```
program : sorting(PARAM: cycles, countQuality, countUnQuality)
(Entry) [VAR: cycles, countQuality, countUnQuality, info1, info2, mark1, mark2, mark,
        picked]
        {
(1)         TRIGGER (pathSensor == TRUE) {
(2)             cycles := cycles + 1;
(3)             IF (cycles % 2 == 1) {
(4)                 PATHSET << 'path1';
                }
(5)             ELSIF (cycles % 2 == 0) {
(6)                 PATHSET << 'path2';
                }
            }
(7)         TRIGGER (sensor1 == TRUE) {
(8)             info1 << SREAD1;
(9)             mark1 := checkProduct(info1);
(10)            mark1 >> SWRITE1;
             }
(11)        TRIGGER (sensor2 == TRUE) {
(12)            info2 << SREAD2;
(13)            mark2 := checkProduct(info2);
(14)            mark2 >> SWRITE2;
            }
(15)        TRIGGER (sortSensor == TRUE) {
(16)            mark << SCANNER;
(17)            IF (mark == TRUE) {
(18)                SORTSET << 'Quality';
(19)                countQuality := countQuality + 1;
                }
(20)            ELSE  {
(21)                SORTSET << 'UnQuality';
(22)                countUnQuality := countUnQuality + 1;
                }
(23)            picked := TRUE;
            }

(24)        TRIGGER (picked == TRUE) {
(25)            IF(countUnQuality / countQuality > 0.01){
(26)                STOP;
                }
(27)            picked := FALSE;
            }
        }
```