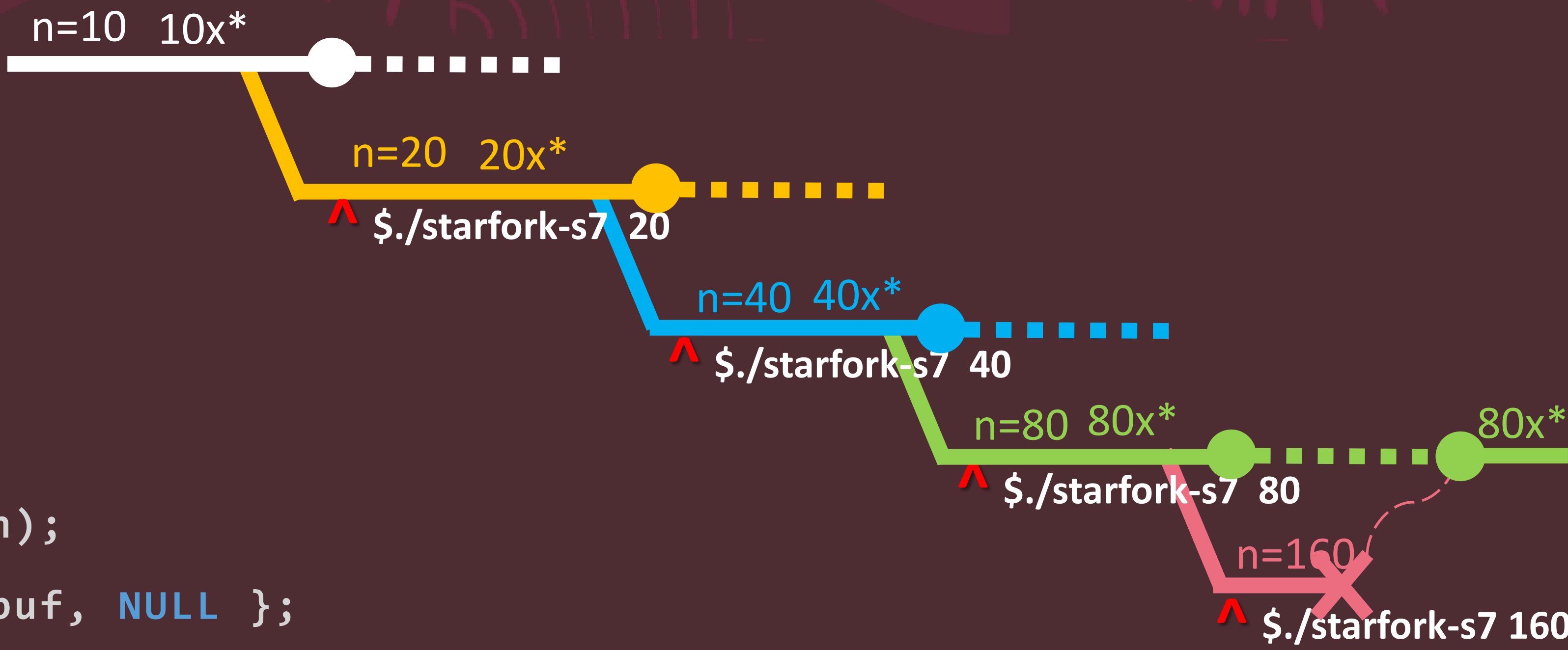
stdout

```
**********
********************
****************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```c
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



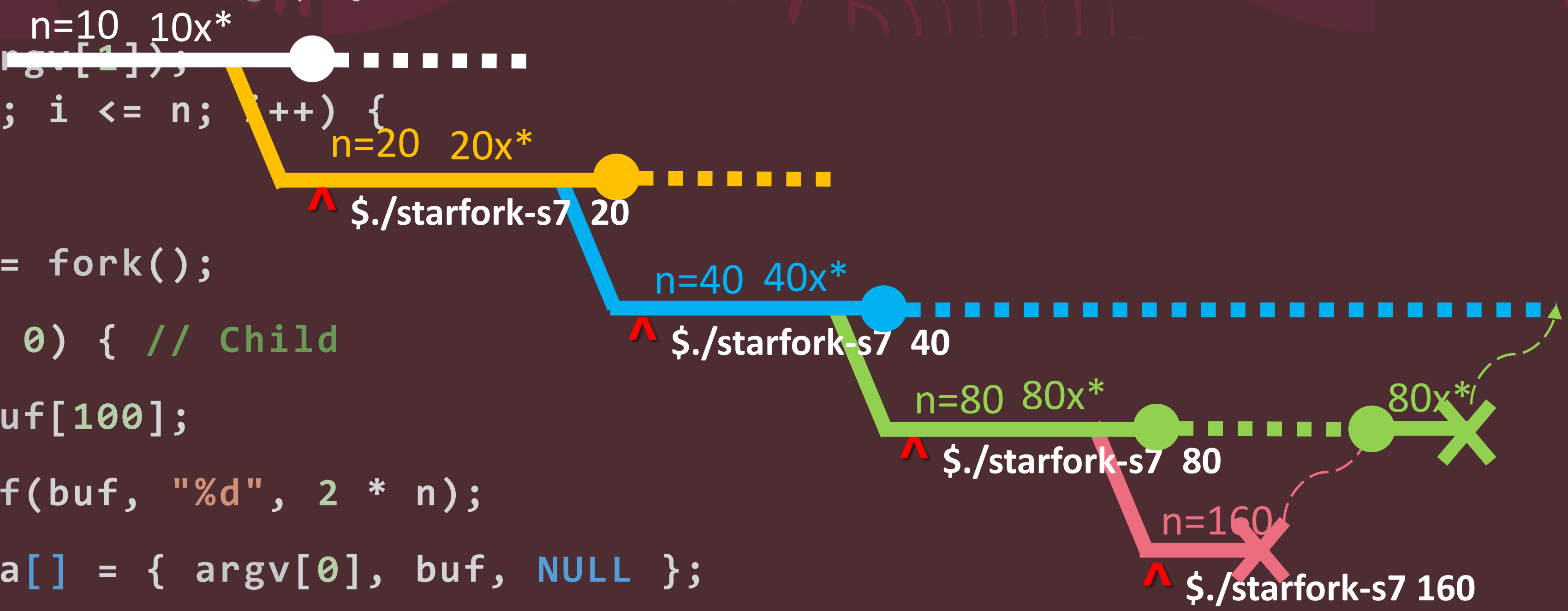
stdout

```
********************
****************************************
********************************************************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



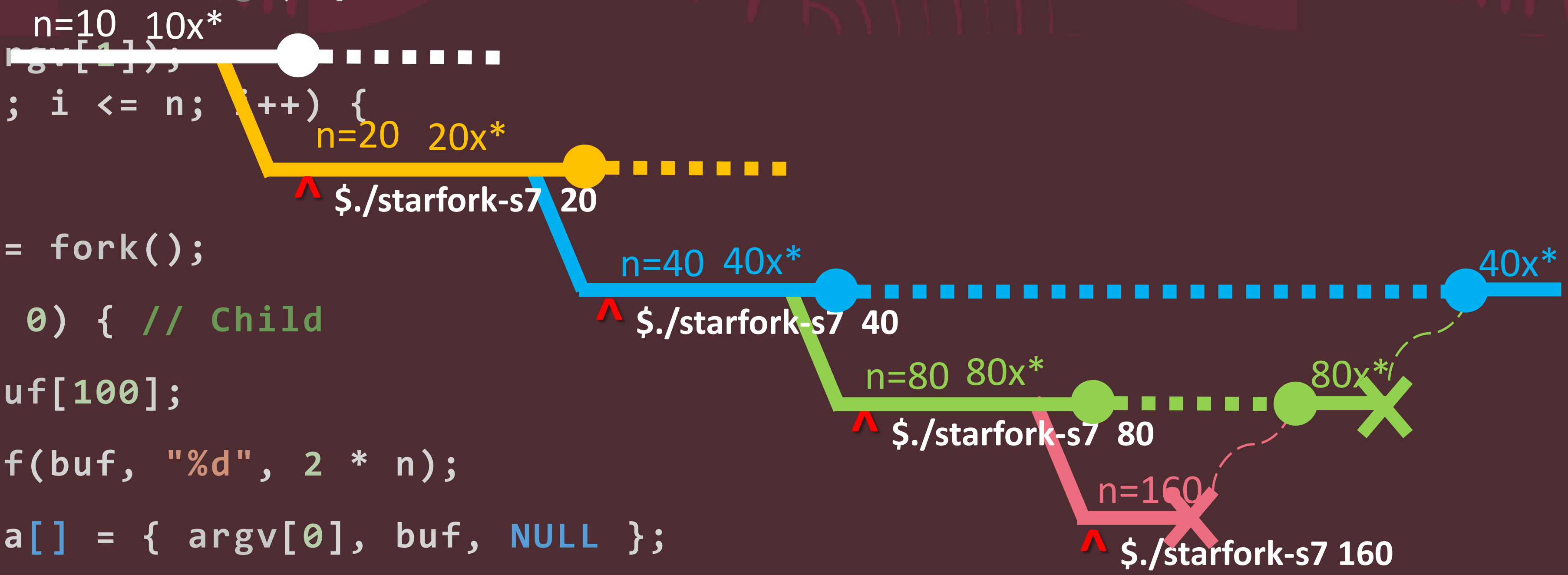
stdout

```
********************
****************************************
********************************************************************************
********************************************************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



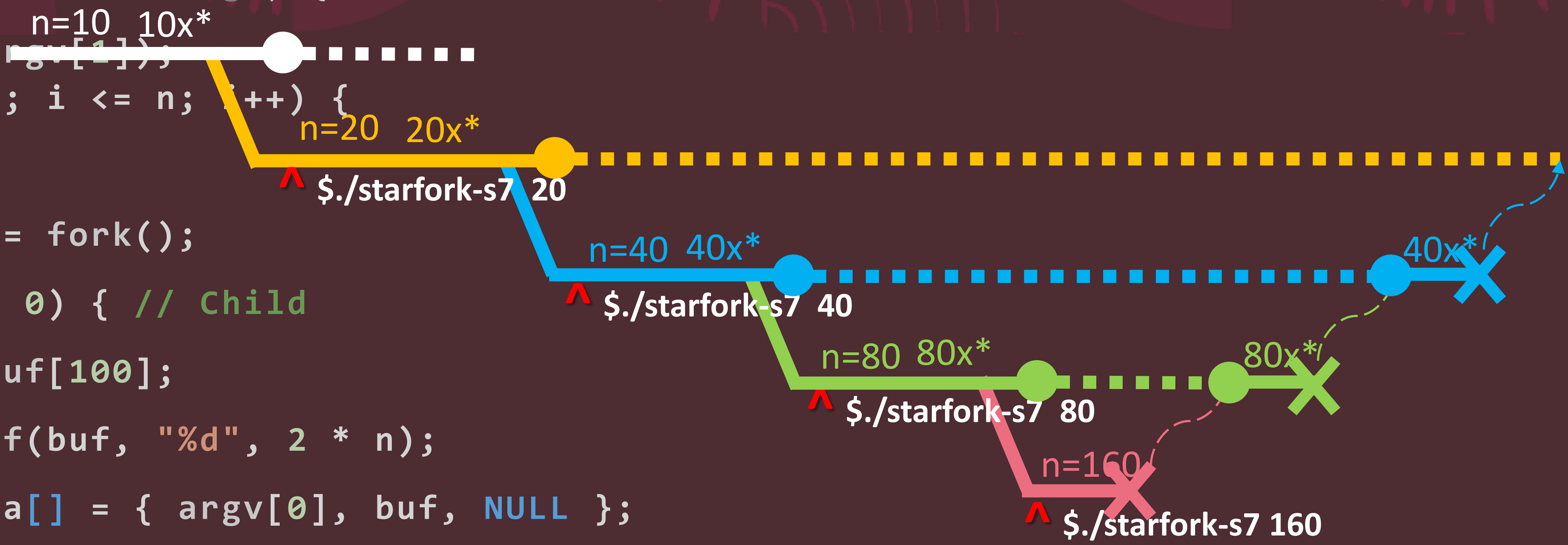
stdout

```
****************************************
********************************************************************************
********************************************************************************
****************************************
```

Part-7

```
$ ./starfork-s7  10  2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



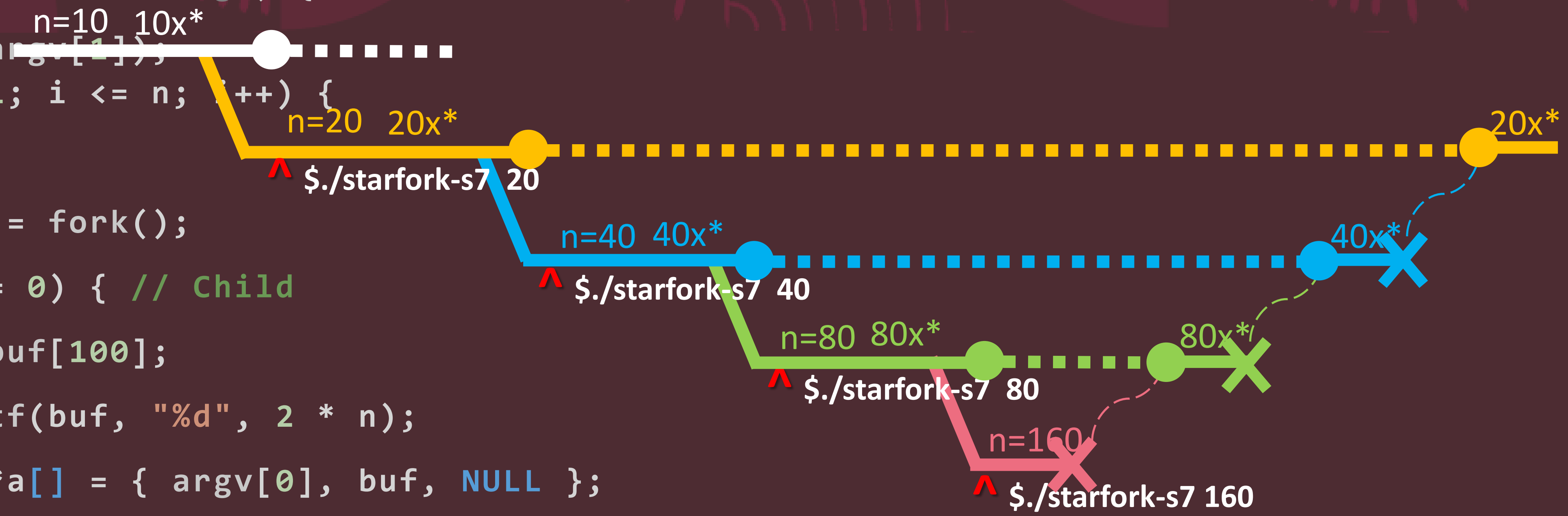
stdout

```
****************************************
********************************************************************************
********************************************************************************
****************************************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



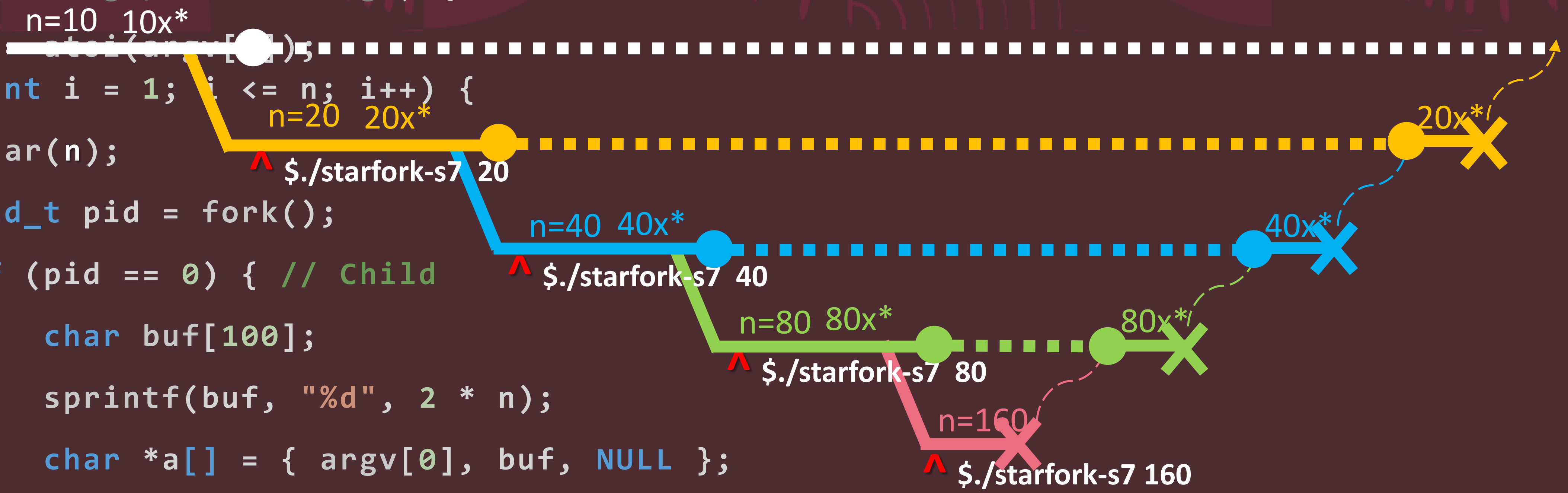
stdout

```
********************************************************************************
********************************************************************************
****************************************
********************
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
```



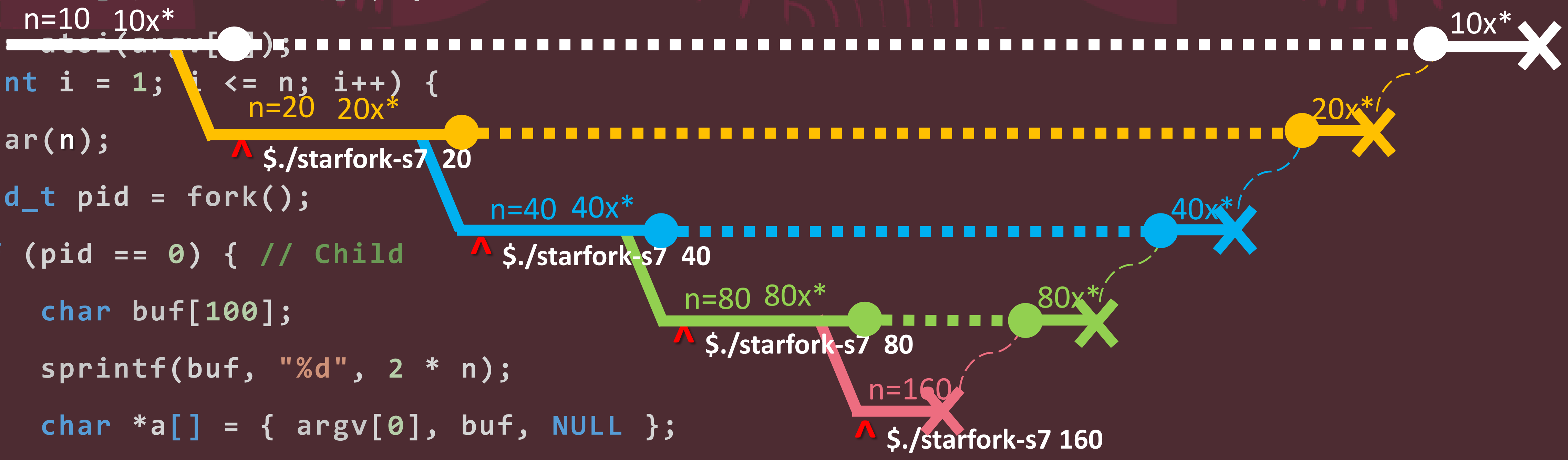
stdout

```
********************************************************************************
********************************************************************************
****************************************
********************
```

Part-7

$ ./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

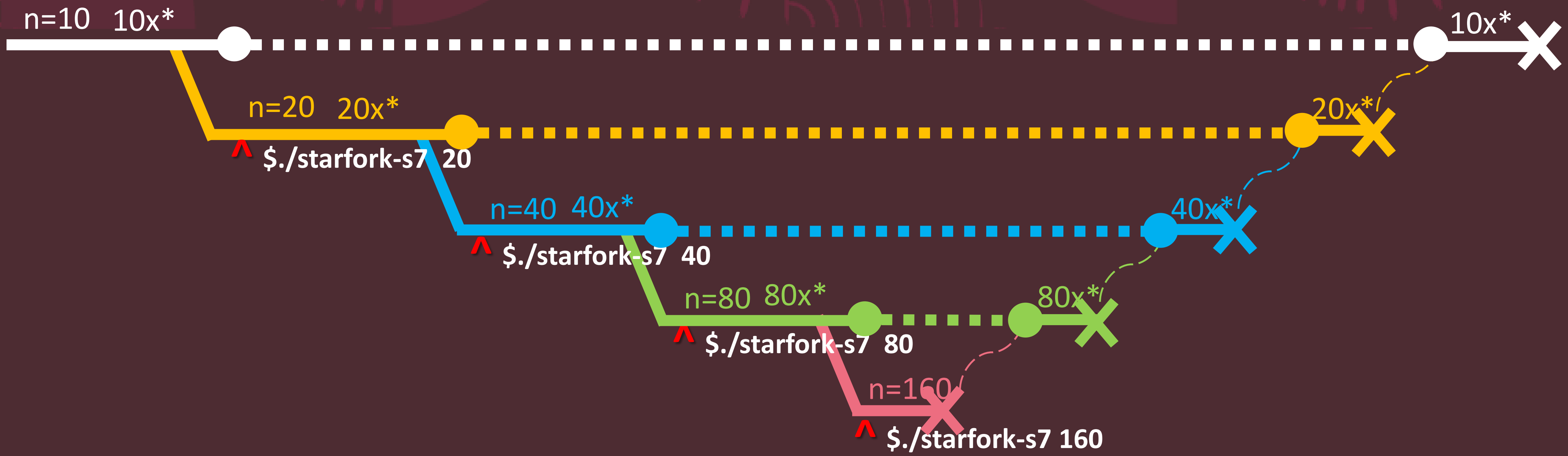


stdout

```
********************************************************************************
****************************************
********************
**********
```

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```



stdout

```
**********
********************
****************************************
********************************************************************************
********************************************************************************
****************************************
********************
**********
```