STA CS-589
FINAL PROJECT

VENDING MACHINE

SWAGATAM GUHATHAKURTA
A20358829
Sguhatha.hawk.iit.edu

Table of Contents

1. MBT : Transition Pair Testing

ID	Transition
1	Vending_machine / k=0;k1=0;t=0; price=0
2	Insert_large_cups(n)[n>0]/k=k+n
3	Insert_small+cups(n)[n>0]/k1=k1+n
4	Set_price(p)[p>0]/price=p
5	Dispose
6	Coin[t+25<price] / t=t+25
7	Coin[(t+25>=price)&&(price>0)]/s=0;t=0
8	Insert_large_cups(n)[n>0]/k=n
9	Insert_small_cups(n)[n>0]/k1=n
10	Cancel/return coins
11	Tea[(k1>1)&&(s==2)]/dispose small cup of tea; k1=k1-1
12	Tea[(k>1)&&(s==1)]/dispose large cup of tea; k=k-1
13	Tea[(k>1)&&(s==1)]/dispose large cup of tea with sugar;k=k-1
14	Cacel/ return coins
15	Tea[(k1>1)&&(s==2)]/dispose small cup of tea with sugar; k1=k1-1
16	Coin/return coin
17	Small_cup / s=2
18	Large_cup / s=1
19	Large_cup / s=1
20	Coin / return coin
21	Small_cup / s=2
22	Sugar
23	Sugar
24	Tea[(k==1) && (s==1)]/dispose large cup of tea; k=k-1
25	Tea[(k1==1) && (s==2)]/dispose small cup of tea; k1=k1-1
26	Tea[(k==1) && (s==1)]/dispose large cup of tea with sugar; k=k-1
27	Tea[(k1==1) && (s==2)]/dispose small cup of tea with sugar; k1=k1-1
28	Coin / return coin
29	Coin / return coin

Transition pair for idle

(T1,T2)	(T1,T3)	(T1,T7)	(T1,T6)	(T1,T4)	(T1,T5)
(T2,T2)	(T2,T3)	(T2,T7)	(T2,T6)	(T2,T4)	(T2,T5)
(T3,T2)	(T3,T3)	(T3,T7)	(T3,T6)	(T3,T4)	(T3,T5)
(T15,T2)	(T15,T3)	(T15,T7)	(T15,T6)	(T15,T4)	(T15,T5)
(T14,T2)	(T14,T3)	(T14,T7)	(T14,T6)	(T14,T4)	(T14,T5)
(T13,T2)	(T13,T3)	(T13,T7)	(T13,T6)	(T13,T4)	(T13,T5)
(T12,T2)	(T12,T3)	(T12,T7)	(T12,T6)	(T12,T4)	(T12,T5)
(T11,T2)	(T11,T3)	(T11,T7)	(T11,T6)	(T11,T4)	(T11,T5)
(T10,T2)	(T10,T3)	(T10,T7)	(T10,T6)	(T10,T4)	(T10,T5)
(T6,T2)	(T6,T3)	(T6,T7)	(T6,T6)	(T6,T4)	(T6,T5)
(T8,T2)	(T8,T3)	(T8,T7)	(T8,T6)	(T8,T4)	(T8,T5)
(T9,T2)	(T9,T3)	(T9,T7)	(T9,T6)	(T9,T4)	(T9,T5)
(T4,T2)	(T4,T3)	(T4,T7)	(T4,T6)	(T4,T4)	(T4,T5)

Transition pair for coins inserted

(T7,T11)	(T7,T12)	(T7,T20)	(T7,T21)	(T7,T22)	(T7,T25)	(T7,T24)	(T7,T19)	(T7,T10)
(T20,T11)	(T20,T12)	(T20,T20)	(T20,T21)	(T20,T22)	(T20,T25)	(T20,T24)	(T20,T19)	(T20,T10)
(T21,T11)	(T21,T12)	(T21,T20)	(T21,T21)	(T21,T22)	(T21,T25)	(T21,T24)	(T21,T19)	(T21,T10)
(T23,T11)	(T23,T12)	(T23,T20)	(T23,T21)	(T23,T22)	(T23,T25)	(T23,T24)	(T23,T19)	(T23,T10)
(T19,T11)	(T19,T12)	(T19,T20)	(T19,T21)	(T19,T22)	(T19,T25)	(T19,T24)	(T19,T19)	(T19,T10)

Transition pair for sugar

(T22,T13)	(T22,T14)	(T22,T15)	(T22,T16)	(T22,T17)	(T22,T18)	(T22,T27)	(T22,T26)	(T22,T23)
(T16,T13)	(T16,T14)	(T16,T15)	(T16,T16)	(T16,T17)	(T16,T18)	(T16,T27)	(T16,T26)	(T16,T23)
(T17,T13)	(T17,T14)	(T17,T15)	(T17,T16)	(T17,T17)	(T17,T18)	(T17,T27)	(T17,T26)	(T17,T23)
(T18,T13)	(T18,T14)	(T18,T15)	(T18,T16)	(T18,T17)	(T18,T18)	(T18,T27)	(T18,T26)	(T18,T23)

Transition pair for no large_cups

(T24,T29)	(T24,T8)
(T26,T29)	(T26,T8)
(T29,T29)	(T29,T8)

Transition pair for no small_cups

(T25,T28)	(T25,T9)
(T27,T28)	(T27,T9)
(T28,T28)	(T28,T9)

Note:

The points parked in red are not executable.

- **(T7,T11),(T7,T12),(T7,T25),(T7,T24): In all these pair transition the precondition fails.Without the value of s assigned to either 1 or 2 the transitions cannot be executed. Here s=0.**
- **(T21,T12),(T21,T24): s=2 but the pairs requires s to be 1.**
- **(T19,T11),(T19,T25): s=1 but the pairs requires s to be 2.**
- **(T17,T13),(T17,T26): s=2 but the pairs requires s to be 1.**
- **(T18,T15),(T18,T27): s=1 but the pairs requires s to be 2.**

Test Cases:

ID	Test Cases	Transition pairs Covered	Covered by Test Case # in TS.txt
1	vending_machine(),insert_large_cups(1) ,insert_small_cups(1), insert_small_cups(1),set_price(25), set_price(50),coin(),coin(),small_cup(),small_cup(),sugar(), tea(),coin(),coin(),large_cup(),large_cup(),sugar(),tea(), coin(),coin(),insert_large_cups(1),coin(),coin(),coin(),coin(), large_cup(),tea(),insert_large_cups(1),dispose()	**(T1,T2),(T2,T3),(T3,T3) (T3,T4),(T4,T4),(T4,T6),(T6,T7), (T7,T21),(T21,T21),(T21,T22), (T22,T15),(T15,T6),(T6,T7),(T7,T19), (T19,T19),(T19,T22),(T22,T26), (T26,T29),(T29,T29),(T29,T8),(T8,T6), (T6,T7),(T7,T20),(T20,T20),(T20,T19), (T19,T24),(T24,T8),(T8,T5)**	**1**
2	vending_machine(),insert_small_cups(1) ,insert_large_cups(1), insert_large_cups(1),set_price(25),coin(), sugar(), large_cup(), large_cup(),tea(),set_price(75),coin(), coin(),coin(), coin(), small_cup(),tea(),coin(),coin(), insert_small_cups(1),dispose()	**(T1,T3),(T3,T2),(T2,T2) (T2,T4),(T4,T7),(T7,T22), (T22,T18),(T18,T18),(T18,T13), (T13,T4),(T4,T6),(T6,T6),(T6,T7), (T7,T20),(T20,T21),(T21,T25), (T25,T28),(T28,T28),(T28,T9), (T9,T5)**	**2**
3	vending_machine,set_price(25),insert_large_cups(1), coin(),cancel(),coin(),sugar(),coin(),sugar(),coin(), large_cup(),tea(),coin(),insert_large_cups(1),coin(), sugar(),cancel(),dispose()	**(T1,T4),(T4,T2),(T2,T7),(T7,T10), (T10,T7),(T7,T22),(T22,T16),(T16,T23) ,(T23,T20),(T20,T19),(T19,T24), (T24,T29),(T29,T8),(T8,T7),(T7,T22), (T22,T14),(T14,T5)**	**3**
4	vending_machine(),dispose()	**(T1,T5)**	**4**
5	vending_machine(),set_price(50),insert_large_cups(1), coin(),insert_large_cups(1),coin(),large_cup(),tea(), insert_small_cups(2),coin(),coin(),small_cup(),tea(),coin(), coin(),sugar(),small_cup(),tea(),coin(), insert_small_cups(1),coin(),coin(),sugar(),large_cup(),tea(), insert_large_cups(1),set_price(25),dispose()	**(T1,T4),(T4,T2),(T2,T6),(T6,T2),(T2,T7) ,(T7,T19),(T19,T12),(T12,T3),(T3,T6), (T6,T7),(T7,T21),(T21,T11),(T11,T6), (T6,T7),(T7,T22),(T22,T17),(T17,T27), (T27,T28),(T28,T9),(T9,T6),(T6,T7), (T7,T22),(T22,T18),(T18,T26),(T26,T8) ,(T8,T4),(T4,T5)**	**5**
6	vending_machine(),set_price(25), insert_small_cups(2), coin(),small_cup(),coin(),tea(), insert_large_cups(2),coin(), large_cup(),coin(),tea(),coin(),small_cup(),tea(), insert_small_cups(1),coin(),sugar(),small_cup(),tea(), insert_small_cups(1),coin(),large_cup(),coin(),tea(), insert_large_cups(1), insert_large_cups(1),dispose()	**(T1,T4),(T4,T3),(T3,T7),(T7,T21), (T21,T20),(T20,T11),(T11,T2),(T2,T7) ,(T7,T19),(T19,20),(T20,T12),(T12,T7) ,(T7,T21),(T21,T25),(T25,T9),(T9,T7), (T7,T22),(T22,T17),(T17,T27),(T27,T9) ,(T9,T7),(T7,T19),(T19,T20),(T20,T24), (T24,T8),(T8,T2),(T2,T5)**	**6**
7	vending_machine(),set_price(25), coin(),sugar(),cancel(), insert_large_cups(2),coin(),large_cup(),sugar(),tea(), insert_small_cups(2),coin(),small_cup(),sugar(),tea(),coin() ,small_cup(),sugar(),tea(),insert_small_cups(1), set_price(25),coin(),sugar(),sugar(),small_cup(),large_cup() ,small_cup(),cancel(),dispose()	**(T1,T4),(T4,T7),(T7,T22),(T22,T14), (T14,T2),(T2,T7),(T7,T19),(T19,T22), (T22,T13),(T13,T3),(T3,T7),(T7,T21), (T21,T22),(T22,T15),(T15,T7),(T7,T21) ,(T21,T22),(T22,T27),(T27,T9),(T9,T4), (T4,T7),(T7,T22),(T22,T23),(T23,T21), T(21,T19),(T19,T21),(T21,T10), (T10,T5)**	**7**

8	vending_machine(),set_price(25), insert_small_cups(3), coin(),sugar(),small_cup(),coin(),small_cup(),cancel(), coin(),sugar(),small_cup(),small_cup(),tea(), insert_large_cups(1),coin(),sugar(),small_cup(), large_cup(),small_cup(),sugar(),tea(),coin(),large_cup(), cancel(),set_price(25), insert_small_cups(2),dispose()	**(T1,T4),(T4,T3),(T3,T7),(T7,T22), (T22,T17),(T17,T16),(T16,T17), (T17,T14),(T14,T7),(T7,T22),(T22,T17) ,(T17,T17),(T17,T15),(T15,T2),(T2,T7), (T7,T22),(T22,T17),(T17,T18), (T18,T17),(T17,T23),(T23,T11), (T11,T7),(T7,T19),(T19,T10),(T10,T4), (T4,T3),(T3,T5)**	**8**
9	vending_machine(),set_price(25), insert_large_cups(3), coin(),sugar(),large_cup(),coin(),large_cup(),cancel(), coin(),sugar(),large_cup(),large_cup(),tea(), insert_small_cups(1),coin(),sugar(),large_cup(), small_cup(),large_cup(),sugar(),tea(),coin(),small_cup(), cancel(),set_price(25), insert_large_cups(2),dispose()	**(T1,T4),(T4,T2),(T2,T7),(T7,T22), (T22,T18),(T18,T16),(T16,T18), (T18,T14),(T14,T7),(T7,T22),(T22,T18) ,(T18,T18),(T18,T13),(T13,T3),(T3,T7), (T7,T22),(T22,T18),(T18,T17), (T17,T18),(T18,T23),(T23,T12), (T12,T7),(T7,T21),(T21,T10),(T10,T4), (T4,T2),(T2,T5)**	**9**
10	vending_machine(),set_price(25),coin(),sugar(),coin(), coin(),cancel(),insert_small_cups(2),coin(),sugar(), small_cup(),coin(),tea(),set_price(25),coin(),sugar(), small_cup(),coin(),tea(),insert_small_cups(1), insert_large_cups(2),coin(),sugar(),large_cup(),coin(),tea(), coin(),sugar(),large_cup(),coin(),tea(),insert_large_cups(1), insert_small_cups(1),dispose()	**(T1,T4),(T4,T7),(T7,T22),(T22,T16), (T16,T16),(T16,T14),(T14,T3),(T3,T7),(T7,T22),(T22,T17),(T17,T16),(T16,T15) ,(T15,T4),(T4,T7),(T7,T22),(T22,T17), (T17,T16),(T16,T27),(T27,T9),(T9,T2), (T2,T7),(T7,T22),(T22,T18),(T18,16), (T16,T13),(T13,T7),(T7,T22),(T22,T18) ,(T18,T16),(T16,T26),(T26,T8),(T8,T3), (T3,T5)**	**10**
11	vending_machine(),set_price(25), insert_large_cups(1), coin(),sugar(),sugar(),sugar(),sugar(),large_cup(),sugar(), sugar(),tea(),insert_large_cups(1), insert_small_cups(1), coin(),small_cup(),sugar(),sugar(),tea(), insert_small_cups(1),insert_small_cups(1),coin(),sugar(), sugar(),cancel(),insert_small_cups(1),dispose()	**(T1,T4),(T4,T2),(T2,T7),(T7,T22), (T22,T23),(T23,T22),(T22,T23), (T23,T19),(T19,T22),(T22,T23), (T23,T24),(T24,T8),(T8,T3),(T3,T7), (T7,T21),(T21,T22),(T22,T23), (T23,T25),(T25,T9),(T9,T3),(T3,T7), (T7,T22),(T22,T23),(T23,T10),(T10,T3) ,(T3,T5)**	**11**
12	vending_machine(),set_price(25), insert_small_cups(1), coin(),coin(),sugar(),sugar(),small_cup(),coin(),tea(), insert_small_cups(1),coin(),coin(),cancel(), insert_large_cups(1),dispose()	**(T1,T4),(T4,T3),(T3,T7),(T7,T20), (T20,T22),(T22,T23),(T23,T21), (T21,T20),(T20,T25),(T25,T9),(T9,T7), (T7,T20),(T20,T10),(T10,T2),(T2,T5)**	**12**
13	vending_machine(),set_price(100),coin(),set_price(75), coin(),insert_small_cups(4),coin(),small_cup(),tea(), set_price(25),coin(),small_cup(),tea(),insert_small_cups(1), coin(),small_cup(),tea(),dispose()	**(T1,T4),(T4,T6),(T6,T4),(T4,T6),(T6,T3) ,(T3,T7),(T7,T21),(T21,T11),(T11,T4), (T4,T7),(T7,T21),(T21,T11),(T11,T3), (T3,T7),(T7,T21),(T21,T11),(T11,T5)**	**13**
14	vending_machine(),set_price(25),insert_large_cups(4), coin(),large_cup(),tea(),insert_large_cups(1),coin(), large_cup(),tea(),set_price(50),coin(),coin(),large_cup(), tea(),coin(),coin(),large_cup(),tea(),dispose()	**(T1,T4),(T4,T2),(T2,T7),(T7,T19), (T19,T12),(T12,T2),(T2,T7),(T7,T19), (T19,T12),(T12,T4),(T4,T6),(T6,T7), (T7,T19),(T19,T12),(T12,T6),(T6,T7), (T7,T19),(T19,T12),(T12,T5)**	**14**
15	vending_machine(),set_price(25),insert_large_cups(4), coin(),sugar(),large_cup(),tea(),insert_large_cups(1),coin(),	**(T1,T4),(T4,T2),(T2,T7),(T7,T22), (T22,T18),(T18,T13),(T13,T2),(T2,T7),**	**15**

	sugar(),large_cup(),tea(),set_price(50),coin(),coin(), sugar(),large_cup(),tea(),coin(),coin(),large_cup(),tea(), dispose()	**(T7,T22),(T22,T18),(T18,T13),(T13,T4) ,(T4,T6),(T6,T7),(T7,T22),(T22,T18), (T18,T13),(T13,T6),(T6,T7),(T7,T22), (T22,T18),(T18,T13),(T13,T5)**	
16	vending_machine(),set_price(25),coin(),sugar(),cancel(), set_price(50),coin(),coin(),sugar(),cancel(),coin(),coin(), cancel(),coin(),dispose()	**(T1,T4),(T4,T7),(T7,T22),(T22,T14), (T14,T4),(T4,T6),(T6,T7),(T7,T22), (T22,T14),(T14,T6),(T6,T7),(T7,T10), (T10,T6),(T6,T5)**	**16**
17	vending_machine(),set_price(25), insert_small_cups(5), coin(),sugar(),small_cup(),tea(),insert_small_cups(1), coin(),sugar(),small_cup(),tea(),dispose()	**(T1,T4),(T4,T3),(T3,T7),(T7,T22), (T22,T17),(T17,T15),(T15,T3),(T3,T7), (T7,T22),(T22,T17),(T17,T15),(T15,T5)**	**17**

2. Ghost/Default Transition Testing :

STATE	TRANSITIONS COVERED	COVERED BY TEST CASES #
Idle	vending_machine(),coin(),insert_large_cups(0),insert_small_cups(0), set_price(0),cancel(),tea(),small_cup(),large_cup(),sugar(),dispose()	**18**
Coins inserted	vending_machine(),set_price(25),coin(),insert_large_cups(0), insert_small_cups(0),set_price(0), tea(),dispose()	**19**
Sugar	vending_machine(),set_price(25),coin(),sugar(), insert_large_cups(0),insert_small_cups(0),set_price(10), tea(),dispose()	**20**
No small cups	vending_machine(),set_price(25), insert_small_cups(1),coin(),small_cup(),tea(), insert_large_cups(0),insert_small_cups(0),set_price(0), cancel(),small_cup(),large_cup(),sugar(),tea(),dispose()	**21**
No large cups	vending_machine(),set_price(25), insert_large_cups(1),coin(),large_cup(),tea(), insert_large_cups(0),insert_small_cups(0),set_price(0), cancel(),small_cup(),large_cup(),sugar(),tea(),dispose()	**22**

3. MULTIPLE CONDITION TESTING:

1) public final int coin()

if (x == 1)

X==1	COVERED BY TEST CASES #
T	1
F	1

1.1) x==1 -> T
if ((t + 25 >= price) && (price > 0))

T+25>=price	Price>0	COVERED BY TEST CASES #
T	T	1
T	F	18
F	T	1
F	F	impossible

1.1.1) if ((t + 25 >= price) && (price > 0)) -> T & F

else if (t + 25 < price)

T+25<price	COVERED BY TEST CASES #
T	impossible
F	18

1.1.2) if ((t + 25 >= price) && (price > 0)) -> F & T

else if (t + 25 < price)

T+25<price	COVERED BY TEST CASES #
T	1
F	impossible

1.3)x==1 ->F
else if ((x > 1) && (x < 6))

x>1	X<6	COVERED BY TEST CASES #
T	T	1
T	F	Impossible
F	T	Impossible
F	F	impossible

2) public final int small_cup()

if ((x == 2) || (x == 3))

X==2	X==3	COVERED BY TEST CASES #
T	T	Impossible
T	F	1
F	T	5
F	F	18

3) public final int large_cup()

if ((x == 2) || (x == 3))

X==2	X==3	**COVERED BY TEST CASES #**
T	T	Impossible
T	F	1
F	T	5
F	F	18

4) public final int sugar()

if ((x == 2) || (x == 3))

X==2	X==3	**COVERED BY TEST CASES #**
T	T	Impossible
T	F	1
F	T	3
F	F	18

4.1)
if ((x == 2) || (x == 3)) -> T & F

if (x == 2)

X==2	**COVERED BY TEST CASES #**
T	1
F	Impossible

4.2)
if ((x == 2) || (x == 3)) -> F & T

if (x == 2)

X==2	**COVERED BY TEST CASES #**
T	Impossible
F	3

4.3.1) if (x == 2) ->F
Else

X!=2	**COVERED BY TEST CASES #**
T	3
F	Impossible

5) public final int tea()

if ((x == 2) || (x == 3))

X==2	X==3	**COVERED BY TEST CASES #**
T	T	impossible
T	F	1
F	T	1
F	F	18

5.1)
if ((x == 2) || (x == 3)) -> T & F

if ((x == 2) && (k1 > 1) && (s == 2))

X==2	K1>1	S==2	**COVERED BY TEST CASES #**
T	T	T	5
T	T	F	23
T	F	T	2
T	F	F	1
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 2) && (k > 1) && (s == 1))

X==2	K>1	S==1	**COVERED BY TEST CASES #**
T	T	T	5
T	T	F	24
T	F	T	1
T	F	F	2
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 2) && (k == 1) && (s == 1))

X==2	K==1	S==1	**COVERED BY TEST CASES #**
T	T	T	1
T	T	F	2
T	F	T	25
T	F	F	13
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 2) && (k1 == 1) && (s == 2))

X==2	K1==1	S==2	**COVERED BY TEST CASES #**
T	T	T	2
T	T	F	27
T	F	T	26
T	F	F	19
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 3) && (k1 == 1) && (s == 2))

X==3	K1==1	S==2	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	Not executable
F	T	F	Not executable
F	F	T	Not executable
f	F	F	19

else if ((x == 3) && (k == 1) && (s == 1))

X==3	K1==1	S==2	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	Not executable
F	T	F	Not executable
F	F	T	Not executable
f	F	F	19

if ((x == 3) && (k1 > 1) && (s == 2))

X==3	K1>1	S==2	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	Not executable
F	T	F	Not executable
F	F	T	Not executable
f	F	F	19

else if ((x == 3) && (k > 1) && (s == 1))

X==3	K>1	S==1	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	Not executable
F	T	F	Not executable
F	F	T	Not executable
f	F	F	19

5.2)

if ((x == 2) || (x == 3)) -> F & T

if ((x == 2) && (k1 > 1) && (s == 2))

X==2	K1>1	S==2	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	1
F	T	F	28
F	F	T	5
f	F	F	1

else if ((x == 2) && (k > 1) && (s == 1))

X==2	K>1	S==1	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	2
F	T	F	29
F	F	T	1
f	F	F	1

else if ((x == 2) && (k == 1) && (s == 1))

X==2	K>1	S==1	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	1
F	T	F	1
F	F	T	2
f	F	F	8

else if ((x == 2) && (k1 == 1) && (s == 2))

X==2	K1==1	S==2	**COVERED BY TEST CASES #**
T	T	T	impossible
T	T	F	impossible
T	F	T	impossible
T	F	F	impossible
F	T	T	5
F	T	F	1
F	F	T	1
f	F	F	7

else if ((x == 3) && (k1 == 1) && (s == 2))

X==3	K1==1	S==2	**COVERED BY TEST CASES #**
T	T	T	5
T	T	F	1
T	F	T	1
T	F	F	7
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 3) && (k == 1) && (s == 1))

X==3	K==1	S==1	**COVERED BY TEST CASES #**
T	T	T	1
T	T	F	1
T	F	T	2
T	F	F	8
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

if ((x == 3) && (k1 > 1) && (s == 2))

X==3	K1>1	S==2	**COVERED BY TEST CASES #**
T	T	T	1
T	T	F	28
T	F	T	29
T	F	F	2
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

else if ((x == 3) && (k > 1) && (s == 1))

X==3	K>1	S==1	**COVERED BY TEST CASES #**
T	T	T	2
T	T	F	29
T	F	T	28
T	F	F	20
F	T	T	impossible
F	T	F	impossible
F	F	T	impossible
f	F	F	impossible

6) public final int insert_large_cups(int n)

if ((x == 1) && (n > 0))

X==1	n>0	**COVERED BY TEST CASES #**
T	T	1
T	F	18
F	T	1
F	F	19

6.1) if ((x == 1) && (n > 0)) T & F

else if ((x == 5) && (n > 0))

X==5	n>0	**COVERED BY TEST CASES #**
T	T	impossible
T	F	impossible
F	T	impossible
F	F	18

6.2) if ((x == 1) && (n > 0)) F & T

else if ((x == 5) && (n > 0))

X==5	n>0	**COVERED BY TEST CASES #**
T	T	1
T	F	impossible
F	T	20
F	F	impossible

6.3) if ((x == 1) && (n > 0)) F & F

else if ((x == 5) && (n > 0))

X==5	n>0	**COVERED BY TEST CASES #**
T	T	impossible
T	F	22
F	T	impossible
F	F	19

7) public final int insert_small_cups(int n)

if ((x == 1) && (n > 0))

X==1	n>0	**COVERED BY TEST CASES #**
T	T	1
T	F	18
F	T	2
F	F	19

7.1) if ((x == 1) && (n > 0)) T & F
else if ((x == 4) && (n > 0))

X==4	n>0	**COVERED BY TEST CASES #**
T	T	impossible
T	F	impossible
F	T	impossible
F	F	18

7.2) if ((x == 1) && (n > 0)) F & T
else if ((x == 4) && (n > 0))

X==4	n>0	**COVERED BY TEST CASES #**
T	T	2
T	F	impossible
F	T	20
F	F	impossible

7.3) if ((x == 1) && (n > 0)) F & F
else if ((x == 4) && (n > 0))

X==4	n>0	**COVERED BY TEST CASES #**
T	T	impossible
T	F	21
F	T	impossible
F	F	19

8) public final int set_price(int p)
if ((x == 1) && (p > 0))

X==1	p>0	**COVERED BY TEST CASES #**
T	T	1
T	F	18
F	T	20
F	F	19

9) public final int cancel()
if ((x == 2) || (x == 3))

X==2	X==3	**COVERED BY TEST CASES #**
T	T	impossible
T	F	3
F	T	3
F	F	18

10) public final int dispose()
if ((x == 1))

X==1	**COVERED BY TEST CASES #**
T	1
F	19

ADDITIONAL TEST CASES:

TRANSITIONS COVERED	COVERED BY TEST CASES #
vending_machine(),set_price(25),insert_small_cups(2),coin(),large_cup(),tea(),cancel(), dispose().	**23**
vending_machine(),set_price(25),insert_large_cups(2),coin(),small_cup(),tea(),cancel(), dispose().	**24**
vending_machine(),set_price(25),coin(),large_cup(),tea(),cancel(),dispose().	**25**
vending_machine(),set_price(25),coin(),small_cup(),tea(),cancel(),dispose().	**26**
vending_machine(),set_price(25),insert_small_cups(1),coin(),large_cup(),tea(),cancel(), dispose().	**27**
vending_machine(),set_price(25),insert_small_cups(2),coin(),large_cup(),sugar(),tea(), cancel(),dispose().	**28**
vending_machine(),set_price(25),insert_large_cups(2),coin(),small_cup(),sugar(),tea(), cancel(),dispose().	**29**

4.OUTPUT OF THE TEST CASES:

TS.txt:

```
Test#1: insert_large_cups 1 insert_small_cups 1 insert_small_cups 1 set_price
25 set_price 50 coin coin small_cup small_cup sugar tea coin coin large_cup
large_cup sugar tea coin coin insert_large_cups 1 coin coin coin coin
large_cup tea insert_large_cups 1 dispose
Test#2: insert_small_cups 1 insert_large_cups 1 insert_large_cups 1 set_price
25 coin sugar large_cup large_cup tea set_price 75 coin coin coin coin
small_cup tea coin coin insert_small_cups 1 dispose
Test#3: set_price 25 insert_large_cups 1 coin cancel coin sugar coin sugar
coin large_cup tea coin insert_large_cups 1 coin sugar cancel dispose
Test#4: dispose
Test#5: set_price 50 insert_large_cups 1 coin insert_large_cups 1 coin
large_cup tea insert_small_cups 2 coin coin small_cup tea coin coin sugar
small_cup tea coin insert_small_cups 1 coin coin sugar large_cup tea
insert_large_cups 1 set_price 25 dispose
Test#6: set_price 25 insert_small_cups 2 coin small_cup coin tea
insert_large_cups 2 coin large_cup coin tea coin small_cup tea
insert_small_cups 1 coin sugar small_cup tea insert_small_cups 1 coin
large_cup coin tea insert_large_cups 1 insert_large_cups 1 dispose
Test#7: set_price 25 coin sugar cancel insert_large_cups 2 coin large_cup
sugar tea insert_small_cups 2 coin small_cup sugar tea coin small_cup sugar
tea insert_small_cups 1 set_price 25 coin sugar sugar small_cup large_cup
small_cup cancel dispose
Test#8: set_price 25 insert_small_cups 3 coin sugar small_cup coin small_cup
cancel coin sugar small_cup small_cup tea insert_large_cups 1 coin sugar
small_cup large_cup small_cup sugar tea coin large_cup cancel set_price 25
insert_small_cups 2 dispose
Test#9: set_price 25 insert_large_cups 3 coin sugar large_cup coin large_cup
cancel coin sugar large_cup large_cup tea insert_small_cups 1 coin sugar
large_cup small_cup large_cup sugar tea coin small_cup cancel set_price 25
insert_large_cups 2 dispose
Test#10: set_price 25 coin sugar coin coin cancel insert_small_cups 2 coin
sugar small_cup coin tea set_price 25 coin sugar small_cup coin tea
insert_small_cups 1 insert_large_cups 2 coin sugar large_cup coin tea coin
sugar large_cup coin tea insert_large_cups 1 insert_small_cups 1 dispose
Test#11: set_price 25 insert_large_cups 1 coin sugar sugar sugar sugar
large_cup sugar sugar tea insert_large_cups 1 insert_small_cups 1 coin
small_cup sugar sugar tea insert_small_cups 1 insert_small_cups 1 coin sugar
sugar cancel insert_small_cups 1 dispose
Test#12: set_price 25 insert_small_cups 1 coin coin sugar sugar small_cup
coin tea insert_small_cups 1 coin coin cancel insert_large_cups 1 dispose
Test#13: set_price 100 coin set_price 75 coin insert_small_cups 4 coin
small_cup tea set_price 25 coin small_cup tea insert_small_cups 1 coin
small_cup tea dispose
Test#14: set_price 25 insert_large_cups 4 coin large_cup tea
insert_large_cups 1 coin large_cup tea set_price 50 coin coin large_cup tea
coin coin large_cup tea dispose
Test#15: set_price 25 insert_large_cups 4 coin sugar large_cup tea
insert_large_cups 1 coin sugar large_cup tea set_price 50 coin coin sugar
large_cup tea coin coin large_cup tea dispose
Test#16: set_price 25 coin sugar cancel set_price 50 coin coin sugar cancel
coin coin cancel coin dispose
Test#17: set_price 25 insert_small_cups 5 coin sugar small_cup tea
insert_small_cups 1 coin sugar small_cup tea dispose
Test#18: coin insert_large_cups 0 insert_small_cups 0 set_price 0 cancel tea
small_cup large_cup sugar dispose
Test#19: set_price 25 coin insert_large_cups 0 insert_small_cups 0 set_price
0 tea dispose
Test#20: set_price 25 coin sugar insert_large_cups 0 insert_small_cups 0
set_price 10 tea dispose
```

```
Test#21: set_price 25 insert_small_cups 1 coin small_cup tea
insert_large_cups 0 insert_small_cups 0 set_price 0 cancel small_cup
large_cup sugar tea dispose
Test#22: set_price 25 insert_large_cups 1 coin large_cup tea
insert_large_cups 0 insert_small_cups 0 set_price 0 cancel small_cup
large_cup sugar tea dispose
Test#23: set_price 25 insert_small_cups 2 coin large_cup tea cancel dispose
Test#24: set_price 25 insert_large_cups 2 coin small_cup tea cancel dispose
Test#25: set_price 25 coin large_cup tea cancel dispose
Test#26: set_price 25 coin small_cup tea cancel dispose
Test#27: set_price 25 insert_small_cups 1 coin large_cup tea cancel dispose
Test#28: set_price 25 insert_small_cups 2 coin large_cup sugar tea cancel
dispose
Test#29: set_price 25 insert_large_cups 2 coin small_cup sugar tea cancel
dispose
$$ $$
```

Test#1:

```
Vending Machine
Initializing Vending Machine Object


       insert_large_cup() method
Enter Number of Large Cups: 1
price= 0
k= 1
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 0
k= 1
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 0
k= 1
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 25
price= 25
k= 1
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle
```

return=1   **PASS**

       set_price() method
Enter Price: 50
price= 50
k= 1
k1= 2
t=0
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 50
k= 1
k1= 2
t=25
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 50
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 50
k= 1
k1= 2
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 50
k= 1
k1= 2
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       sugar() method
price= 50
k= 1
k1= 2
t=0
x=3
s=2
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 50

k= 1
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 50
k= 1
k1= 1
t=25
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 50
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 50
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

       sugar() method
price= 50
k= 1
k1= 1
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 50
k= 0
k1= 1
t=0
x=5
s=1

```
Vending Machine  Current State=no large cups

return=1      PASS

       Coin() method
RETURN COIN
price= 50
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       Coin() method
RETURN COIN
price= 50
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 50
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       Coin() method
RETURN COIN
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted
```

```
return=1      PASS

       Coin() method
RETURN COIN
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 50
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 50
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 50
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 50
k= 1
k1= 1
t=0
x=6
s=1
return=1      PASS
```

Test#2:

```
Vending Machine
Initializing Vending Machine Object


       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 0
k= 0
k1= 1
```

```
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 0
k= 1
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 0
k= 2
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 25
price= 25
k= 2
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 2
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       sugar() method
price= 25
k= 2
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=1
```

```
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 75
price= 75
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 75
k= 1
k1= 1
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 75
k= 1
k1= 1
t=50
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 75
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
```

```
       Coin() method
RETURN COIN
price= 75
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 75
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 75
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       Coin() method
RETURN COIN
price= 75
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       Coin() method
RETURN COIN
price= 75
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 75
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
```

```
SHUT DOWN
price= 75
k= 1
k1= 1
t=0
x=6
s=2
return=1      PASS
```

Test#3:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS
Vending Machine  Current State= idle

price= 25
k= 1
k1= 0
t=0
x=1
s=0
       Coin() method
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
price= 25
```

```
k= 1
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 1
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
       Coin() method
RETURN COIN
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 1
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 0
t=0
x=5
```

```
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS
       Coin() method
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       cancel() method
RETURN COINS

return=1      PASS
Vending Machine  Current State= idle

price= 25
k= 1
k1= 0
t=0
x=1
s=0

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 0
t=0
x=6
s=0
return=1      PASS
```

Test#4:

```
Vending Machine
Initializing Vending Machine Object


       dispose() method
SHUT DOWN
price= 0
k= 0
k1= 0
t=0
```

```
x=6
s=0
return=1      PASS
```

Test#5:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 50
price= 50
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 50
k= 1
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 0
t=25
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 50
k= 2
k1= 0
t=25
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 50
k= 2
k1= 0
```

```
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 50
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 50
k= 1
k1= 2
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 2
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 50
k= 1
k1= 2
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 50
k= 1
k1= 1
t=0
x=1
s=2
```

```
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=25
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 50
k= 1
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 50
k= 1
k1= 1
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 50
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       Coin() method
RETURN COIN
price= 50
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       insert_small_Cup() method
```

```
Enter Number of Small Cups: 1
price= 50
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=25
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 50
k= 1
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 50
k= 1
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 50
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 50
k= 1
k1= 1
t=0
```

```
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 25
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 1
t=0
x=6
s=1
return=1      PASS
```

Test#6:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 0
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 0
```

k1= 2
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 2
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 0
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 2
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 2
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 2
k1= 1
t=0

```
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted
```

return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=3
s=0
Vending Machine Current State=Sugar

return=1   **PASS**

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=3
s=2
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine Current State=no small cups

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

       Coin() method
RETURN COIN

```
price= 25
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 2
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 2
k1= 1
t=0
x=6
s=1
return=1      PASS
```

Test#7:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle
```

return=1   **PASS**

       Coin() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine Current State=Sugar

return=1   **PASS**

       cancel() method
RETURN COINS
return=1   **PASS**
Vending Machine Current State= idle

price= 25
k= 0
k1= 0
t=0
x=1
s=0
       insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 2
k1= 0
t=0
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 25
k= 2
k1= 0
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 2
k1= 0
t=0

```
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 1
k1= 2
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 1
k1= 2
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 2
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle
```

return=1   **PASS**

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=3
s=2
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine Current State=no small cups

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       set_price() method
Enter Price: 25
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25

k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=3
s=0
Vending Machine Current State=Sugar
return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       large_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       cancel() method
RETURN COINS
return=1   **PASS**
Vending Machine Current State= idle

price= 25
k= 1
k1= 1
t=0
x=1
s=2

```
       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 1
t=0
x=6
s=2
return=1      PASS
```

Test#8:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 3
price= 25
k= 0
k1= 3
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 3
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 3
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS
```

```
       Coin() method
RETURN COIN
price= 25
k= 0
k1= 3
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS
Vending Machine  Current State= idle

price= 25
k= 0
k1= 3
t=0
x=1
s=2

       Coin() method
price= 25
k= 0
k1= 3
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 3
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
```

```
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 0
k1= 2
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 2
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 2
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 1
k1= 2
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 1
k1= 2
t=0
x=3
s=1
Vending Machine  Current State=Sugar
```

```
return=1      PASS

       small_cup() method
price= 25
k= 1
k1= 2
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
k1= 2
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 1
k1= 1
t=0
x=1
s=1
       set_price() method
Enter Price: 25
price= 25
```

```
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 1
k1= 3
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 3
t=0
x=6
s=1
return=1      PASS
```

Test#9:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 3
price= 25
k= 3
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 3
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
```

price= 25
k= 3
k1= 0
t=0
x=3
s=0
Vending Machine Current State=Sugar

return=1   **PASS**

       large_cup() method
price= 25
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       large_cup() method
price= 25
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       cancel() method
RETURN COINS
return=1   **PASS**

Vending Machine Current State= idle

price= 25
k= 3
k1= 0
t=0
x=1
s=1
       Coin() method
price= 25
k= 3
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 3
k1= 0
t=0
x=3

```
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 2
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 2
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 2
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 2
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS
```

```
       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 2
k1= 1
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
```

```
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 1
k1= 1
t=0
x=1
s=2
       set_price() method
Enter Price: 25
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 3
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 3
k1= 1
t=0
x=6
s=2
return=1      PASS
```

Test#10:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
```

```
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 0
t=0
x=1
s=0
       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 0
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 2
t=0
x=2
```

```
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 2
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 2
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 2
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 0
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
```

```
price= 25
k= 0
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 1
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 1
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 2
k1= 1
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
```

```
k= 2
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 2
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar
return=1      PASS

       large_cup() method
price= 25
k= 2
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 2
k1= 1
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar
```

return=1   **PASS**

       large_cup() method
price= 25
k= 1
k1= 1
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 1
k1= 1
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 0
k1= 1
t=0
x=5
s=1
Vending Machine Current State=no large cups

return=1   **PASS**

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine Current State= idle

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 2
t=0
x=1
s=1
Vending Machine Current State= idle

return=1   **PASS**

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 2
t=0
x=6
s=1
return=1   **PASS**

Test#11:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
price= 25
k= 1
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
```

```
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 1
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=1
Vending Machine  Current State= idle
```

return=1   **PASS**

       Coin() method
price= 25
k= 1
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       small_cup() method
price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=3
s=2
Vending Machine Current State=Sugar

return=1   **PASS**

price= 25
k= 1
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 1
k1= 0
t=0
x=4
s=2
Vending Machine Current State=no small cups

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25

```
k= 1
k1= 2
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 1
k1= 2
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 1
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 1
k1= 2
t=0
x=1
s=0
       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 1
k1= 3
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 3
t=0
x=6
s=0
```

return=1 **PASS**

Test#12:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 1
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted
```

return=1   **PASS**

       small_cup() method
price= 25
k= 0
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 1
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1   **PASS**

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine Current State=no small cups

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 1
t=0
x=1
s=2
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       Coin() method
RETURN COIN
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

```
       cancel() method
RETURN COINS
return=1
Vending Machine  Current State= idle

price= 25
k= 0
k1= 1
t=0
x=1
s=0
       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 1
k1= 1
t=0
x=6
s=0
return=1      PASS
```

Test#13:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 100
price= 100
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 100
k= 0
k1= 0
t=25
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 75
price= 75
k= 0
k1= 0
t=25
x=1
s=0
Vending Machine  Current State= idle
```

```
return=1      PASS

       Coin() method
price= 75
k= 0
k1= 0
t=50
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 4
price= 75
k= 0
k1= 4
t=50
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 75
k= 0
k1= 4
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 75
k= 0
k1= 4
t=0
x=2
s=2
Vending Machine  Current State=coins inserted
return=1
       Tea() method
DISPOSE SMALL CUP OF TEA
price= 75
k= 0
k1= 3
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 3
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
```

```
k= 0
k1= 3
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 0
k1= 2
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 3
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 3
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 3
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 0
k1= 2
t=0
```

```
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 2
t=0
x=6
s=2
return=1      PASS
```

Test#14:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 4
price= 25
k= 4
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 4
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 4
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
```

```
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 4
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 4
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 4
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 50
price= 50
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 3
k1= 0
t=25
```

```
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 3
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 50
k= 3
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 50
k= 2
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 2
k1= 0
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 50
k= 2
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS
```

```
       Tea() method
DISPOSE LARGE CUP OF TEA
price= 50
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 50
k= 1
k1= 0
t=0
x=6
s=1
return=1      PASS
```

Test#15:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 4
price= 25
k= 4
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 4
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 4
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar
```

return=1   **PASS**

       large_cup() method
price= 25
k= 4
k1= 0
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine Current State= idle

return=1   **PASS**

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 4
k1= 0
t=0
x=1
s=1
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 4
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

price= 25
k= 4
k1= 0
t=0
x=3
s=0
Vending Machine Current State=Sugar

return=1   **PASS**

       large_cup() method
price= 25
k= 4
k1= 0
t=0
x=3
s=1
Vending Machine Current State=Sugar

return=1   **PASS**

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 25

```
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       set_price() method
Enter Price: 50
price= 50
k= 3
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 3
k1= 0
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 3
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 50
k= 3
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       large_cup() method
price= 50
k= 3
k1= 0
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA WITH SUGAR
price= 50
k= 2
k1= 0
t=0
x=1
s=1
```

```
Vending Machine  Current State= idle

return=1 PASS

       Coin() method
price= 50
k= 2
k1= 0
t=25
x=1
s=1
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 50
k= 2
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 50
k= 1
k1= 0
t=0
x=1
s=1
Vending Machine  Current State= idle

return=1 PASS

       dispose() method
SHUT DOWN
price= 50
k= 1
k1= 0
t=0
x=6
s=1
return=1      PASS
```

Test#16:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
```

```
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 0
t=0
x=1
s=0
       set_price() method
Enter Price: 50
price= 50
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 0
k1= 0
t=25
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 50
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS
```

price= 50
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine Current State=Sugar

return=1   **PASS**

       cancel() method
RETURN COINS
return=1   **PASS**

Vending Machine Current State= idle

price= 50
k= 0
k1= 0
t=0
x=1
s=0

       Coin() method
price= 50
k= 0
k1= 0
t=25
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 50
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

       cancel() method
RETURN COINS
return=1   **PASS**

Vending Machine Current State= idle

price= 50
k= 0
k1= 0
t=0
x=1
s=0

       Coin() method
price= 50
k= 0
k1= 0
t=25
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       dispose() method
SHUT DOWN

```
price= 50
k= 0
k1= 0
t=25
x=6
s=0
return=1      PASS
```

Test#17:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 5
price= 25
k= 0
k1= 5
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 5
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 5
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 5
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
```

```
price= 25
k= 0
k1= 4
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 5
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 5
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 5
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

small_cup() method
price= 25
k= 0
k1= 5
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
DISPOSE SMALL CUP OF TEA WITH SUGAR
price= 25
k= 0
k1= 4
t=0
x=1
s=2
Vending Machine  Current State= idle

return=1      PASS

       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 4
t=0
```

```
x=6
s=2
return=1      PASS
```

Test#18:

```
Vending Machine
Initializing Vending Machine Object


       Coin() method
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       insert_large_cup() method
Enter Number of Large Cups: 0
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 0
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       set_price() method
Enter Price: 0
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       cancel() method
return=0      PASS

Vending Machine  Current State= idle

price= 0
k= 0
k1= 0
t=0
x=1
s=0
       Tea() method
price= 0
k= 0
```

```
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       small_cup() method
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle


return=0      PASS

       large_cup() method
price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

price= 0
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=0      PASS

       dispose() method
SHUT DOWN
price= 0
k= 0
k1= 0
t=0
x=6
s=0
return=1      PASS
```

Test#19:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
```

```
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 0
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=0      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 0
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=0      PASS

       set_price() method
Enter Price: 0
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=0      PASS

       Tea() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=0      PASS

       dispose() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=0      PASS
```

Test#20:

```
Vending Machine
Initializing Vending Machine Object
```

```
       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 0
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=0      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 0
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=0      PASS

       set_price() method
Enter Price: 10
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=0      PASS

       Tea() method
price= 25
k= 0
```

```
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=0      PASS

       dispose() method
price= 25
k= 0
k1= 0
t=0
x=3
s=0
Vending Machine  Current State=Sugar

return=0      PASS
```

Test#21:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 1
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 1
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS
```

```
       Tea() method
DISPOSE SMALL CUP OF TEA
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 0
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 0
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       set_price() method
Enter Price: 0
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       cancel() method

return=0      PASS

Vending Machine  Current State=no small cups
price= 25
k= 0
k1= 0
t=0
x=4
s=2

       small_cup() method
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       large_cup() method
```

```
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       Tea() method
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups

return=0      PASS

       dispose() method
price= 25
k= 0
k1= 0
t=0
x=4
s=2
Vending Machine  Current State=no small cups
return=0      PASS
```

Test#22:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 1
price= 25
k= 1
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
```

```
price= 25
k= 1
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 1
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
DISPOSE LARGE CUP OF TEA
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 0
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 0
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       set_price() method
Enter Price: 0
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       cancel() method
return=0
Vending Machine  Current State=no large cups
```

```
price= 25
k= 0
k1= 0
t=0
x=5
s=1

       small_cup() method
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       large_cup() method
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       Tea() method
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS

       dispose() method
price= 25
k= 0
k1= 0
t=0
x=5
s=1
Vending Machine  Current State=no large cups

return=0      PASS
```

Test#23:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
```

```
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 0
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle


return=1      PASS

       Coin() method
price= 25
k= 0
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 0
k1= 2
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
price= 25
k= 0
k1= 2
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=0      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 2
t=0
x=1
s=1

       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 2
t=0
```

x=6
s=1
return=1 **PASS**

Test#24:

Vending Machine
Initializing Vending Machine Object

set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine Current State= idle

return=1 **PASS**

insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 2
k1= 0
t=0
x=1
s=0
Vending Machine Current State= idle

return=1 **PASS**

Coin() method
price= 25
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1 **PASS**

small_cup() method
price= 25
k= 2
k1= 0
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=1 **PASS**

Tea() method
price= 25
k= 2
k1= 0
t=0
x=2
s=2
Vending Machine Current State=coins inserted

return=0 **PASS**

cancel() method
RETURN COINS

return=1 **PASS**

```
Vending Machine  Current State= idle

price= 25
k= 2
k1= 0
t=0
x=1
s=2

       dispose() method
SHUT DOWN
price= 25
k= 2
k1= 0
t=0
x=6
s=2
return=1      PASS
```

Test#25:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 0
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
price= 25
k= 0
k1= 0
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=0      PASS
```

```
       cancel() method
RETURN COINS

return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 0
t=0
x=1
s=1

       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 0
t=0
x=6
s=1
return=1      PASS
```

Test#26:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 0
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 0
k1= 0
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
price= 25
k= 0
k1= 0
t=0
x=2
s=2
Vending Machine  Current State=coins inserted
```

return=0   **PASS**

       cancel() method
RETURN COINS

return=1   **PASS**

Vending Machine Current State= idle

price= 25
k= 0
k1= 0
t=0
x=1
s=2
       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 0
t=0
x=6
s=2
return=1   **PASS**

Test#27:

Vending Machine
Initializing Vending Machine Object

       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       insert_small_Cup() method
Enter Number of Small Cups: 1
price= 25
k= 0
k1= 1
t=0
x=1
s=0
Vending Machine Current State= idle

return=1   **PASS**

       Coin() method
price= 25
k= 0
k1= 1
t=0
x=2
s=0
Vending Machine Current State=coins inserted

return=1   **PASS**

large_cup() method
price= 25
k= 0
k1= 1
t=0

```
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

       Tea() method
price= 25
k= 0
k1= 1
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=0      PASS

       cancel() method
RETURN COINS
return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 1
t=0
x=1
s=1

       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 1
t=0
x=6
s=1
return=1      PASS
```

Test#28:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_small_Cup() method
Enter Number of Small Cups: 2
price= 25
k= 0
k1= 2
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
```

```
k= 0
k1= 2
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       large_cup() method
price= 25
k= 0
k1= 2
t=0
x=2
s=1
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 0
k1= 2
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
price= 25
k= 0
k1= 2
t=0
x=3
s=1
Vending Machine  Current State=Sugar

return=0      PASS

       cancel() method
RETURN COINS

return=1      PASS

Vending Machine  Current State= idle

price= 25
k= 0
k1= 2
t=0
x=1
s=1
       dispose() method
SHUT DOWN
price= 25
k= 0
k1= 2
t=0
x=6
s=1
return=1      PASS
```

Test#29:

```
Vending Machine
Initializing Vending Machine Object


       set_price() method
Enter Price: 25
```

```
price= 25
k= 0
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       insert_large_cup() method
Enter Number of Large Cups: 2
price= 25
k= 2
k1= 0
t=0
x=1
s=0
Vending Machine  Current State= idle

return=1      PASS

       Coin() method
price= 25
k= 2
k1= 0
t=0
x=2
s=0
Vending Machine  Current State=coins inserted

return=1      PASS

       small_cup() method
price= 25
k= 2
k1= 0
t=0
x=2
s=2
Vending Machine  Current State=coins inserted

return=1      PASS

price= 25
k= 2
k1= 0
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=1      PASS

       Tea() method
price= 25
k= 2
k1= 0
t=0
x=3
s=2
Vending Machine  Current State=Sugar

return=0      PASS

       cancel() method
RETURN COINS

return=1      PASS

Vending Machine  Current State= idle
```

```
price= 25
k= 2
k1= 0
t=0
x=1
s=2

       dispose() method
SHUT DOWN
price= 25
k= 2
k1= 0
t=0
x=6
s=2
return=1      PASS
```

5.Conclusion:

My learning from this project is how easy a design model helps the developer to write a code and tester to test test-cases. Designing test case are interesting as it throws a challenge to tester to make sure that the code has no defects though it's just a tester's perception to say that code contains no defects after testing because there is no limit to design the number of test cases.

In the very beginning I have not touched the given class and understood the design of the given EFSM model then I have started editing the actual code for understanding the flow of the code by adding few lines to display the state before the operation returns a value to understand how operations are dependent on each other although the submitted source code is not modified. This step helped me to understand how various operations are interlinked. I have designed a test-driver in which I manually tested the given operations though it's a time-consuming process it allowed me to understand intuitively the structure of the code. Yes, automating the test driver with a input Test Suite file consumes less time to produce results but may not give the satisfaction to say the system is defect free. I felt that inputting the TS.txt file to the test driver and evaluating whether the test cases passed or failed in the test driver class would have been a better option rather than doing it manually.

6.Source Code:

VendingMachine.java

```
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;
	}
	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;
}
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;
}


public final void Variables()
{

        System.out.println("price= "+price);
        System.out.println("k= "+k);
        System.out.println("k1= "+k1);
        System.out.println("t="+t);
        System.out.println("x="+x);
        System.out.println("s="+s);

}


public final void CurrentState()
{
        if(x==1)
        {System.out.println("Vending Machine  Current State= idle\n");

        }

        if(x==3)
        {System.out.println("Vending Machine  Current State= Sugar");

        }
        if(x==2)
        {

                System.out.println("Vending Machine  Current State= coins inserted");
        }
```

```
                if(x==5)
                {

                        System.out.println("Vending Machine  Current State= no large cups");
                }

                if(x==4)
                {

                        System.out.println("Vending Machine  Current State= no small cups");
                }


        }
}
```

TestDriver.java

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.file.Files;
import java.util.Scanner;

public class TestDriver
{

	public static void main(String[] args) throws IOException
	{
		// TODO Auto-generated method stub
		System.out.println("Driver for Vending Machine");
		VendingMachine vendingmachine = new VendingMachine();
		Scanner values;
		String value=null;
		//char choice;
		int i=1,m=0,temp=1,returnValue;
		while(temp==1)
		{

			System.out.println("0. coin()");
			System.out.println("1. small_cup()");
			System.out.println("2. large_cup()");
			System.out.println("3. sugar()");
			System.out.println("4. tea()");
			System.out.println("5. insert_large_cups(int n)");
			System.out.println("6. insert_small_cups(int)");
			System.out.println("7. set_price(int p)");
			System.out.println("8. cancel()");
			System.out.println("9. dispose()");
			System.out.println("Testing-related methods");
			System.out.println("a. Show_variables()");
			System.out.println("q. Quit Vending Machine class driver");

			Scanner keyboard = new Scanner(System.in);
			String choice= keyboard.nextLine();


			switch (choice)
			{
				case "0": // coin
```

```
                    System.out.print("\tcoin selected\n");
                    returnValue = vendingmachine.coin();
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
                    System.out.println("The value returned by the
method:"+returnValue);
                    break;

                case "4": // tea
                    System.out.print("\ttea\n");
                    returnValue = vendingmachine.tea();
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
                    System.out.println("The value returned by the
method:"+returnValue);
                    //                          Sugar=1;
                    break;
                case "3": // sugar
                    returnValue = vendingmachine.sugar();
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
                    System.out.println("The value returned by the
method:"+returnValue);
                    break;
                case "1": // small cup
                    System.out.print("Small Cups Selected\n");
                    returnValue = vendingmachine.small_cup();
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
                    System.out.println("The value returned by the
method:"+returnValue);
                    break;
                case "2": // Large Cup
                    System.out.print("large Cups Selected \n");
                    returnValue = vendingmachine.large_cup();
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
                    System.out.println("The value returned by the
method:"+returnValue);
                    break;
                case "5": // Insert Large Cup
                    System.out.print("\tinsert large cup\n");
                     values = new Scanner(System.in);
                     value= values.nextLine();
                    System.out.println("Enter value  of parameter k: "+value);
                    returnValue =
vendingmachine.insert_large_cups(Integer.parseInt(value));
                    vendingmachine.Variables();
                    vendingmachine.CurrentState();
```

```
                                        System.out.println("The value returned by the
method:"+returnValue);
                                        break;
                                case "6": // Insert small Cup
                                        System.out.print("\tinsert small cup\n");
                                        values = new Scanner(System.in);
                                         value= values.nextLine();
                                        System.out.println("Enter value  of parameter k1:
"+value);
                                        returnValue =
vendingmachine.insert_small_cups(Integer.parseInt(value));
                                        vendingmachine.Variables();
                                        vendingmachine.CurrentState();
                                        System.out.println("The value returned by the
method:"+returnValue);
                                        break;
                                case "7": // Set price
                                        System.out.print("\tset price\n");
                                        values = new Scanner(System.in);
                                        value= values.nextLine();
                                        System.out.println("Enter Price: "+value);
                                        returnValue =
vendingmachine.set_price(Integer.parseInt(value));
                                        vendingmachine.Variables();
                                        vendingmachine.CurrentState();
                                        System.out.println("The value returned by the
method:"+returnValue);
                                        break;
                                case "8": // cancel
                                        System.out.print("\tcancel\n");
                                        returnValue = vendingmachine.cancel();
                                        vendingmachine.Variables();
                                        vendingmachine.CurrentState();
                                        System.out.println("The value returned by the
method:"+returnValue);
                                        break;

                                case "9": //Set dispose
                                        System.out.print("dispose\n");
                                        returnValue=vendingmachine.dispose();
                                        vendingmachine.Variables();
                                        vendingmachine.CurrentState();
                                        vendingmachine = new VendingMachine();
                                        break;
                                case "a": //show variables
                                        System.out.print("show_variables() testing-related
method\n");
                                        vendingmachine.Variables();
                                        vendingmachine.CurrentState();
```

```
                                        break;

                                case "q":
                                        System.exit(0);

                                default:System.out.println("Invalid Input");
                                break;
                        }
                }

        }
}
```