Fall 2016
CS 589 PROJECT REPORT

Seungho Han
A20360115

1. Model Based Testing

idle
In: T1, T2, T3, T4, T6, T8, T9, T10, T11, T12, T13, T14, T15 (13)
Out: T2, T3, T4, T5, T6, T7 (6)

Transaction pairs	Test Case	Transaction pairs	Test Case	Transaction pairs	Test Case
(T1, T2)	Test #1	(T2, T2)	Test #2	(T3, T2)	Test #2
(T1, T3)	Test #2	(T2, T3)	Test #1	(T3, T3)	Test #2
(T1, T4)	Test #3	(T2, T4)	Test #2	(T3, T4)	Test #1
(T1, T5)	Test #4	(T2, T5)	Test #3	(T3, T5)	Test #2
(T1, T6)	X executable	(T2, T6)	Test #3	(T3, T6)	Test #1
(T1, T7)	X executable	(T2, T7)	Test #3	(T3, T7)	Test #1
(T4, T2)	Test #3	(T6, T2)	Test #3	(T8, T2)	Test #5
(T4, T3)	Test #5	(T6, T3)	Test #7	(T8, T3)	Test #8
(T4, T4)	Test #2	(T6, T4)	Test #7	(T8, T4)	Test #8
(T4, T5)	Test #6	(T6, T5)	Test #7	(T8, T5)	Test #8
(T4, T6)	Test #1	(T6, T6)	Test #7	(T8, T6)	Test #8
(T4, T7)	Test #5	(T6, T7)	Test #1	(T8, T7)	Test #8
(T9, T2)	Test #5	(T10, T2)	Test #10	(T11, T2)	Test #11
(T9, T3)	Test #5	(T10, T3)	Test #10	(T11, T3)	Test #2
(T9, T4)	Test #9	(T10, T4)	Test #10	(T11, T4)	Test #11
(T9, T5)	Test #9	(T10, T5)	Test #1	(T11, T5)	Test #11
(T9, T6)	Test #9	(T10, T6)	Test #2	(T11, T6)	Test #11
(T9, T7)	Test #9	(T10, T7)	Test #10	(T11, T7)	Test #11
(T12, T2)	Test #12	(T13, T2)	Test #3	(T14, T2)	Test #13
(T12, T3)	Test #12	(T13, T3)	Test #15	(T14, T3)	Test #13
(T12, T4)	Test #12	(T13, T4)	Test #15	(T14, T4)	Test #13
(T12, T5)	Test #12	(T13, T5)	Test #15	(T14, T5)	Test #13
(T12, T6)	Test #12	(T13, T6)	Test #15	(T14, T6)	Test #13
(T12, T7)	Test #12	(T13, T7)	Test #15	(T14, T7)	Test #13
(T15, T2)	Test #14				
(T15, T3)	Test #14				
(T15, T4)	Test #14				
(T15, T5)	Test #1				
(T15, T6)	Test #14				
(T15, T7)	Test #14				

coins inserted

In: T7, T19, T20, T21, T23

Out: T10, T11, T12, T19, T20, T21, T22, T24, T25

Transaction pairs	Test Case	Transaction pairs	Test Case	Transaction pairs	Test Case
(T7, T10)	Test #2	(T19, T10)	Test #16	(T20, T10)	Test #17
(T7, T11)	X executable	(T19, T11)	X executable	(T20, T11)	Test #17
(T7, T12)	X executable	(T19, T12)	Test #16	(T20, T12)	Test #17
(T7, T19)	Test #1	(T19, T19)	Test #16	(T20, T19)	Test #5
(T7, T20)	Test #5	(T19, T20)	Test #16	(T20, T20)	Test #5
(T7, T21)	Test #2	(T19, T21)	Test #1	(T20, T21)	Test #5
(T7, T22)	Test #3	(T19, T22)	Test #16	(T20, T22)	Test #17
(T7, T24)	X executable	(T19, T24)	Test#5	(T20, T24)	Test #17
(T7, T25)	X executable	(T19, T25)	X executable	(T20, T25)	Test #17
(T21, T10)	Test #18	(T23, T10)	Test #19		
(T21, T11)	Test #2	(T23, T11)	Test #19		
(T21, T12)	X executable	(T23, T12)	Test #19		
(T21, T19)	Test #16	(T23, T19)	Test #16		
(T21, T20)	Test #17	(T23, T20)	Test #19		
(T21, T21)	Test #18	(T23, T21)	Test #17		
(T21, T22)	Test #1	(T23, T22)	Test #14		
(T21, T24)	X executable	(T23, T24)	Test #19		
(T21, T25)	Test #5	(T23, T25)	Test #19		

sugar

In: T16, T17, T18, T22

Out: T13, T14, T15, T16, T17, T18, T23, T26, T27

Transaction pairs	Test Case	Transaction pairs	Test Case	Transaction pairs	Test Case
(T16, T13)	Test #20	(T17, T13)	X executable	(T18, T13)	Test #1
(T16, T14)	Test #20	(T17, T14)	Test #21	(T18, T14)	Test #22
(T16, T15)	Test #20	(T17, T15)	Test #14	(T18, T15)	X executable
(T16, T16)	Test #20	(T17, T16)	Test #20	(T18, T16)	Test #20
(T16, T17)	Test #20	(T17, T17)	Test #3	(T18, T17)	Test #3
(T16, T18)	Test #20	(T17, T18)	Test #1	(T18, T18)	Test #3
(T16, T23)	Test #14	(T17, T23)	Test #19	(T18, T23)	Test #19
(T16, T26)	Test #20	(T17, T26)	X executable	(T18, T26)	Test #22
(T16, T27)	Test #20	(T17, T27)	Test #21	(T18, T27)	X executable
(T22, T13)	Test #23				
(T22, T14)	Test #20				
(T22, T15)	Test #23				
(T22, T16)	Test #14				
(T22, T17)	Test #1				
(T22, T18)	Test #14				
(T22, T23)	Test #3				
(T22, T26)	Test #22				
(T22, T27)	Test #23				

no large_cups

In: T24, T26, T29

Out: T8, T29

Transaction pairs	Test Case	Transaction pairs	Test Case	Transaction pairs	Test Case
(T24, T8)	Test #8	(T26, T8)	Test #22	(T29, T8)	Test #5
(T24, T29)	Test #5	(T26, T29)	Test #20	(T29, T29)	Test #5

no small cups

In: T25, T27, T28

Out: T9, T28

Transaction pairs	Test Case	Transaction pairs	Test Case	Transaction pairs	Test Case
(T25, T9)	Test #5	(T27, T9)	Test #21	(T28, T9)	Test #5
(T25, T28)	Test #5	(T27, T28)	Test #20	(T28, T28)	Test #5

T1->T6 cannot be executable. When this vending machine started, the price is set to 0. However, T6 is executable when the price is at least bigger than t+25.

T1->T7 cannot be executable. When this vending machine started, the price is set to 0. However, to execute T7 right after T1, price should be bigger than 0.

T7->T11, T7->T12, T7->T24 and T7->T25 cannot be executable. T11, T12, T24, and T25 are executed only when we set 's' with 1 or 2.

T21->T12 cannot be executable since T21 sets cup size to small but T12 gives a tea of large cup.

T21->T24 cannot be executed since T21 sets cup size to small but T24 is disposing large cup of tea.

T17->T13 cannot be executed since T17 sets cup size to small but T13 is disposing large cup of tea.

T17->T26 cannot be executable since T17 sets cup size to large

T18->T27 is also not executable for above reason (T17->T26)

T18->T15 is not executable since T18 sets cup size to large but T15 is to offer small cup

T19->11 cannot be executed since T19 sets cup size to large but T11 is disposing small cup of tea.

T19->25 is same as right above.

Test #1: vending_machine(), insert_large_cups(10), insert_small_cups(10), set_price(45), coin(), coin(), large_cup(), small_cup(), sugar(), small_cup(), large_cup(), tea(), insert_small_cups(1), coin(), insert_small_cups(1), coin(), cancel(), dispose()

T1, T2, T3, T4, T6, T7, T19, T21, T22, T17, T18, T13, T3, T6, T3, T7, T10, T5

Test #2: vending_machine(), insert_small_cups(10), insert_small_cups(10), insert_large_cups(10), insert_large_cups(10), set_price(45), set_price(46), coin(), coin(), cancel(), coin(), coin(), small_cup(), tea(), insert_small_cups(10), dispose()

T1, T3, T3, T2, T2, T4, T4, T6, T7, T10, T6, T7, T21, T11, T3, T5

Test #3: vending_machine(), set_price(45), insert_large_cups(10), coin(), insert_large_cups(5), coin(), sugar(), sugar(), sugar(), small_cup(), small_cup(), large_cup(), large_cup(), small_cup(), large_cup(), tea(), insert_large_cups(1), dispose(),

T1, T4, T2, T6, T2, T7, T22, T23, T22, T17, T17, T18, T18, T17, T18, T13, T2, T5

Test #4: vending_machine(), dispose()

T1, T5

Test #5: vending_machine(), set_price(45), insert_small_cups(1), set_price(20), coin(), large_cup(), coin(), coin(), small_cup(), tea(), coin(), coin(), insert_small_cups(1), insert_large_cups(1), coin(), small_cup(), tea(), insert small_cups(1), insert_small_cups(1), coin(), large_cup(), tea(), coin(), coin(), insert_large_cups(1), insert_large_cups(1), dispose()

T1, T4, T3, T4, T7, T19, T20, T20, T21, T25, T28, T28, T9, T2, T7, T21, T25, T9, T3, T7, T19, T24, T29, T29, T8, T2, T5

Test #6: vending_machine(), set_price(10), dispose()

T1, T4, T5

Test #7: vending_machine(), set_price(1000), coin(), coin(), insert_small_cups(10), coin(), set_price(500), coin(), dispose()

T1, T4, T6, T6, T3, T6, T4, T6, T5

Test #8: vending_machine(), set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), insert_large_cups(1), insert_small_cups(1), coin(), large_cup(), tea(), insert_large_cups(1), set_price(15), coin(), large_cup(), tea(), insert_large_cups(1), set_price(40), coin(), coin(), large_cup(), tea(), insert_large_cups(1), coin(), set_price(10), coin(), large_cup(), tea(), insert_large_cups(1), coin(), large_cup(), tea(), insert_large_cups(1), dispose()

T1, T4, T2, T7, T19, T24, T8, T3, T7, T19, T24, T8, T4, T7, T19, T24, T8, T4, T6, T7, T19, T24, T8, T6, T4, T7, T19, T24, T8, T7, T19, T24, T8, T5

Test #9: vending_machine(), set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), insert_small_cups(1), set_price(45), coin(), coin(), small_cup(), tea(), insert_small_cups(1), coin(), set_price(20), coin(), small_cup(), tea(), insert_small_cups(1), coin(), small_cup(), tea, insert_small_cups(1), dispose()

T1, T4, T3, T7, T21, T25, T9, T4, T6, T7, T21, T25, T9, T6, T4, T7, T21, T25, T9, T7, T21, T25, T9, T5

Test #10: vending_machine(), set_price(10), coin(), cancel(), coin(), cancel(), insert_large_cups(1), coin(), cancel(), insert_small_cups(1), coin(), cancel(), set_price(15), coin(), coin(), cancel(), set_price(45), coin(), coin(), cancel(), coin(), coin(), cancel(), dispose()

T1, T4, T7, T10, T7, T10, T2, T7, T10, T3, T7, T10, T4, T6, T7, T10, T4, T6, T7, T10, T6, T7, T10, T5

Test #11: vending_machine(), set_price(20), insert_small_cups(100), coin(), small_cup(), tea(), insert_large_cups(10), coin(), small_cup(), tea(), insert_small_cups(1), coin(), small_cup(), tea(), set_price(30), coin(), coin(), small_cup(), tea(), coin(), set_price(20), coin(), small_cup(), tea(), coin(), small_cup(), tea(), dispose()

T1, T4, T3, T7, T21, T11, T2, T7, T21, T11, T3, T7, T21, T11, T4, T6, T7, T21, T11, T6, T4, T7, T21, T11, T7, T21, T11, T5

Test #12: vending_machine(), set_price(20), insert_large_cups(100), coin(), large_cup(), tea(), insert_large_cups(10), coin(), large_cup(), tea(), insert_small_cups(1), coin(), large_cup(), tea(), set_price(30), coin(), coin(), large_cup(), tea(), coin(), set_price(20), coin(), large_cup(), tea(), coin(), large_cup(), tea, dispose()

T1, T4, T3, T7, T19, T12, T2, T7, T19, T12, T3, T7, T19, T12, T4, T6, T7, T19, T12, T6, T4, T7, T19, T12, T7, T19, T12, T5

Test #13: vending_machine(), set_price(20), insert_small_cups(1), coin(), sugar(), cancel(), insert_large_cups(), coin(), sugar(), cancel(), insert_small_cups(1), coin(), sugar(), cancel(), set_price(40), coin(), coin(), sugar(), cancel(), coin(), set_price(20), coin(), sugar(), cancel(), dispose()
T1, T4, T3, T7, T22, T14, T2, T7, T22, T14, T3, T7, T22, T14, T4, T6, T7, T22, T14, T6, T4, T7, T22, T14, T7, T22, T14, T5

Test #14: vending_machine(), set_price(20), insert_small_cups(100), coin(), sugar(), coin(), sugar(), sugar(), large_cup(), small_cup(), tea(), insert_large_cups(1), coin(), sugar(), small_cup(), tea(), insert_small_cups(1), coin(), sugar(), small_cup(), tea(), set_price(45), coin(), coin(), sugar(), small_cup(), tea(), coin(), set_price(20), coin(), sugar(), small_cup(), tea(), coin(), sugar(), small_cup(), tea(), dispose
T1, T4, T3, T7, T22, T16, T23, T22, T18, T17, T15, T2, T7, T22, T17, T15, T3, T7, T22, T17, T15, T4, T6, T7, T22, T17, T15, T6, T4, T7, T22, T17, T15, T7, T22, T17, T15, T5

Test #15: vending_machine(), set_price(20), insert_large_cups(100), coin(), sugar(), large_cup(), tea(), insert_large_cups(1), coin(), sugar(), large_cup(), tea(), insert_small_cups(1), coin(), sugar(), large_cup(), tea(), set_price(40), coin(), coin(), sugar(), large_cup(), tea(), coin(), set_price(20), coin(), sugar(), large_cup(), tea(), coin(), sugar(), large_cup(), tea(), dispose()
T1, T4, T2, T7, T22, 18, T13, T2, T7, T22, T18, T13, T3, T7, T22, T18, T13, T4, T6, T7, T22, T18, T13, T6, T4, T7, T22, T18, T13, T7, T22, T18, T13, T5

Test #16: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), large_cup(), cancel(), coin(), large_cup(), tea(), coin(), large_cup(), large_cup(), coin(), large_cup(), small_cup(), large_cup(), sugar(), sugar(), large_cup(), tea(), insert_large_cups(1), dispose()
T1, T4, T3, T2, T7, T19, T10, T7, T19, T12, T7, T19, T19, T20, T19, T21, T19, T22, T23, T19, T24, T8, T5

Test #17: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), coin(), cancel(), coin(), small_cup(), coin(), tea(), coin(), large_cup(), coin(), tea(), coin(), coin(), sugar(), sugar(), small_cup(), coin(), tea(), insert_small_cups(1), coin(), large_cup(), coin(), tea(), insert_large_cups(1), dispose()
T1, T4, T3, T2, T7, T20, T10, T7, T21, T20, T11, T7, T19, T20, T12, T7, T20, T22, T23, T21, T20, T25, T9, T7, T19, T20, T24, T8, T5

Test #18: vending_machine(), set_price(20), insert_small_cups(1), insert_large_cups(2), coin(), small_cup(), cancel(), coin(), small_cup(); small_cup(), tea(), insert_small_cup(1), dispose()
T1, T4, T3, T2, T7, T21, T10, T7, T21, T21, T25, T9, T5

Test #19: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), sugar(), sugar(), cancel(), coin(), sugar(), small_cup(), sugar(), tea(), coin(), sugar(), large_cup(), sugar, tea(), coin(), sugar(), sugar(), coin(), sugar(), small_cup(), sugar(), tea(), insert_small_cups(1), coin(), sugar(), large_cup(), sugar(), tea(), insert_large_cups(1), dispose()
T1, T4, T3, T2, T7, T22, T23, T10, T7, T22, T17, T23, T11, T7, T22, T18, T23, T12, T7, T22, T23, T20, T22, T17, T23, T25, T9, T7, T22, T18, T23, T24, T8, T5

Test #20: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), sugar(), large_cup(), coin(), tea(), coin(), sugar(), coin(), cancel(), coin(), sugar(), small_cup(), coin(), tea(), coin(), sugar(), coin(), coin(), small_cup(), coin(), large_cup(), coin(), tea(), coin(), insert_large_cups(1), coin(), sugar(), small_cup(), coin(), tea(), coin(), insert_small_cups(1), dispose()
T1, T4, T3, T2, T7, T22, T18, T16, T13, T7, T22, T16, T14, T7, T22, T17, T16, T15, T7, T22, T16, T16, T17, T16, T18, T16, T26, T29, T8, T7, T22, T17, T16, T27, 28, T9, T5

Test #21: vending_machine(), set_price(20), insert_small_cups(1), insert_large_cups(1), coin(), sugar(), small_cup(), cancel(), coin(), sugar(), small_cup(), tea(), insert_small_cups(1), dispose()
T1, T4, T3, T2, T7, T22, T17, T14, T7, T22, T17, T27, T9, T5

Test #22: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), sugar(), large_cup(), tea(), coin(), sugar(), large_cup(), cancel(), coin(), sugar(), large_cup(), tea(), insert_large_cups(1), coin(), large_cup(), sugar(), tea(), insert_large_cups(1), dispose
T1, T4, T3, T2, T7, T22, T18, T13, T7, T22, T18, T14, T7, T22, T18, T26, T8, T7, T19, T22, T26, T8, T5

Test #23: vending_machine(), set_price(20), insert_small_cups(2), insert_large_cups(2), coin(), small_cup(), sugar(), tea(), coin(), large_cup(), sugar(), tea(), coin(), small_cup(), sugar(), tea(), insert_small_cups(1), dispose()
T1, T4, T3, T2, T7, T21, T22, T15, T7, T19, T22, T13, T7, T21, T22, T27, T9, T5

2. Default (Ghost) Transition Testing

Idle

Ghost transitions	Test Case
coin()[0>price]	Test #24: set_price(-10), coin(), dispose()
small_cup()	Test #25: small_cup(), dispose()
large_cup()	Test #26: large_cup(), dispose()
sugar()	Test #27: sugar(), dispose()
tea()	Test #28: tea(), dispose()
insert_large_cups(n)[n<=0]	Test #29: insert_small_cups(0), dispose()
insert_small_cups(n)[n<=0]	Test #30: inset_small_cups(0), dispose()
set_price(p)[p<=0]	Test #31: set_price(-10), dispose()
cancel()	Test #32: cancel(), dispose()

no_small_cups

Ghost transitions	Test Case
insert_small_cups(n)[n<=0]	Test# 33
small_cup()	Test# 34
large_cup()	Test# 35
sugar()	Test# 36
tea()	Test# 37
insert_large_cups(n)	Test# 38
set_price(p)	Test# 39
cancel()	Test# 40
dispose()	Test# 41

Test# 33: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), insert_small_cups(-1) , insert_small_cups(1), dispose()
Test# 34: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), small_cup(), insert_small_cups(1), dispose
Test# 35: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), tea(), large_cup(), insert_small_cups(1), dispose
Test# 36: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), sugar(), insert_small_cups(1), dispose
Test# 37: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), tea(), insert_small_cups(1), dispose

Test# 38: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), insert_large_cups(1) , insert_small_cups(1), dispose

Test# 39: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), set_price(10) , insert_small_cups(1), dispose

Test# 40: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), cancel(), insert_small_cups(1), dispose

Test# 41: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), dispose(), insert_small_cups(1), dispose

no_large_cups

Ghost transitions	Test Case
insert_large_cups(n)[n<=0]	Test# 42
small_cup()	Test# 43
large_cup()	Test# 44
sugar()	Test# 45
tea()	Test# 46
insert_small_cups(n)	Test# 47
set_price(p)	Test# 48
cancel()	Test# 49
dispose()	Test# 50

Test# 42: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), insert_large_cups(-1), insert_large_cups(1), dispose

Test# 43: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), small_cup(), insert_large_cups(1), dispose

Test# 44: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), large_cup(), insert_large_cups(1), dispose

Test# 45: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), sugar(), insert_large_cups(1), dispose

Test# 46: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), tea(), insert_large_cups(1), dispose

Test# 47: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), insert_small_cups(1) , insert_large_cups(1), dispose

Test# 48: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), set_price(10) , insert_large_cups(1), dispose

Test# 49: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), cancel(), insert_large_cups(1), dispose

Test# 50: set_price(20), insert_large_cups(1), coin(), large_cup(), tea(), dispose(), insert_large_cups(1), dispose()

coin_inserted

Ghost transitions	Test Case
tea()[k<=0; k1<=0; s!=1; s!= 2]	Test #51
insert_large_cups(n)	Test #52
insert_small_cups(n)	Test #53
set_price(10)	Test #54
dispose	Test #55

Test #51: set_price(10), coin(), tea(), cancel(), dispose()

Test #52: set_price(10), coin(), insert_large_cups(1) , cancel(), dispose()

Test #53: set_price(10), coin(), insert_small_cups(1) , cancel(), dispose()

Test #54: set_price(10), coin(), set_price(10) , cancel(), dispose()

Test #55: set_price(10), coin(), dispose(), cancel(), dispose()

sugar

Ghost transitions	Test Case
tea()k<=0; k1<=0; s!=1; s!= 2]	Test# 56
insert_large_cups(n)	Test# 57
insert_small_cups(n)	Test# 58
set_price(10)	Test# 59
dispose()	Test# 60

Test #56: set_price(10), coin(), sugar(), tea(), cancel), cancel(), dispose()
Test #57: set_price(10), coin(), sugar(), insert_large_cups(1), cancel(), dispose()
Test #58: set_price(10), coin(), sugar(), insert_small_cups(1), cancel(), dispose()
Test #59: set_price(10), coin(), sugar(), set_price(10), cancel(), dispose()
Test #60: set_price(10), coin(), sugar(), dispose(), cancel(), dispose()

3. Multiple-condition testing

coin()

	x == 1	Test Cases
1	T	Test# 1
2	F	Test# 20

	t + 25 >= price	price > 0	Test Cases
1	T	T	Test# 1
2	T	F	X executable
3	F	T	Test# 1
4	F	F	X executable

	t + 25 < price	Test Cases
1	T	Test# 5
2	F	Test# 1

	x > 1	x < 6	Test Cases
1	T	T	Test# 14
2	T	F	Test# 14
3	F	T	N/A
4	F	F	N/A

small_cup()

	x == 2	x == 3	Test Cases
1	T	T	X possible
2	T	F	Test# 2
3	F	T	Test# 14
4	F	F	Test# 34

large_cup()

	x == 2	x == 3	Test Cases
1	T	T	X possible

2	T	F	Test# 12
3	F	T	Test# 15
4	F	F	Test# 35

sugar()

	x == 2	x == 3	Test Cases
1	T	T	X possible
2	T	F	Test# 3
3	F	T	Test# 14
4	F	F	Test# 36

	x == 2	Test Cases
1	T	Test# 3
2	F	Test# 14

tea()

	x == 2	x == 3	Test Cases
1	T	T	X possible
2	T	F	Test# 17
3	F	T	Test# 22
4	F	F	Test# 37

	x == 2	k1 > 1	s == 2	Test Cases
1	T	T	T	Test# 2
2	T	T	F	Test# 12
3	T	F	T	executable
4	T	F	F	Test# 12
5	F	T	T	Test# 14
6	F	T	F	Test# 22
7	F	F	T	X executable
8	F	F	F	Test# 15

	x == 2	k > 1	s == 1	Test Cases
1	T	T	T	Test# 12
2	T	T	F	Test# 11
3	T	F	T	X executable
4	T	F	F	Test# 11
5	F	T	T	Test# 15
6	F	T	F	Test# 19
7	F	F	T	X executable
8	F	F	F	Test# 14

	x == 2	k == 1	s == 1	Test Cases
1	T	T	T	Test# 5
2	T	T	F	X executable
3	T	F	T	Test# 12

4	T	F	F	Test# 2
5	F	T	T	X executable
6	F	T	F	Test# 14
7	F	F	T	Test# 15
8	F	F	F	Test# 61

Test# 61: set_price(20), insert_large_cups(10), insert_small_cups(10), coin(), sugar(), small_cup(), tea(), dispose()

	x == 2	k1 == 1	s == 2	Test Cases
1	T	T	T	Test# 5
2	T	T	F	Test# 8
3	T	F	T	Test# 11
4	T	F	F	Test# 16
5	F	T	T	Test #21
6	F	T	F	Test# 20
7	F	F	T	X executable
8	F	F	F	Test# 62

Test# 62: set_price(20), insert_small_cups(10), insert_large_cups(10), coin(), sugar(), large_cup(), tea(), dispose()

	x == 3	k1 == 1	s == 2	Test Cases
1	T	T	T	Test# 21
2	T	T	F	Test# 23
3	T	F	T	Test# 14
4	T	F	F	Test# 1
5	F	T	T	X executable
6	F	T	F	Test# 8
7	F	F	T	Test# 11
8	F	F	F	Test# 16

	x == 3	k == 1	s == 1	Test Cases
1	T	T	T	Test# 22
2	T	T	F	Test# 14
3	T	F	T	Test# 15
4	T	F	F	Test# 61
5	F	T	T	X executable
6	F	T	F	Test# 17
7	F	F	T	Test# 12
8	F	F	F	Test# 2

---

	x == 3	k1 > 1	s == 2	Test Cases
1	T	T	T	Test# 14
2	T	T	F	Test# 1
3	T	F	T	X executable
4	T	F	F	Test# 3
5	F	T	T	X executable

6	F	T	F	Test# 16
7	F	F	T	X executable
8	F	F	F	Test# 12

	x == 3	k > 1	s == 1	Test Cases
1	T	T	T	Test# 1
2	T	T	F	Test# 63
3	T	F	T	X executable
4	T	F	F	Test# 64
5	F	T	T	X executable
6	F	T	F	Test# 11
7	F	F	T	X executable
8	F	F	F	Test# 11

Test# 63: set_price(20), insert_large_cups(10), insert_small_cups(10), coin(), sugar(), small_cup(), tea(), dispose()

Test# 64: set_price(20), insert_small_cups(10), coin(), sugar(), small_cup(), tea(), dispose()

insert_large_cups(n)

	x == 1	n > 0	Test Cases
1	T	T	Test# 1
2	T	F	Test# 29
3	F	T	Test# 38
4	F	F	Test# 42

	x == 5	n > 0	Test Cases
1	T	T	Test# 17
2	T	F	Test# 42
3	F	T	Test #1
4	F	F	Test #65

Test# 65: set_price(20), insert_small_cups(1), coin(), small_cup(), tea(), insert_large_cups(-1), insert_small_cups(1), dispose()

insert_small_cups(n)

	x == 1	n > 0	Test Cases
1	T	T	Test# 1
2	T	F	Test# 30
3	F	T	Test# 17
4	F	F	Test# 33

	x == 4	n > 0	Test Cases
1	T	T	Test# 20
2	T	F	Test# 34
3	F	T	Test# 1
4	F	F	New Test

set_price()

	x == 1	p>0	Test Cases
1	T	T	Test# 1
2	T	F	Test# 31
3	F	T	Test# 39
4	F	F	new testcase

cancel()

	x == 2	x == 3	Test Cases
1	T	T	X possible
2	T	F	Test# 10
3	F	T	Test# 13
4	F	F	Test# 32

dispose()

	x == 1	Test Cases
1	T	Test# 1
2	F	Test# 51

4. Test Suit and the reulst of execution

TS.txt

```
Test#1: insert_large_cups 10 insert_small_cups 10 set_price 45 coin coin large_cup small_cup sugar small_cup large_cup tea insert_small_cups 1 coin insert_small_cups 1 coin cancel dispose
Test#2: insert_small_cups 10 insert_small_cups 10 insert_large_cups 10 insert_large_cups 10 set_price 45 set_price 46 coin coin cancel coin coin small_cup tea insert_small_cups 10 dispose
Test#3: set_price 45 insert_large_cups 10 coin insert_large_cups 5 coin sugar sugar sugar small_cup small_cup large_cup large_cup small_cup large_cup tea insert_large_cups 1 dispose
Test#4: dispose
Test#5: set_price 45 insert_small_cups 1 set_price 20 coin large_cup coin coin small_cup tea coin coin insert_small_cups 1 insert_large_cups 1 coin small_cup tea insert_small_cups 1 insert_small_cups 1 coin large_cup tea coin coin insert_large_cups 1 insert_large_cups 1 dispose
Test#6: set_price 10 dispose
Test#7: set_price 100 coin coin insert_small_cups 10 coin set_price 500 coin dispose
Test#8: set_price 20 insert_large_cups 1 coin large_cup tea insert_large_cups 1 insert_small_cups 1 coin large_cup tea insert_large_cups 1 set_price 15 coin large_cup tea insert_large_cups 1 set_price 40 coin coin large_cup tea insert_large_cups 1 coin set_price 10 coin large_cup tea insert_large_cups 1 coin large_cup tea insert_large_cups 1 dispose
Test#9: set_price 20 insert_small_cups 1 coin small_cup tea insert_small_cups 1 set_price 45 coin coin small_cup tea insert_small_cups 1 coin set_price 20 coin small_cup tea insert_small_cups 1 coin small_cup tea insert_small_cups 1 dispose
Test#10: set_price 10 coin cancel coin cancel insert_large_cups 1 coin cancel insert_small_cups 1 coin cancel set_price 15 coin coin cancel set_price 45 coin coin cancel coin coin cancel dispose
Test#11: set_price 20 insert_small_cups 100 coin small_cup tea insert_large_cups 10 coin small_cup tea insert_small_cups 1 coin small_cup tea set_price 30 coin coin small_cup tea coin set_price 20 coin small_cup tea coin small_cup tea dispose
Test#12: set_price 20 insert_large_cups 100 coin large_cup tea insert_large_cups 10 coin large_cup tea insert_small_cups 1 coin large_cup tea set_price 30 coin coin large_cup tea coin set_price 20 coin large_cup tea coin large_cup tea dispose
Test#13: set_price 20 insert_small_cups 1 coin sugar cancel insert_large_cups 10 coin sugar cancel insert_small_cups 1 coin sugar cancel set_price 40 coin coin sugar cancel coin set_price 20 coin sugar cancel dispose
Test#14: set_price 20 insert_small_cups 100 coin sugar coin sugar sugar large_cup small_cup tea insert_large_cups 1 coin sugar small_cup tea insert_small_cups 1 coin sugar small_cup tea set_price 45 coin coin sugar small_cup tea coin set_price 20 coin sugar small_cup tea coin sugar small_cup tea dispose
Test#15: set_price 20 insert_large_cups 100 coin sugar large_cup tea insert_large_cups 1 coin sugar large_cup tea insert_small_cups 1 coin sugar large_cup tea set_price 40 coin coin sugar large_cup tea coin set_price 20 coin sugar large_cup tea coin sugar large_cup tea dispose
Test#16: set_price 20 insert_small_cups 2 insert_large_cups 2 coin large_cup cancel coin large_cup tea coin large_cup large_cup coin large_cup small_cup large_cup sugar sugar large_cup tea insert_large_cups 1 dispose
Test#17: set_price 20 insert_small_cups 2 insert_large_cups 2 coin coin cancel coin small_cup coin tea coin large_cup coin tea coin coin sugar sugar small_cup coin tea insert_small_cups 1 coin large_cup coin tea insert_large_cups 1 dispose
Test#18: set_price 20 insert_small_cups 1 insert_large_cups 2 coin small_cup cancel coin small_cup small_cup tea insert_small_cups 1 dispose
```

Test#19: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar sugar cancel coin sugar small_cup sugar tea coin sugar large_cup sugar tea coin sugar sugar coin sugar small_cup sugar tea insert_small_cups 1 coin sugar large_cup sugar tea insert_large_cups 1 dispose

Test#20: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar large_cup coin tea coin sugar coin cancel coin sugar small_cup coin tea coin sugar coin coin small_cup coin large_cup coin tea coin insert_large_cups 1 coin sugar small_cup coin tea coin insert_small_cups 1 dispose

Test#21: set_price 20 insert_small_cups 1 insert_large_cups 1 coin sugar small_cup cancel coin sugar small_cup tea insert_small_cups 1 dispose

Test#22: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar large_cup tea coin sugar large_cup cancel coin sugar large_cup tea insert_large_cups 1 coin large_cup sugar tea insert_large_cups 1 dispose

Test#23: set_price 20 insert_small_cups 2 insert_large_cups 2 coin small_cup sugar tea coin large_cup sugar tea coin small_cup sugar tea insert_small_cups 1 dispose

Test#24: set_price -10 coin dispose

Test#25: small_cup dispose

Test#26: large_cup dispose

Test#27: sugar dispose

Test#28: tea dispose

Test#29: insert_small_cups 0 dispose

Test#30: insert_small_cups 0 dispose

Test#31: set_price -10 dispose

Test#32: cancel dispose

Test#33: set_price 20 insert_small_cups 1 coin small_cup tea insert_small_cups -1 insert_small_cups 1 dispose

Test#34: set_price 20 insert_small_cups 1 coin small_cup tea small_cup insert_small_cups 1 dispose

Test#35: set_price 20 insert_small_cups 1 coin small_cup tea large_cup insert_small_cups 1 dispose

Test#36: set_price 20 insert_small_cups 1 coin small_cup tea sugar insert_small_cups 1 dispose

Test#37: set_price 20 insert_small_cups 1 coin small_cup tea tea insert_small_cups 1 dispose

Test#38: set_price 20 insert_small_cups 1 coin small_cup tea insert_large_cups 1 insert_small_cups 1 dispose

Test#39: set_price 20 insert_small_cups 1 coin small_cup tea set_price 10 insert_small_cups 1 dispose

Test#40: set_price 20 insert_small_cups 1 coin small_cup tea cancel insert_small_cups 1 dispose

Test#41: set_price 20 insert_small_cups 1 coin small_cup tea dispose insert_small_cups 1 dispose

Test#42: set_price 20 insert_large_cups 1 coin large_cup tea insert_large_cups -1 insert_large_cups 1 dispose

Test#43: set_price 20 insert_large_cups 1 coin large_cup tea small_cup insert_large_cups 1 dispose

Test#44: set_price 20 insert_large_cups 1 coin large_cup tea large_cup insert_large_cups 1 dispose

Test#45: set_price 20 insert_large_cups 1 coin large_cup tea sugar insert_large_cups 1 dispose

Test#46: set_price 20 insert_large_cups 1 coin large_cup tea tea insert_large_cups 1 dispose

Test#47: set_price 20 insert_large_cups 1 coin large_cup tea insert_small_cups 1 insert_large_cups 1 dispose

Test#48: set_price 20 insert_large_cups 1 coin large_cup tea set_price 10 insert_large_cups 1 dispose

Test#49: set_price 20 insert_large_cups 1 coin large_cup tea cancel insert_large_cups 1 dispose

Test#50: set_price 20 insert_large_cups 1 coin large_cup tea dispose insert_large_cups 1 dispose

Test#51: set_price 10 coin tea cancel dispose

Test#52: set_price 10 coin insert_large_cups 1 cancel dispose

Test#53: set_price 10 coin insert_small_cups 1 cancel dispose

Test#54: set_price 10 coin set_price 10 cancel dispose

Test#55: set_price 10 coin dispose cancel dispose

Test#56: set_price 10 coin sugar tea cancel cancel dispose

Test#57: set_price 10 coin sugar insert_large_cups 1 cancel dispose

Test#58: set_price 10 coin sugar insert_small_cups 1 cancel dispose

Test#59: set_price 10 coin sugar set_price 10 cancel dispose

Test#60: set_price 10 coin sugar dispose cancel dispose

Test#61: set_price 20 insert_large_cups 10 insert_small_cups 10 coin sugar small_cup tea dispose

Test#62: set_price 20 insert_small_cups 10 insert_large_cups 10 coin sugar large_cup tea dispose

Test#63: set_price 20 insert_large_cups 10 insert_small_cups 10 coin sugar small_cup tea dispose

Test#64: set_price 20 insert_small_cups 10 coin sugar small_cup tea dispose

Test#65: set_price 20 insert_small_cups 1 coin small_cup tea insert_large_cups -1 insert_small_cups 1 dispose

$$ $$

Test#1: insert_large_cups 10 insert_small_cups 10 set_price 45 coin coin large_cup small_cup sugar small_cup large_cup tea insert_small_cups 1 coin insert_small_cups 1 coin cancel dispose

```
return value=1 state=Idle price=0 k=10 k1=0 t=0 s=0
return value=1 state=Idle price=0 k=10 k1=10 t=0 s=0
return value=1 state=Idle price=45 k=10 k1=10 t=0 s=0
return value=1 state=Idle price=45 k=10 k1=10 t=25 s=0
return value=1 state=Coin_Inserted price=45 k=10 k1=10 t=0 s=0
return value=1 state=Coin_Inserted price=45 k=10 k1=10 t=0 s=1
return value=1 state=Coin_Inserted price=45 k=10 k1=10 t=0 s=2
return value=1 state=Sugar price=45 k=10 k1=10 t=0 s=2
return value=1 state=Sugar price=45 k=10 k1=10 t=0 s=2
return value=1 state=Sugar price=45 k=10 k1=10 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=45 k=9 k1=10 t=0 s=1
return value=1 state=Idle price=45 k=9 k1=11 t=0 s=1
return value=1 state=Idle price=45 k=9 k1=11 t=25 s=1
return value=1 state=Idle price=45 k=9 k1=12 t=25 s=1
return value=1 state=Coin_Inserted price=45 k=9 k1=12 t=0 s=0
RETURN COINS
return value=1 state=Idle price=45 k=9 k1=12 t=0 s=0
SHUT DOWN
```

```
Actual result:          return value=1 state=exit price=45 k=9 k1=12 t=0 s=0
Expected result:        return value=1 state=exit price=45 k=9 k1=12 t=0 s=0        Test Passed
```

Test#2: insert_small_cups 10 insert_small_cups 10 insert_large_cups 10 insert_large_cups 10 set_price 45 set_price 46 coin coin cancel coin coin small_cup tea insert_small_cups 10 dispose

```
return value=1 state=Idle price=0 k=0 k1=10 t=0 s=0
return value=1 state=Idle price=0 k=0 k1=20 t=0 s=0
return value=1 state=Idle price=0 k=10 k1=20 t=0 s=0
return value=1 state=Idle price=0 k=20 k1=20 t=0 s=0
return value=1 state=Idle price=45 k=20 k1=20 t=0 s=0
return value=1 state=Idle price=46 k=20 k1=20 t=0 s=0
return value=1 state=Idle price=46 k=20 k1=20 t=25 s=0
return value=1 state=Coin_Inserted price=46 k=20 k1=20 t=0 s=0
RETURN COINS
return value=1 state=Idle price=46 k=20 k1=20 t=0 s=0
return value=1 state=Idle price=46 k=20 k1=20 t=25 s=0
return value=1 state=Coin_Inserted price=46 k=20 k1=20 t=0 s=0
return value=1 state=Coin_Inserted price=46 k=20 k1=20 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=46 k=20 k1=19 t=0 s=2
return value=1 state=Idle price=46 k=20 k1=29 t=0 s=2
SHUT DOWN
return value=1 state=exit price=46 k=20 k1=29 t=0 s=2
```

```
Actual result :         return value=1 state=exit price=46 k=20 k1=29 t=0 s=2
Expected result         return value=1 state=exit price=46 k=20 k1=29 t=0 s=2       Test Passed
```

Test#3: set_price 45 insert_large_cups 10 coin insert_large_cups 5 coin sugar sugar sugar small_cup small_cup large_cup large_cup small_cup large_cup tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=45 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=45 k=10 k1=0 t=0 s=0
return value=1 state=Idle price=45 k=10 k1=0 t=25 s=0
return value=1 state=Idle price=45 k=15 k1=0 t=25 s=0
return value=1 state=Coin_Inserted price=45 k=15 k1=0 t=0 s=0
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=45 k=15 k1=0 t=0 s=0
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=0
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=2
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=2
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=1
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=1
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=2
return value=1 state=Sugar price=45 k=15 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=45 k=14 k1=0 t=0 s=1
return value=1 state=Idle price=45 k=15 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=45 k=15 k1=0 t=0 s=1
```

```
Actual Result:      return value=1 state=exit price=45 k=15 k1=0 t=0 s=1
Expected Result:    return value=1 state=exit price=45 k=15 k1=0 t=0 s=1      Test Passed
```

Test#4: dispose

```
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
```

```
Actual Result:      return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected Result:    return value=1 state=exit price=0 k=0 k1=0 t=0 s=0        Test Passed
```

Test#5: set_price 45 insert_small_cups 1 set_price 20 coin large_cup coin coin small_cup tea coin coin insert_small_cups 1 insert_large_cups 1 coin small_cup tea insert_small_cups 1 insert_small_cups 1 coin large_cup tea coin coin insert_large_cups 1 insert_large_cups 1 dispose

```
return value=1 state=Idle price=45 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=45 k=0 k1=1 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=1
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=1
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
RETURN COIN
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
```

```
RETURN COIN
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
RETURN COIN
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
RETURN COIN
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=2 k1=2 t=0 s=1
```

Actual result: return value=1 state=exit price=20 k=2 k1=2 t=0 s=1
Expected result: return value=1 state=exit price=20 k=2 k1=2 t=0 s=1 Test Passed

Test#6: set_price 10 dispose

```
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
```

Actual result: return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
Expected result: return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0   Test passed

Test#7: set_price 100 coin coin insert_small_cups 10 coin set_price 500 coin dispose

```
return value=1 state=Idle price=100 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=100 k=0 k1=0 t=25 s=0
return value=1 state=Idle price=100 k=0 k1=0 t=50 s=0
return value=1 state=Idle price=100 k=0 k1=10 t=50 s=0
return value=1 state=Idle price=100 k=0 k1=10 t=75 s=0
return value=1 state=Idle price=500 k=0 k1=10 t=75 s=0
return value=1 state=Idle price=500 k=0 k1=10 t=100 s=0
SHUT DOWN
return value=1 state=exit price=500 k=0 k1=10 t=100 s=0
```

Actual result: return value=1 state=exit price=500 k=0 k1=10 t=100 s=0
Expected result: return value=1 state=exit price=500 k=0 k1=10 t=100 s=0

Test#8: set_price 20 insert_large_cups 1 coin large_cup tea insert_large_cups 1 insert_small_cups 1 coin large_cup tea insert_large_cups 1 set_price 15 coin large_cup tea insert_large_cups 1 set_price 40 coin coin large_cup tea insert_large_cups 1 coin set_price 10 coin large_cup tea insert_large_cups 1 coin large_cup tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Idle price=15 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=15 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=15 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=15 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=15 k=1 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=1 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=1 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=40 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=40 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=40 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=1 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=1 k1=1 t=25 s=1
return value=1 state=Idle price=10 k=1 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=10 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=10 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=10 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=10 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=10 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=10 k=1 k1=1 t=0 s=1
SHUT DOWN
return value=1 state=exit price=10 k=1 k1=1 t=0 s=1
```

```
Actual result:       return value=1 state=exit price=10 k=1 k1=1 t=0 s=1
Expected result :    return value=1 state=exit price=10 k=1 k1=1 t=0 s=1      Test passed
```

Test#9: set_price 20 insert_small_cups 1 coin small_cup tea insert_small_cups 1 set_price 45 coin coin small_cup tea insert_small_cups 1 coin set_price 20 coin small_cup tea insert_small_cups 1 coin small_cup tea insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
return value=1 state=Idle price=45 k=0 k1=1 t=0 s=2
return value=1 state=Idle price=45 k=0 k1=1 t=25 s=2
return value=1 state=Coin_Inserted price=45 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=45 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=45 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=45 k=0 k1=1 t=0 s=2
return value=1 state=Idle price=45 k=0 k1=1 t=25 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=25 s=2
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
```

```
Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2          Test Passed
```

Test#10: set_price 10 coin cancel coin cancel insert_large_cups 1 coin cancel insert_small_cups 1 coin cancel set_price 15 coin coin cancel set_price 45 coin coin cancel coin coin cancel dispose

```
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=10 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=1 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=1 k1=0 t=0 s=0
return value=1 state=Idle price=10 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=1 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=1 k1=1 t=0 s=0
return value=1 state=Idle price=15 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=15 k=1 k1=1 t=0 s=0
RETURN COIN
return value=1 state=Coin_Inserted price=15 k=1 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=15 k=1 k1=1 t=0 s=0
return value=1 state=Idle price=45 k=1 k1=1 t=0 s=0
return value=1 state=Idle price=45 k=1 k1=1 t=25 s=0
return value=1 state=Coin_Inserted price=45 k=1 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=45 k=1 k1=1 t=0 s=0
return value=1 state=Idle price=45 k=1 k1=1 t=25 s=0
return value=1 state=Coin_Inserted price=45 k=1 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=45 k=1 k1=1 t=0 s=0
SHUT DOWN
return value=1 state=exit price=45 k=1 k1=1 t=0 s=0
```

```
Actual result:          return value=1 state=exit price=45 k=1 k1=1 t=0 s=0
Expected result:        return value=1 state=exit price=45 k=1 k1=1 t=0 s=0          Test Passed
```

Test#11: set_price 20 insert_small_cups 100 coin small_cup tea insert_large_cups 10 coin small_cup tea insert_small_cups 1 coin small_cup tea set_price 30 coin coin small_cup tea coin set_price 20 coin small_cup tea coin small_cup tea dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=100 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=100 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=100 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=0 k1=99 t=0 s=2
return value=1 state=Idle price=20 k=10 k1=99 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=10 k1=99 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=99 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=10 k1=98 t=0 s=2
return value=1 state=Idle price=20 k=10 k1=99 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=10 k1=99 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=99 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=10 k1=98 t=0 s=2
return value=1 state=Idle price=30 k=10 k1=98 t=0 s=2
return value=1 state=Idle price=30 k=10 k1=98 t=25 s=2
return value=1 state=Coin_Inserted price=30 k=10 k1=98 t=0 s=0
return value=1 state=Coin_Inserted price=30 k=10 k1=98 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=30 k=10 k1=97 t=0 s=2
return value=1 state=Idle price=30 k=10 k1=97 t=25 s=2
return value=1 state=Idle price=20 k=10 k1=97 t=25 s=2
return value=1 state=Coin_Inserted price=20 k=10 k1=97 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=97 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=10 k1=96 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=10 k1=96 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=96 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=10 k1=95 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=10 k1=95 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=10 k1=95 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=10 k1=95 t=0 s=2      Test passed
```

Test#12: set_price 20 insert_large_cups 100 coin large_cup tea insert_large_cups 10 coin large_cup tea insert_small_cups 1 coin large_cup tea set_price 30 coin coin large_cup tea coin set_price 20 coin large_cup tea coin large_cup tea dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=100 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=100 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=100 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=99 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=109 k1=0 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=109 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=109 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=108 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=108 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=108 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=108 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=107 k1=1 t=0 s=1
return value=1 state=Idle price=30 k=107 k1=1 t=0 s=1
return value=1 state=Idle price=30 k=107 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=30 k=107 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=30 k=107 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=30 k=106 k1=1 t=0 s=1
return value=1 state=Idle price=30 k=106 k1=1 t=25 s=1
return value=1 state=Idle price=20 k=106 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=20 k=106 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=106 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=105 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=105 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=105 k1=1 t=0 s=1
```

```
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=104 k1=1 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=104 k1=1 t=0 s=1

Actual result:      return value=1 state=exit price=20 k=104 k1=1 t=0 s=1
Expected result:    return value=1 state=exit price=20 k=104 k1=1 t=0 s=1     Test passed
```

Test#13: set_price 20 insert_small_cups 1 coin sugar cancel insert_large_cups 10 coin sugar cancel insert_small_cups 1 coin sugar cancel set_price 40 coin coin sugar cancel coin set_price 20 coin sugar cancel dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=1 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=10 k1=1 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=10 k1=2 t=0 s=0
return value=1 state=Idle price=40 k=10 k1=2 t=0 s=0
return value=1 state=Idle price=40 k=10 k1=2 t=25 s=0
return value=1 state=Coin_Inserted price=40 k=10 k1=2 t=0 s=0
return value=1 state=Sugar price=40 k=10 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=40 k=10 k1=2 t=0 s=0
return value=1 state=Idle price=40 k=10 k1=2 t=25 s=0
return value=1 state=Idle price=20 k=10 k1=2 t=25 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=10 k1=2 t=0 s=0
SHUT DOWN
return value=1 state=exit price=20 k=10 k1=2 t=0 s=0

Actual result:      return value=1 state=exit price=20 k=10 k1=2 t=0 s=0
Expected result:    return value=1 state=exit price=20 k=10 k1=2 t=0 s=0      Test Passed
```

Test#14: set_price 20 insert_small_cups 100 coin sugar coin sugar sugar large_cup small_cup tea insert_large_cups 1 coin sugar small_cup tea insert_small_cups 1 coin sugar small_cup tea set_price 45 coin coin sugar small_cup tea coin set_price 20 coin sugar small_cup tea coin sugar small_cup tea dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=100 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=100 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=100 t=0 s=0
RETURN COIN
return value=1 state=Sugar price=20 k=0 k1=100 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=100 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=100 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=100 t=0 s=1
return value=1 state=Sugar price=20 k=0 k1=100 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=0 k1=99 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=99 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=99 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=99 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=99 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=98 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=99 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=99 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=99 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=99 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=98 t=0 s=2
return value=1 state=Idle price=45 k=1 k1=98 t=0 s=2
return value=1 state=Idle price=45 k=1 k1=98 t=25 s=2
return value=1 state=Coin_Inserted price=45 k=1 k1=98 t=0 s=0
return value=1 state=Sugar price=45 k=1 k1=98 t=0 s=0
return value=1 state=Sugar price=45 k=1 k1=98 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=45 k=1 k1=97 t=0 s=2
return value=1 state=Idle price=45 k=1 k1=97 t=25 s=2
return value=1 state=Idle price=20 k=1 k1=97 t=25 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=97 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=97 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=97 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=96 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=96 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=96 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=96 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=95 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=95 t=0 s=2
```

```
Actual result :          return value=1 state=exit price=20 k=1 k1=95 t=0 s=2
Expected result:         return value=1 state=exit price=20 k=1 k1=95 t=0 s=2       Test passed
```

Test#15: set_price 20 insert_large_cups 100 coin sugar large_cup tea insert_large_cups 1 coin sugar large_cup tea insert_small_cups 1 coin sugar large_cup tea set_price 40 coin coin sugar large_cup tea coin set_price 20 coin sugar large_cup tea coin sugar large_cup tea dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=100 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=100 k1=0 t=0 s=0
return value=1 state=Sugar price=20 k=100 k1=0 t=0 s=0
return value=1 state=Sugar price=20 k=100 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=99 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=100 k1=0 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=100 k1=0 t=0 s=0
return value=1 state=Sugar price=20 k=100 k1=0 t=0 s=0
return value=1 state=Sugar price=20 k=100 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=99 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=99 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=99 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=99 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=99 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=98 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=98 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=98 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=40 k=98 k1=1 t=0 s=0
return value=1 state=Sugar price=40 k=98 k1=1 t=0 s=0
return value=1 state=Sugar price=40 k=98 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=40 k=97 k1=1 t=0 s=1
return value=1 state=Idle price=40 k=97 k1=1 t=25 s=1
return value=1 state=Idle price=20 k=97 k1=1 t=25 s=1
return value=1 state=Coin_Inserted price=20 k=97 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=97 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=97 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=96 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=96 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=96 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=96 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=95 k1=1 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=95 k1=1 t=0 s=1
```

```
Actual result:           return value=1 state=exit price=20 k=95 k1=1 t=0 s=1
Expected result:         return value=1 state=exit price=20 k=95 k1=1 t=0 s=1       Test passed
```

Test#16: set_price 20 insert_small_cups 2 insert_large_cups 2 coin large_cup cancel coin large_cup tea coin large_cup large_cup coin large_cup small_cup large_cup sugar sugar large_cup tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=1
RETURN COINS
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=2 t=0 s=1
```

```
Actual result:       return value=1 state=exit price=20 k=1 k1=2 t=0 s=1
Expected result:     return value=1 state=exit price=20 k=1 k1=2 t=0 s=1       Test Passed
```

Test#17: set_price 20 insert_small_cups 2 insert_large_cups 2 coin coin cancel coin small_cup coin tea coin large_cup coin tea coin coin sugar sugar small_cup coin tea insert_small_cups 1 coin large_cup coin tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=2
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=2
```

```
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=1
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=2
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=1
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=1 t=0 s=1
```

```
Actual result:      return value=1 state=exit price=20 k=1 k1=1 t=0 s=1
Expected result:    return value=1 state=exit price=20 k=1 k1=1 t=0 s=1      Test Passed
```

Test#18: set_price 20 insert_small_cups 1 insert_large_cups 2 coin small_cup cancel coin small_cup small_cup tea insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=2
RETURN COINS
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=2 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=2 k1=1 t=0 s=2
```

```
Actual result:          return value=1 state=exit price=20 k=2 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=2 k1=1 t=0 s=2        Test Passed
```

Test#19: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar sugar cancel coin sugar small_cup sugar tea coin sugar large_cup sugar tea coin sugar sugar coin sugar small_cup sugar tea insert_small_cups 1 coin sugar large_cup sugar tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
RETURN COIN
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=1 t=0 s=1
```

```
Actual result:          return value=1 state=exit price=20 k=1 k1=1 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=1 t=0 s=1        Test Passed
```

Test#20: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar large_cup coin tea coin sugar coin cancel coin sugar small_cup coin tea coin sugar coin coin small_cup coin large_cup coin tea coin insert_large_cups 1 coin sugar small_cup coin tea coin insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=1
RETURN COIN
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=0
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=0
RETURN COINS
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=2
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=1
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=no_large_cup price=20 k=0 k1=1 t=0 s=1
RETURN COIN
return value=1 state=no_large_cup price=20 k=0 k1=1 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
RETURN COIN
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
```

```
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
RETURN COIN
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=1 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=1 k1=1 t=0 s=2
```

Test#21: set_price 20 insert_small_cups 1 insert_large_cups 1 coin sugar small_cup cancel coin sugar small_cup tea insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
RETURN COINS
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=1 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=1 k1=1 t=0 s=2          Test Passed
```

Test#22: set_price 20 insert_small_cups 2 insert_large_cups 2 coin sugar large_cup tea coin sugar large_cup cancel coin sugar large_cup tea insert_large_cups 1 coin large_cup sugar tea insert_large_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=1
RETURN COINS
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=0
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=2 t=0 s=1
return value=1 state=Sugar price=20 k=1 k1=2 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=no_large_cup price=20 k=0 k1=2 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=2 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=2 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=2 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=2 t=0 s=1          Test Passed

Test#23: set_price 20 insert_small_cups 2 insert_large_cups 2 coin small_cup sugar tea coin large_cup sugar
tea coin small_cup sugar tea insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=2 t=0 s=0
return value=1 state=Idle price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=2 t=0 s=2
return value=1 state=Sugar price=20 k=2 k1=2 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=2 k1=1 t=0 s=2
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=2 k1=1 t=0 s=1
return value=1 state=Sugar price=20 k=2 k1=1 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=1
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=1 t=0 s=2
return value=1 state=Sugar price=20 k=1 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=no_small_cup price=20 k=1 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=1 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=1 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=1 k1=1 t=0 s=2          Test Passed

Test#24: set_price -10 coin dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
```

```
Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#25: small_cup dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#26: large_cup dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#27: sugar dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#28: tea dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#29: insert_small_cups 0 dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#30: insert_small_cups 0 dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
```

```
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#31: set_price -10 dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#32: cancel dispose
return value=0 state=Idle price=0 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=0 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=0 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=0 k=0 k1=0 t=0 s=0  Test Passed

Test#33: set_price 20 insert_small_cups 1 coin small_cup tea insert_small_cups -1 insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed

Test#34: set_price 20 insert_small_cups 1 coin small_cup tea small_cup insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed
```

Test#35: set_price 20 insert_small_cups 1 coin small_cup tea large_cup insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:      return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:    return value=1 state=exit price=20 k=0 k1=1 t=0 s=2      Test Passed
```

Test#36: set_price 20 insert_small_cups 1 coin small_cup tea sugar insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:      return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:    return value=1 state=exit price=20 k=0 k1=1 t=0 s=2      Test Passed
```

Test#37: set_price 20 insert_small_cups 1 coin small_cup tea tea insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:      return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:    return value=1 state=exit price=20 k=0 k1=1 t=0 s=2      Test Passed
```

Test#38: set_price 20 insert_small_cups 1 coin small_cup tea insert_large_cups 1 insert_small_cups 1 dispose

```
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed

Test#39: set_price 20 insert_small_cups 1 coin small_cup tea set_price 10 insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed

Test#40: set_price 20 insert_small_cups 1 coin small_cup tea cancel insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed

Test#41: set_price 20 insert_small_cups 1 coin small_cup tea dispose insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
```

```
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:      return value=1 state=exit price=20 k=0 k1=1 t=0 s=2        Test Passed

Test#42: set_price 20 insert_large_cups 1 coin large_cup tea insert_large_cups -1 insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:      return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed

Test#43: set_price 20 insert_large_cups 1 coin large_cup tea small_cup insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:      return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed

Test#44: set_price 20 insert_large_cups 1 coin large_cup tea large_cup insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
```

```
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed

Test#45: set_price 20 insert_large_cups 1 coin large_cup tea sugar insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed

Test#46: set_price 20 insert_large_cups 1 coin large_cup tea tea insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed

Test#47: set_price 20 insert_large_cups 1 coin large_cup tea insert_small_cups 1 insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1        Test Passed
```

```
Test#48: set_price 20 insert_large_cups 1 coin large_cup tea set_price 10 insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1          Test Passed

Test#49: set_price 20 insert_large_cups 1 coin large_cup tea cancel insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1          Test Passed

Test#50: set_price 20 insert_large_cups 1 coin large_cup tea dispose insert_large_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=1 k1=0 t=0 s=1
DISPOSE LARGE CUP OF TEA
return value=1 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=0 state=no_large_cup price=20 k=0 k1=0 t=0 s=1
return value=1 state=Idle price=20 k=1 k1=0 t=0 s=1
SHUT DOWN
return value=1 state=exit price=20 k=1 k1=0 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=1 k1=0 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=1 k1=0 t=0 s=1          Test Passed

Test#51: set_price 10 coin tea cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=0 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
```

```
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:         return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:       return value=1 state=exit price=10 k=0 k1=0 t=0 s=0        Test passed

Test#52: set_price 10 coin insert_large_cups 1 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=0 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:         return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:       return value=1 state=exit price=10 k=0 k1=0 t=0 s=0        Test passed

Test#53: set_price 10 coin insert_small_cups 1 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=0 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:         return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:       return value=1 state=exit price=10 k=0 k1=0 t=0 s=0        Test passed

Test#54: set_price 10 coin set_price 10 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=0 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:         return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:       return value=1 state=exit price=10 k=0 k1=0 t=0 s=0        Test passed

Test#55: set_price 10 coin dispose cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=0 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
RETURN COINS
```

```
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0          Test passed

Test#56: set_price 10 coin sugar tea cancel cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=1 state=Sugar price=10 k=0 k1=0 t=0 s=0
return value=0 state=Sugar price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=0 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0          Test passed

Test#57: set_price 10 coin sugar insert_large_cups 1 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=1 state=Sugar price=10 k=0 k1=0 t=0 s=0
return value=0 state=Sugar price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0          Test passed

Test#58: set_price 10 coin sugar insert_small_cups 1 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=1 state=Sugar price=10 k=0 k1=0 t=0 s=0
return value=0 state=Sugar price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:          return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0          Test passed

Test#59: set_price 10 coin sugar set_price 10 cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=1 state=Sugar price=10 k=0 k1=0 t=0 s=0
return value=0 state=Sugar price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:      return value=1 state=exit price=10 k=0 k1=0 t=0 s=0       Test passed

Test#60: set_price 10 coin sugar dispose cancel dispose
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
return value=1 state=Coin_Inserted price=10 k=0 k1=0 t=0 s=0
return value=1 state=Sugar price=10 k=0 k1=0 t=0 s=0
return value=0 state=Sugar price=10 k=0 k1=0 t=0 s=0
RETURN COINS
return value=1 state=Idle price=10 k=0 k1=0 t=0 s=0
SHUT DOWN
return value=1 state=exit price=10 k=0 k1=0 t=0 s=0

Actual result:        return value=1 state=exit price=10 k=0 k1=0 t=0 s=0
Expected result:      return value=1 state=exit price=10 k=0 k1=0 t=0 s=0       Test passed

Test#61: set_price 20 insert_large_cups 10 insert_small_cups 10 coin sugar small_cup tea dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=10 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=10 k1=9 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=10 k1=9 t=0 s=2

Actual result:        return value=1 state=exit price=20 k=10 k1=9 t=0 s=2
Expected result:      return value=1 state=exit price=20 k=10 k1=9 t=0 s=2      Test passed

Test#62: set_price 20 insert_small_cups 10 insert_large_cups 10 coin sugar large_cup tea dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=10 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=10 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=1
DISPOSE LARGE CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=9 k1=10 t=0 s=1
SHUT DOWN
```

```
return value=1 state=exit price=20 k=9 k1=10 t=0 s=1

Actual result:          return value=1 state=exit price=20 k=9 k1=10 t=0 s=1
Expected result:        return value=1 state=exit price=20 k=9 k1=10 t=0 s=1

Test#63: set_price 20 insert_large_cups 10 insert_small_cups 10 coin sugar small_cup tea dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=10 k1=10 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=10 k1=10 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=10 k1=9 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=10 k1=9 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=10 k1=9 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=10 k1=9 t=0 s=2      Test passed

Test#64: set_price 20 insert_small_cups 10 coin sugar small_cup tea dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=10 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=10 t=0 s=0
return value=1 state=Sugar price=20 k=0 k1=10 t=0 s=2
DISPOSE SMALL CUP OF TEA WITH SUGAR
return value=1 state=Idle price=20 k=0 k1=9 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=9 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=9 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=9 t=0 s=2       Test passed

Test#65: set_price 20 insert_small_cups 1 coin small_cup tea insert_large_cups -1 insert_small_cups 1 dispose
return value=1 state=Idle price=20 k=0 k1=0 t=0 s=0
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=0
return value=1 state=Coin_Inserted price=20 k=0 k1=1 t=0 s=2
DISPOSE SMALL CUP OF TEA
return value=1 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=0 state=no_small_cup price=20 k=0 k1=0 t=0 s=2
return value=1 state=Idle price=20 k=0 k1=1 t=0 s=2
SHUT DOWN
return value=1 state=exit price=20 k=0 k1=1 t=0 s=2

Actual result:          return value=1 state=exit price=20 k=0 k1=1 t=0 s=2
Expected result:        return value=1 state=exit price=20 k=0 k1=1 t=0 s=2       Test Passed
End of the file
```

5. Conclusion

Fortunately, I couldn't find any defect from VendingMachine class while executing test cases. To implement test environment, I added test-oriented methods in the VendingMachine class which do not have influence on original source code. Test-oriented methods were useful when I should check the state whenever I need and when I should have to see the internal processing of operations.

As I performed in this project, testing with input files can be automated to get actual result. For automating the process, I used 'Pattern' and 'Match' with regular expressions. Whenever the 'matcher' meets operations which is matched to regular expression. And then, the operation is removed from string so that test driver can execute next operation (If I have next chance to test software, I will try to use stack data structure). However, expecting result should be partially automated since EFSM and source code can be different from each other.

6. Source Code

class VendingMachine

```
package vendingMachine;

public class VendingMachine {
    private int x;
    private int price;
    private int k;
    private int k1;
    private int t;
    private int s;

    public VendingMachine() {
        k1 = 0;
        k = 0;
        t = 0;
        price = 0;
        x = 1;
    }

    // method for testing, show price with value 'price'
    public int show_price() {
        return price;
    }

    // method for testing, show # of large cups with value 'k'
    public int show_nOfLargeCup() {
        return k;
    }

    // method for testing, show # of small cups with value 'k1'
    public int show_nOfSmallCup() {
        return k1;
    }

    // method for testing, show current value with value 't'
    public int show_currentvalue() {
        return t;
    }

    // method for testing, show size of cup with value 's'
    public int show_sizeofcup() {
        return s;
    }

    // method for testing, show current state with value 'x'
    public void show_state() {
        switch(x) {
```

```
        case 1:
            System.out.print("Idle");
            break;
        case 2:
            System.out.print("Coin_Inserted");
            break;
        case 3:
            System.out.print("Sugar");
            break;
        case 4:
            System.out.print("no_small_cup");
            break;
        case 5:
            System.out.print("no_large_cup");
            break;
        case 6:
            System.out.print("exit");
            break;
            default:
                break;
        }
    }

    public final int coin() {
        if (x == 1) {
            if ((t + 25 >= price) && (price > 0)) {
                s = 0;
                t = 0;
                x = 2;
                return 1;
            } else if (t + 25 < price) {
                t = t + 25;
                return 1;
            }
        } else if ((x > 1) && (x < 6)) {
            System.out.print("RETURN COIN");
            System.out.print("\n");
            return 1;
        }
        return 0;
    }

    public final int small_cup() {
        if ((x == 2) || (x == 3)) {
            s = 2;
            return 1;
        }
        return 0;
    }
```

```java
public final int large_cup() {
        if ((x == 2) || (x == 3)) {
                s = 1;
                return 1;
        }
        return 0;
}

public final int sugar() {
        if ((x == 2) || (x == 3)) {
                if (x == 2) {
                        x = 3;
                } else {
                        x = 2;
                }
                return 1;
        }
        return 0;
}

public final int tea() {
        if ((x == 2) || (x == 3)) {
                if ((x == 2) && (k1 > 1) && (s == 2)) {
                        System.out.print("DISPOSE SMALL CUP OF TEA");
                        System.out.print("\n");
                        k1 = k1 - 1;
                        x = 1;
                        return 1;
                } else if ((x == 2) && (k > 1) && (s == 1)) {
                        System.out.print("DISPOSE LARGE CUP OF TEA");
                        System.out.print("\n");
                        k = k - 1;
                        x = 1;
                        return 1;
                } else if ((x == 2) && (k == 1) && (s == 1)) {
                        System.out.print("DISPOSE LARGE CUP OF TEA");
                        System.out.print("\n");
                        k = k - 1;
                        x = 5;
                        return 1;
                } else if ((x == 2) && (k1 == 1) && (s == 2)) {
                        System.out.print("DISPOSE SMALL CUP OF TEA");
                        System.out.print("\n");
                        k1 = k1 - 1;
                        x = 4;
                        return 1;
                } else if ((x == 3) && (k1 == 1) && (s == 2)) {
                        System.out.print("DISPOSE SMALL CUP OF TEA WITH SUGAR");
```

```
            System.out.print("\n");
            k1 = k1 - 1;
            x = 4;
            return 1;
        } else if ((x == 3) && (k == 1) && (s == 1)) {
            System.out.print("DISPOSE LARGE CUP OF TEA WITH SUGAR");
            System.out.print("\n");
            k = k - 1;
            x = 5;
            return 1;
        }
        if ((x == 3) && (k1 > 1) && (s == 2)) {
            System.out.print("DISPOSE SMALL CUP OF TEA WITH SUGAR");
            System.out.print("\n");
            k1 = k1 - 1;
            x = 1;
            return 1;
        } else if ((x == 3) && (k > 1) && (s == 1)) {
            System.out.print("DISPOSE LARGE CUP OF TEA WITH SUGAR");
            System.out.print("\n");
            k = k - 1;
            x = 1;
            return 1;
        }
        return 0;
    }
    return 0;
}

public final int insert_large_cups(int n) {
    if ((x == 1) && (n > 0)) {
        k = k + n;
        return 1;
    } else if ((x == 5) && (n > 0)) {
        k = n;
        x = 1;
        return 1;
    }
    return 0;
}

public final int insert_small_cups(int n) {
    if ((x == 1) && (n > 0)) {
        k1 = k1 + n;
        return 1;
    } else if ((x == 4) && (n > 0)) {
        k1 = n;
        x = 1;
        return 1;
```

```
		}
		return 0;
	}

	public final int set_price(int p) {
		if ((x == 1) && (p > 0)) {
			price = p;
			return 1;
		}
		return 0;
	}

	public final int cancel() {
		if ((x == 2) || (x == 3)) {
			System.out.print("RETURN COINS");
			System.out.print("\n");
			x = 1;
			return 1;
		}
		return 0;
	}

	public final int dispose() {
		if ((x == 1)) {
			System.out.print("SHUT DOWN");
			System.out.print("\n");
			x = 6;
			return 1;
		}
		return 0;
	}
}
```

class TestDriver

- TestDriver class has 3 methods:
    - main()
    - void menu()
    - static public void testWithTestSuite(BufferedReader read)

```
package vendingMachine;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class TestDriver {

public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);
        String scInput;
        char menuInput;

        int nOfSmallCup;
        int nOfLargeCup;
        int sizeOfCup;
        int price;
        int currentvalue;

        VendingMachine mainVendingMachine = new VendingMachine();

        FileReader TS = new FileReader(new File("TS.txt"));
        BufferedReader read = new BufferedReader(TS);

        while(true) {
                Menu();                                 // Call Menu
                scInput = sc.next();                    // Get Input from Console
                menuInput = scInput.charAt(0);

                switch(menuInput) {
                case '0':       // coin method of main
                        System.out.println("coin() method");
                        System.out.println("The value returned by the method : "+mainVendingMachine.coin());
                        System.out.println("Press any key to continue\n");
                        System.in.read();
                        break;

                case '1':       // small_cup method of main
                        System.out.println("small_cup() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.small_cup());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '2':       // large_cup method of main
                System.out.println("large_cup() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.large_cup());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '3':       // sugar method of main
                System.out.println("sugar() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.sugar());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '4':       // tea method of main
                System.out.println("tea() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.tea());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '5':       //  insert_large_cups(int n) method of main
                int nlarge;     // number of cups to be inserted
                System.out.println("insert_large_cups(int n) method");
                System.out.print("Enter value of parameter n : ");
                nlarge = sc.nextInt();
                System.out.println("The value returned by the method : " +
mainVendingMachine.insert_large_cups(nlarge));
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '6':       // insert_small_cups(int n) method of main
                int nsmall;     // number of cups to be inserted
                System.out.println("insert_small_cups(int n) method");
                System.out.print("Enter value of parameter n : ");
                nsmall = sc.nextInt();
                System.out.println("The value returned by the method : " +
mainVendingMachine.insert_small_cups(nsmall));
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;
        case '7':       // set_price(int n) of main
                int nprice;     // number of cups to be inserted
                System.out.println("set_price(int n) method");
                System.out.print("Enter value of parameter n : ");
                nprice = sc.nextInt();
                System.out.println("The value returned by the method : " +
mainVendingMachine.set_price(nprice));
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;

        case '8': {     // cancel method of main
                System.out.println("cancel() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.cancel());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;
        }
        case '9': {     // dispose method of main
                System.out.println("dispose() method");
                System.out.println("The value returned by the method : " + mainVendingMachine.dispose());
                System.out.println("Press any key to continue\n");
                System.in.read();
                break;
        }
        case 'a':       // Show Variables of Vendeng Machine of main
                sizeOfCup = mainVendingMachine.show_sizeofcup();
                nOfLargeCup = mainVendingMachine.show_nOfLargeCup();
                nOfSmallCup = mainVendingMachine.show_nOfSmallCup();
                price = mainVendingMachine.show_price();
                currentvalue = mainVendingMachine.show_currentvalue();

                mainVendingMachine.show_state();
                System.out.print(" price=" + price);
                System.out.print(" k=" + nOfLargeCup);
                System.out.print(" k1=" + nOfSmallCup);
                System.out.print(" t=" + currentvalue);
                System.out.print(" s=" + sizeOfCup);
                System.out.println();
                break;
        case 'b': {     // Test with TestSuite
                testWithTestSuite(read);
                break;
        }
        case 'q': {
                return;
        }
      }
    }
  }
}
```

```
static public void Menu() {
		System.out.println("\t\tDRIVER for the vending machine");

		System.out.println("\t0. coin()");
		System.out.println("\t1. small_cup()");
		System.out.println("\t2. large_cup()");
		System.out.println("\t3. sugar()");
		System.out.println("\t4. tea()");
		System.out.println("\t5. insert_large_cups(int n)");
		System.out.println("\t6. insert_small_cups(int n)");
		System.out.println("\t7. set_price(int p)");
		System.out.println("\t8. cancel()");
		System.out.println("\t9. dispose()");

		System.out.println("\nTesting-related methods");
		System.out.println("a. Show_variables()");
		System.out.println("b. Test Vending Machine with TS.txt");
		System.out.println("q. Quit Vending Machine class driver");
}
```

```
static public void testWithTestSuite(BufferedReader read) throws NumberFormatException, IOException {
        // Get Buffered TS.txt from main method as parameter

        String testCase;
        int nOfSmallCup;
        int nOfLargeCup;
        int sizeOfCup;
        int price;
        int result;
        int currentvalue;

        // get test case line by line from TS
        while ((testCase = read.readLine()) != null) {
            VendingMachine vendingMachine = new VendingMachine();

            // if test case starts with $$ then end of file
            if(testCase.startsWith("$$")) {
                System.out.println("End of the file");
                break;
            }

            // split test case into ‘Test# n’ and ‘operations’ with “: “
            String[] test = testCase.split(": ");
            String testNumber = test[0];
            String operations = test[1];

            System.out.println(testNumber+": " + operations);

            /* define pattern and matcher to find operations in test case while testing is going on */
            Pattern p_set_price = Pattern.compile("set_price -?[0-9][0-9]?[0-9]?");
            Matcher m_set_price = p_set_price.matcher(operations);

            Pattern p_insert_small_cups = Pattern.compile("insert_small_cups -?[0-9][0-9]?[0-9]?");
            Matcher m_insert_small_cups = p_insert_small_cups.matcher(operations);

            Pattern p_insert_large_cups = Pattern.compile("insert_large_cups -?[0-9][0-9]?[0-9]?");
            Matcher m_insert_large_cups = p_insert_large_cups.matcher(operations);

            Pattern p_coin = Pattern.compile("coin");
            Matcher m_coin = p_coin.matcher(operations);

            Pattern p_small_cup = Pattern.compile("small_cup");
            Matcher m_small_cup = p_small_cup.matcher(operations);

            Pattern p_large_cup = Pattern.compile("large_cup");
            Matcher m_large_cup = p_large_cup.matcher(operations);

            Pattern p_sugar = Pattern.compile("sugar");
            Matcher m_sugar = p_sugar.matcher(operations);
```

```
Pattern p_tea = Pattern.compile("tea");
Matcher m_tea = p_tea.matcher(operations);

Pattern p_cancel = Pattern.compile("cancel");
Matcher m_cancel = p_cancel.matcher(operations);

Pattern p_dispose = Pattern.compile("dispose");
Matcher m_dispose = p_dispose.matcher(operations);

// While the string is null
while(!operations.equals("")) {

        if(operations.startsWith("set_price") && m_set_price.find()) {

        // if pattern matched, split test case into  operation list and parameter with “ “
        // then remove the operation and parameter from operation list string

        String settemp = m_set_price.group(0);
        String[] parameterOfSetPrice = settemp.split(" ");
        operations = operations.substring(settemp.length()+1, operations.length());

        //Get return value and class variables from VendingMachine class
        result = vendingMachine.set_price(Integer.parseInt(parameterOfSetPrice[1]));

        sizeOfCup = vendingMachine.show_sizeofcup();
        nOfLargeCup = vendingMachine.show_nOfLargeCup();
        nOfSmallCup = vendingMachine.show_nOfSmallCup();
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        // print status of Vending Machine class
        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();


}

else if(operations.startsWith("insert_small_cups") && m_insert_small_cups.find()) {
        String smallcuptemp = m_insert_small_cups.group(0);
        String[] parameterOfInsertSmallCups = smallcuptemp.split(" ");

operations = operations.substring(smallcuptemp.length()+1, operations.length());
```

```
    result = vendingMachine.insert_small_cups(Integer.parseInt(parameterOfInsertSmallCups[1]));
    sizeOfCup = vendingMachine.show_sizeofcup();
    nOfLargeCup = vendingMachine.show_nOfLargeCup();
    nOfSmallCup = vendingMachine.show_nOfSmallCup();
    price = vendingMachine.show_price();
    currentvalue = vendingMachine.show_currentvalue();

    System.out.print("return value=" + result + " state=");
    vendingMachine.show_state();
    System.out.print(" price=" + price);
    System.out.print(" k=" + nOfLargeCup);
    System.out.print(" k1=" + nOfSmallCup);
    System.out.print(" t=" + currentvalue);
    System.out.print(" s=" + sizeOfCup);
    System.out.println();
}

else if(operations.startsWith("insert_large_cups") && m_insert_large_cups.find()) {
    String largecuptemp = m_insert_large_cups.group(0);
    String[] parameterOfInsertLargeCups = largecuptemp.split(" ");

    operations = operations.substring(largecuptemp.length()+1, operations.length());

    result = vendingMachine.insert_large_cups(Integer.parseInt(parameterOfInsertLargeCups[1]));
    sizeOfCup = vendingMachine.show_sizeofcup();
    nOfLargeCup = vendingMachine.show_nOfLargeCup();
    nOfSmallCup = vendingMachine.show_nOfSmallCup();
    price = vendingMachine.show_price();
    currentvalue = vendingMachine.show_currentvalue();

    System.out.print("return value=" + result + " state=");
    vendingMachine.show_state();
    System.out.print(" price=" + price);
    System.out.print(" k=" + nOfLargeCup);
    System.out.print(" k1=" + nOfSmallCup);
    System.out.print(" t=" + currentvalue);
    System.out.print(" s=" + sizeOfCup);
    System.out.println();
}

else if(operations.startsWith("coin") && m_coin.find()) {
    String cointemp = m_coin.group(0);
    operations = operations.substring(cointemp.length()+1, operations.length());

    result = vendingMachine.coin();
    sizeOfCup = vendingMachine.show_sizeofcup();
    nOfLargeCup = vendingMachine.show_nOfLargeCup();
    nOfSmallCup = vendingMachine.show_nOfSmallCup();
```

```
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();
    }

    else if(operations.startsWith("small_cup") && m_small_cup.find()) {
        String smallcuptemp = m_small_cup.group(0);
        operations = operations.substring(smallcuptemp.length()+1, operations.length());

        result = vendingMachine.small_cup();
        sizeOfCup = vendingMachine.show_sizeofcup();
        nOfLargeCup = vendingMachine.show_nOfLargeCup();
        nOfSmallCup = vendingMachine.show_nOfSmallCup();
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();
    }

    else if(operations.startsWith("large_cup") && m_large_cup.find()) {
        String largecuptemp = m_large_cup.group(0);
        operations = operations.substring(largecuptemp.length()+1, operations.length());

        result = vendingMachine.large_cup();
        sizeOfCup = vendingMachine.show_sizeofcup();
        nOfLargeCup = vendingMachine.show_nOfLargeCup();
        nOfSmallCup = vendingMachine.show_nOfSmallCup();
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();
}

else if(operations.startsWith("sugar") && m_sugar.find()) {
        String sugartemp = m_sugar.group(0);
        operations = operations.substring(sugartemp.length()+1, operations.length());

        result = vendingMachine.sugar();
        sizeOfCup = vendingMachine.show_sizeofcup();
        nOfLargeCup = vendingMachine.show_nOfLargeCup();
        nOfSmallCup = vendingMachine.show_nOfSmallCup();
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();
}

else if(operations.startsWith("tea") && m_tea.find()) {
        String teatemp = m_tea.group(0);
        operations = operations.substring(teatemp.length()+1, operations.length());

        result = vendingMachine.tea();
        sizeOfCup = vendingMachine.show_sizeofcup();
        nOfLargeCup = vendingMachine.show_nOfLargeCup();
        nOfSmallCup = vendingMachine.show_nOfSmallCup();
        price = vendingMachine.show_price();
        currentvalue = vendingMachine.show_currentvalue();

        System.out.print("return value=" + result + " state=");
        vendingMachine.show_state();
        System.out.print(" price=" + price);
        System.out.print(" k=" + nOfLargeCup);
        System.out.print(" k1=" + nOfSmallCup);
        System.out.print(" t=" + currentvalue);
        System.out.print(" s=" + sizeOfCup);
        System.out.println();
}

else if(operations.startsWith("cancel") && m_cancel.find()) {
```

```
                String canceltemp = m_cancel.group(0);
                operations = operations.substring(canceltemp.length()+1, operations.length());

                result = vendingMachine.cancel();
                sizeOfCup = vendingMachine.show_sizeofcup();
                nOfLargeCup = vendingMachine.show_nOfLargeCup();
                nOfSmallCup = vendingMachine.show_nOfSmallCup();
                price = vendingMachine.show_price();
                currentvalue = vendingMachine.show_currentvalue();

                System.out.print("return value=" + result + " state=");
                vendingMachine.show_state();
                System.out.print(" price=" + price);
                System.out.print(" k=" + nOfLargeCup);
                System.out.print(" k1=" + nOfSmallCup);
                System.out.print(" t=" + currentvalue);
                System.out.print(" s=" + sizeOfCup);
                System.out.println();
            }

            else if(operations.startsWith("dispose") && m_dispose.find()) {
                String disposetemp = m_dispose.group(0);
                operations = operations.substring(disposetemp.length()+1, operations.length());

                result = vendingMachine.dispose();
                sizeOfCup = vendingMachine.show_sizeofcup();
                nOfLargeCup = vendingMachine.show_nOfLargeCup();
                nOfSmallCup = vendingMachine.show_nOfSmallCup();
                price = vendingMachine.show_price();
                currentvalue = vendingMachine.show_currentvalue();

                System.out.print("return value=" + result + " state=");
                vendingMachine.show_state();
                System.out.print(" price=" + price);
                System.out.print(" k=" + nOfLargeCup);
                System.out.print(" k1=" + nOfSmallCup);
                System.out.print(" t=" + currentvalue);
                System.out.print(" s=" + sizeOfCup);
                System.out.println();
            }
        }
    }
}
```