CS343 A6 Testing Doc

Aidan Campbell - aj6campb
Brendan Engelman - badengel

Case 1 - Lost WATCard

For this case, the given soda.config file was used.

```
@ubuntu2004-010% ./soda soda.config 127
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S
                        R0              S                               S2      S2
                C0,5    R1      S       G3      S1,5    S3,3
D0,2            W       R2                                                              t0,5
                C1,5    N0,0                    V0
D1,2                    N1,1                            V1
D1,2    D2                      P3      P
                                d0,3                            r
                                U0,5
                                U0,3
                                U0,5
D0,3                            U0,4                            R                       L0
                C0,5            D0,0    G4      L               B1,1
D1,3                                            G1,0                                    t1,5
        D2      W       N1,2                            V2                              T1,5
        F                       P4      P
                                d1,4                                    r
                                U1,4
                                U1,3
                                U1,5
D0,2            W               U1,4                                    R               t0,5
D0,1                    N1,0    D1,0    G9              V0
D0,3                            P9      P
                                d2,9                                            r
                                U2,3
                                U2,4
                                U2,2
                                U2,2                    G3,0    B3,0                    T0,5
D0,1                    N1,1    D2,0    G4      B1,3    V1      B1,0            R
                                P4      P
```

In the output above, we can see that Student 0 attempts to get their initial WATCard as shown by the “C0,5” message in WATOff and “t0,5” in courier 0. Unfortunately, the courier loses the WATCard. This is correctly displayed by courier 0 printing “L0” and the student printing “L”. Another request is sent to WATOff (“C0,5”). It is correctly sent to the back of the job queue as shown by how courier 0 handles this request after it finishes creating a WATCard for student 1. This time, the courier successfully creates the WATCard (“T0,5” is printed). WATOff also recognizes when the requestWork call is complete (through “W”). Afterwards, we see that student 0 successfully buys a soda with that WATCard (“B1,3”). This output is in line with the specification.

Case 2 - Insufficient Funds

For this case, the given soda.config file was used.

```
@ubuntu2004-002% ./soda soda.config 123
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S

                                                ...
                                U1,5            B3,1                    R       B3,2
D1,2                            D1,0    P
                                P9
                                d2,9                                            r
                                U2,1                                            R
                                D2,4
                                d0,4                            r
                                U0,2
                                U0,3
D1,1                            U0,5    G8                      R
                                D0,3
                                d1,3                                    r
                                U1,3
                                U1,5
                T0,7            U1,5                                    R
                                D1,0                                                    t0,7
D1,1            W               P8      P                                               T0,7
                                d2,8                                            r
                                D2,7                                            R
                                d0,7                            r
                                U0,2
D1,1                            U0,3    G2      B3,6            R               B3,4
```

In the output above, we see that after Student 0 buys a soda (“B3,1”), their WATCard balance becomes $1. Since this balance is less than the soda cost ($2), they will have insufficient funds on their next purchase. Accordingly, we see that in the WATCard Office column, a transfer is made for $7. This is the correct transfer value since it is equal to the cost of a soda plus $5. After, we see that Courier 0 starts and completes this transfer (without losing the WATCard). We can also see that the WATCard Office recognizes that the call to requestWork by the courier is complete. After a few more frames, the student makes a successful purchase of soda. Their new balance is correct since existing balance + $7 - soda cost = $6. It’s also worth noting that the purchase does not occur immediately after the funds have been transferred. This is in line with how a student must randomly yield (for 1-10 seconds) before attempting another purchase.

Case 3 - Multiple Free Sodas

For this case, the given soda.config file was used.

```
@ubuntu2004-010% ./soda soda.config 127
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S
                        R0              S                               S2      S2
                C0,5    R1      S       G3      S1,5    S3,3
D0,2            W       R2                                                              t0,5
                C1,5    N0,0                    V0

                                                ...
D0,3                            P4      P                       A                         T0,7
                                d1,4                                    r
D0,2                            U1,1            A1,8                    R
                                D1,2
                                d2,2                                            r
                                U2,2
                                U2,4    G7                                      R
                                D2,2
                                d0,2                            r
                                U0,5
                                U0,2
                                U0,2                            R
D1,3                            D0,2            A1,8            A
D0,1                            P7      P       B1,6            B1,1
D1,2                            d1,7                                    r
                                D1,6                                    R
```

In the output above, we see that Student 0 starts with vending machine 0. In the excluded code (represented by "..."), student 0 does not get a new vending machine. Later in the output, we can see that machine 0 gave out a free soda. Student 0, whose vending machine is 0, received this free soda. Accordingly, Student 0 printed the appropriate message ("A1,8"). We see that Machine 0 gave out another free soda so Student 0 prints out another "A1,8" message. Finally, Student 0 buys a soda without getting one for free ("B1,6") and their WATCard balance is updated appropriately. Student 0 receives multiple free sodas in a single buy attempt. This scenario is possible under the specification. Moreover, we see there is a delay between the free soda messages. This is in part due to the student yielding while drinking the soda (note that "B1,6" comes immediately after "A1,8" despite there being a yield).

Case 4 - Machine Out of Stock

For this case, the given soda.config file was used.

```
@ubuntu2004-010% ./soda soda.config 127
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S
                        R0              S                               S2      S2
                C0,5    R1      S       G3      S1,5    S3,3
D0,2            W       R2                                                              t0,5
                C1,5    N0,0                    V0
D1,2                    N1,1                            V1
D1,2    D2                      P3      P
                                d0,3                            r
                                U0,5
                                U0,3
                                U0,5
D0,3                            U0,4                            R                       L0
                C0,5            D0,0    G4      L               B1,1
D1,3                                            G1,0                                    t1,5
        D2      W       N1,2                            V2                              T1,5
        F                       P4      P
                                d1,4                                    r
                                U1,4
                                U1,3
                                U1,5
D0,2            W               U1,4                                    R               t0,5
D0,1                    N1,0    D1,0    G9              V0
D0,3                            P9      P
                                d2,9                                            r
                                U2,3
                                U2,4
                                U2,2
                                U2,2                            G3,0    B3,0            T0,5
D0,1                    N1,1    D2,0    G4      B1,3    V1      B1,0            R
                                P4      P
```

In the output above, we can see that Student 1 starts with vending machine 1 and their favourite soda flavour is 3. Unfortunately, they try to buy soda but V1 does not have any. As a result, they get a new vending machine as shown by the message “V2”. Student 1 encounters the same problem again so they switch machines a second time to V0 (as shown by “V0”). Finally their favourite soda flavour is available so they buy one with their gift card (“G3, 0”). We can see that there are large gaps in between the Vx messages. This is in line with how the student must yield 1-10 before attempting another purchase. We also see that the student keeps switching vending machines until they successfully buy a soda. This is also in line with the specification. We also see that the Student 1 visits vending machines 1 then 2 then 0. This sequence is ascending order, wrapping around at the last machine, as desired.

Case 5 - Flat Tire

For this case, the given soda.config file was used.

```
@ubuntu2004-010% ./soda soda.config 135
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S
                        R0              S                               S2      S2
                C0,5    R1      S       G6      S2,5    S2,6
D0,2            W       R2                                                              t0,5
                C1,5    N0,0                    V0
D0,1                    N1,1                            V1
D0,1    D2
D1,1                            P6      P
                                d0,6                            r
                                U0,3
                                U0,4
                                U0,5
                                U0,2                            R
D0,1                            D0,0    G3
                                P3      P
                                d1,3                                    r
                                U1,4
                                U1,4
                                U1,4
                        N0,1    U1,5            V1                      R               T0,5
                                D1,0
        D2                      X
D1,2    F       W                       G5                                              t1,5
D1,3                    N1,2                    G2,0    V2              B2,0            T1,5
D0,1                    N1,0                            V0
D1,1                    N0,2    P5      P       V2
                                d2,5                                            r
                                U2,3
                                U2,5
```

In the output above we see that the Truck gets a flat tire after delivering to vending machine 1. This is correctly shown through the “X” message. The flat tire also occurs at the correct time, after a delivery is complete (indicated by “D1,0”). We can also see that the Truck does not print anything for several flushes. This is because the Truck is yielding while the flat tire is being replaced.

Case 6 - Empty Production Run

For this case, the given soda.config was used but MaxShippedPerFlavour was changed from 3 to 1 (to make this case more likely).

```
@ubuntu2004-010% ./soda soda.config 140
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S

                                                ...
D0,2                    N1,2    P0      P               V2
                                        G4
                                P4      P
                                d1,4                                    r
```

In the output above, we can see that the truck picks up a shipment of 0 bottles (correctly indicated by “P0”). In our implementation, the truck stops delivering when it runs out of bottles. As a result, when the truck received the shipment of 0 bottles, it immediately stopped “delivering” this shipment and got a new shipment as soon as possible (indicated by “P4”).

Case 7 - Leftover Soda in Truck After Loop

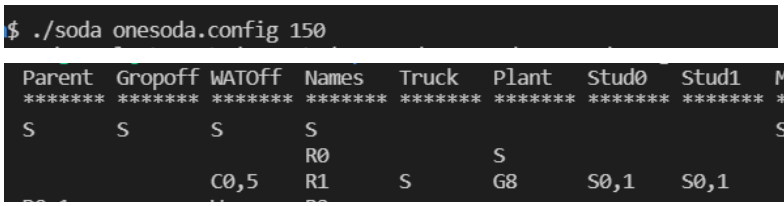
For this case, the given soda.config was used but MaxShippedPerFlavour was changed from 3 to 100 (to make this case more likely).

```
@ubuntu2004-010% ./soda soda.config 140
Parent  Gropoff WATOff  Names   Truck   Plant   Stud0   Stud1   Mach0   Mach1   Mach2   Cour0
******* ******* ******* ******* ******* ******* ******* ******* ******* ******* ******* *******
S       S       S       S                                       S2                      S
                        R0              S                               S2      S2
                C0,5    R1      S       G113    S2,8    S0,5
D0,3            W       R2                                                              t0,5
                C1,5    N0,0                    V0
D1,3                    N1,1                            V1
D0,3    D2              N0,1    P113    P       V1                                      T0,5
                                d0,113                          r
D1,2            W               U0,5                                                    t1,5
                                D0,98                           R                       T1,5
                                d1,98                                   r
                                U1,5    G232                            R
                                D1,83
                                d2,83                                           r
D0,1                    N0,2    U2,5            V2                              R
        D2                      D2,68                   B0,3            B0,4
D0,1    F                                               G0,0            B0,3
D0,1                    N0,0    P232    P       V0
                                d0,232                          r
```

In the output above, we can see that the first shipment has 113 bottles (indicated by “P113”). We also see that after the truck completes the cycle, there are 68 bottles that have not been delivered to the vending machines. This is indicated by “D2,68” which is the last delivered message before the next shipment message (“P232”). The number of bottles in the “P232” command is taken directly from the value given by the plant so we know it refers to the bottles generated by the plant specifically for the second shipment (i.e. it is not affected by the remaining 68 bottles from shipment 1). After, we see the message “d0,232”. This indicates there are 232 remaining bottles in this second shipment, which is equal to the number of new second shipment bottles. Therefore, this implies that the remaining 68 expired bottles from the previous shipment were discarded. Discarding bottles after a complete cycle is the expected behaviour as defined by the specification.

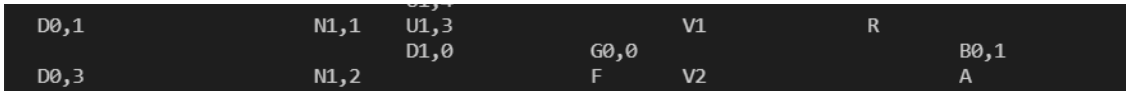
Case 8 - Student Buys Using Gift Card Only

For this test case, we will be using a new config file, onesoda.config
This file is the same as soda.config, but MaxPurchases is 1.



…

…



…

We are interested in specifically the Stud0 column. Here, we can see from the entry ‘S0,1’ that the student decides to only buy 1 soda. From the entry ‘G0,0’ we see that the student uses their gift card for a purchase. We can see here that the student immediately finishes after purchasing their single soda with their gift card, as ‘G0,0’ is followed by ‘F’. This verifies that the program does not encounter abnormal behaviour in the case where a student does not buy any sodas with their WATCard. We can see this is the case even in the case where the student has previously received a WATCard, which we can see based on the entry ‘T0,5’ in the Cour0 column, indicating the student received a WATCard prior to purchasing their soda. Additionally, from the lack of an error message at the end of the program, we can verify that this situation does not cause a memory leak - it is possible that a poorly coded program could leak the memory for the requested WATCard since the card is unused.