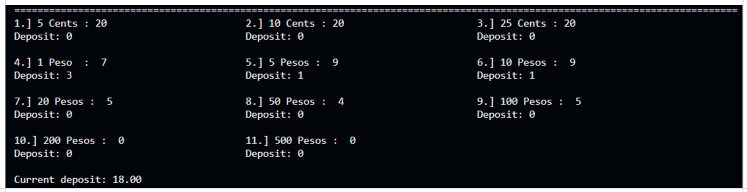
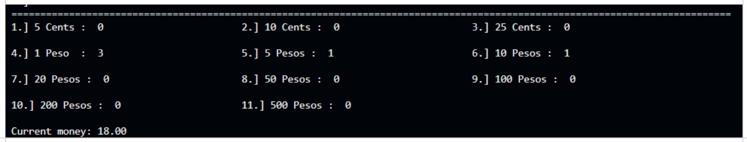
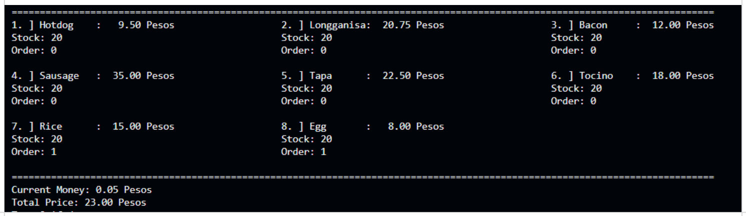
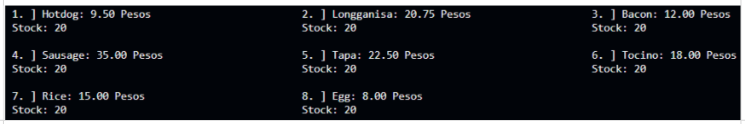
Function	#	Description	Sample Input	Expected result	Actual Result	Pass/Fail
displayDenominations()	1	Check the display when i > 0 && i < 4	i = 3	3.] 25 Cents: 0	3.] 25 Cents: 0	Pass
displayDenominations()	2	Check the display when i == 4	i = 4	4.] 1 Peso: 0	4.] 1 Peso: 0	Pass
displayDenominations()	3	Check the display when i > 4 && i < 12	i = 5	5.] 5 Pesos: 0	5.] 5 Pesos: 0	Pass

displayDenominations()	4	Check display when nMode == 1	nMode = 1
			the function is called as:
			displayDenominations(aDenomCashier, aDenomUser, fMoney, 1);
				Current denominations and deposits are displayed	
					`==========================================================================================`
					`1.] 5 Cents : 20          2.] 10 Cents : 20          3.] 25 Cents : 20`
					`Deposit: 0                Deposit: 0                 Deposit: 0`
					
					`4.] 1 Peso  :  7          5.] 5 Pesos :  9           6.] 10 Pesos :  9`
					`Deposit: 3                Deposit: 1                 Deposit: 1`
					
					`7.] 20 Pesos :  5         8.] 50 Pesos :  4          9.] 100 Pesos :  5`
					`Deposit: 0                Deposit: 0                 Deposit: 0`
					
					`10.] 200 Pesos :  0       11.] 500 Pesos :  0`
					`Deposit: 0                Deposit: 0`
					
					`Current deposit: 18.00`
						Pass

displayDenominations()	5	Check display when nMode == 0	nMode = 0
			the function is called as:
			displayDenominations(aDenomUser, aDenomCashier, fMoney, 0);
				Current denominations and money are displayed	
					`==========================================================================================`
					`1.] 5 Cents :  0          2.] 10 Cents :  0          3.] 25 Cents :  0`
					
					`4.] 1 Peso  :  3          5.] 5 Pesos :  1           6.] 10 Pesos :  1`
					
					`7.] 20 Pesos :  0         8.] 50 Pesos :  0          9.] 100 Pesos :  0`
					
					`10.] 200 Pesos :  0       11.] 500 Pesos :  0`
					
					`Current money: 18.00`
						Pass

getDenomination()	6	Check the result when nSelection and nMultiplier is a whole number	nSelection = 1 nMultiplier = 2	2 5-Cent coins 0.10 Pesos	1.] 5 Cents = 2 2.] 10 Cents = 0 3.] 25 Cents = 0 4.] 1 Peso = 0 5.] 5 Pesos = 0 6.] 10 Pesos = 0 7.] 20 Pesos = 0 8.] 50 Pesos = 0 9.] 100 Pesos = 0 10.] 200 Pesos = 0 11.] 500 Pesos = 0 Current money: 0.10 Pesos	Pass
getDenomination()	7	Check the result when nDenom & nMultiplier have the same int	nSelection = 1 nMultiplier = 1	1 5-Cent coin 0.05 Pesos	1.] 5 Cents = 1 2.] 10 Cents = 0 3.] 25 Cents = 0 4.] 1 Peso = 0 5.] 5 Pesos = 0 6.] 10 Pesos = 0 7.] 20 Pesos = 0 8.] 50 Pesos = 0 9.] 100 Pesos = 0 10.] 200 Pesos = 0 11.] 500 Pesos = 0 Current money: 0.05 Pesos	Pass
getDenomination()	8	Check the result when there are already existing denominations	Existing denominations 5 1-Peso Coins 1 5-Peso Coin nSelection = 6 nMultiplier = 1	5 1-Peso Coins 1 5-Peso Coin 1 10-Peso Coin 20.00 Pesos	1.] 5 Cents: 0 2.] 10 Cents: 0 3.] 25 Cents: 0 4.] 1 Peso: 5 5.] 5 Pesos: 1 6.] 10 Pesos: 1 7.] 20 Pesos: 0 8.] 50 Pesos: 0 9.] 100 Pesos: 0 10.] 200 Pesos: 0 11.] 500 Pesos: 0 Current money: 20.00 Pesos Type 0 if done	Pass
acceptDenominations()	9	Check the result when the function runs once	nSelection = 6 nMultiplier = 1	1 10-Peso Coin 10.00 Pesos	1.] 5 Cents: 0 2.] 10 Cents: 0 3.] 25 Cents: 0 4.] 1 Peso: 0 5.] 5 Pesos: 0 6.] 10 Pesos: 1 7.] 20 Pesos: 0 8.] 50 Pesos: 0 9.] 100 Pesos: 0 10.] 200 Pesos: 0 11.] 500 Pesos: 0 Current money: 10.00 Pesos	Pass

acceptDenominations()	10	Check the result when the function runs more than once	Cycle 1: nDenom = 6, nMultiplier = 3 Cycle 2: nDenom = 7, nMultiplier = 1 Cycle 3: nDenom = 8, nMultiplier = 1	3 10-Peso Coins 1 20-Peso Bill 1 50-Peso Bill 100.00 Pesos	1.] 5 Cents: 0 2.] 10 Cents: 0 3.] 25 Cents: 0 4.] 1 Peso: 0 5.] 5 Pesos: 0 6.] 10 Pesos: 3 7.] 20 Pesos: 1 8.] 50 Pesos: 1 9.] 100 Pesos: 0 10.] 200 Pesos: 0 11.] 500 Pesos: 0 Current money: 100.00 Pesos	Pass
acceptDenominations()	11	Check the result when either nDenom or nMultiplier have invalid values	nSelection = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass
displayItems()	12	Check the order of the items if it is displayed correctly	No input	Display items by threes, ordered from left to right, top to bottom Hotdog, Longganisa, Bacon Sausage, Tapa, Tocino Rice, Egg	1.] Hotdog: 9.50 Pesos Stock: 20 Order: 0 · 2.] Longganisa: 20.75 Pesos Stock: 20 Order: 0 · 3.] Bacon: 12.00 Pesos Stock: 20 Order: 0 4.] Sausage: 35.00 Pesos Stock: 20 Order: 0 · 5.] Tapa: 22.50 Pesos Stock: 20 Order: 0 · 6.] Tocino: 18.00 Pesos Stock: 20 Order: 0 7.] Rice: 15.00 Pesos Stock: 20 Order: 0 · 8.] Egg: 8.00 Pesos Stock: 20 Order: 0	Pass
displayItems()	13	Check the display if nMode = 1	nMode = 1	Orders, Current money and Total Price are displayed		Pass
displayItems()	14	Check the display if nMode != 1	nMode = 0	Orders, Current money and Total Price are not displayed		Pass
checkOrder()	15	check result when nOrder is consistently 0 and rice and egg is default to 1	int aOrder[9] = {0, 0, 0, 0, 0, 0, 0, 1, 1};	0	Are there orders: 0	Pass
checkOrder()	16	check result when nOrder is consistently non-zero except for rice and egg	int aOrder[9] = {0, 1, 3, 3, 4, 2, 1, 2, 1};	1	Are there orders: 1	Pass
checkOrder()	17	check result when nOrder has zeros and non-zeros	int aOrder[9] = {0, 1, 3, 0, 4, 0, 0, 1, 1};	1	Are there orders: 1	Pass
checkOrder()	18	check result when the user ordered either rice or egg only	aOrder[8] = 2;	1	Are there orders: 1	Pass
countMoney()	19	check result if it does not have a decimal place	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	968	968	Pass
countMoney()	20	check result if it has a decimal place	int aDenomCashier[] = {0, 20, 24, 22, 10, 0, 0, 0, 0, 0, 0};	18.9	18.9	Pass
countMoney()	21	check result if it is less than 1 peso	aDenomCashier[] = {0, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0};	0.65	0.65	Pass

giveDenominations()	22	check result if fMoney == fChange	int aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};
			aDenomUser[12] = 0;
			
			fTPrice = 18.00;
			fMoney = 18.00;
				no denominations	no denominations	Pass

giveDenominations()	23	check result if fMoney >= fChange and can be given 1 whole denomination	int aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};
			aDenomUser[12] = 0;
			
			fMoney = 100.00;
			fTPrice = 50.00;
				1 50-Peso bill	1.] 50 Pesos: 1	Pass

```
==============================================================================================
1. ] Hotdog     :   9.50 Pesos                2. ] Longganisa:  20.75 Pesos                3. ] Bacon     :  12.00 Pesos
Stock: 20                                     Stock: 20                                    Stock: 20
Order: 0                                      Order: 0                                     Order: 0

4. ] Sausage    :  35.00 Pesos                5. ] Tapa      :  22.50 Pesos                6. ] Tocino    :  18.00 Pesos
Stock: 20                                     Stock: 20                                    Stock: 20
Order: 0                                      Order: 0                                     Order: 0

7. ] Rice       :  15.00 Pesos                8. ] Egg       :   8.00 Pesos
Stock: 20                                     Stock: 20
Order: 1                                      Order: 1

==============================================================================================
Current Money: 0.05 Pesos
Total Price: 23.00 Pesos
```

```
1. ] Hotdog: 9.50 Pesos                 2. ] Longganisa: 20.75 Pesos               3. ] Bacon: 12.00 Pesos
Stock: 20                               Stock: 20                                  Stock: 20

4. ] Sausage: 35.00 Pesos               5. ] Tapa: 22.50 Pesos                     6. ] Tocino: 18.00 Pesos
Stock: 20                               Stock: 20                                  Stock: 20

7. ] Rice: 15.00 Pesos                  8. ] Egg: 8.00 Pesos
Stock: 20                               Stock: 20
```

giveDenominations()	24	check result if fMoney >= fChange but has to give smaller denominations	int aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};
			aDenomUser[12] = 0;
			
			fMoney = 600.00;
			fTPrice = 0.00;
				2 50-peso bills
				5 100-peso bills
					1.] 50 Pesos: 2
					2.] 100 Pesos: 5
						Pass

giveDenominations()	25	check result if user inserted denominations	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0}; aDenomUser[11] = 2 fMoney = 1000 fPrice = 18	1.] 1 Peso: 2 2.] 10 Pesos: 1 3.] 20 Pesos: 1 4.] 50 Pesos: 1 5.] 100 Pesos: 4 6.] 500 Pesos: 1	1.] 1 Peso: 2 2.] 10 Pesos: 1 3.] 20 Pesos: 1 4.] 50 Pesos: 1 5.] 100 Pesos: 4 6.] 500 Pesos: 1	Pass

getOrder()	26	check flow if nOrder <= nStock	aStock[1] = 20;
			nSelection = 1
			fMoney = 1000;
				1.] Hotdog
				20 stocks
				1 order
					1.] Hotdog
					20 stocks
					1 order
						Pass

getOrder()	27	check flow if nOrder > nStock	aStock[1] = 0;
			nSelection = 0;
			fMoney = 1000;
				Out of stock	ERROR: Out of Stock! Sorry!	Pass

getOrder()	28	check flow if fMoney < fTotalPrice and decided to put more denominations	aStock[1] = 20;
			nSelection = 1
			fMoney = 0;
				The given money is insufficient.
				add more money
				if fMoney >= fTotalPrice
				
				1.] Hotdog
				20 stocks
				1 order
					The given money is insufficient. Would you like to add more money to the machine? 1
					Please enter a denomination: 9
					Please enter the amount of the denomination: 1
					
					1.] Hotdog
					20 stocks
					1 order
						Pass

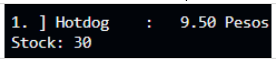
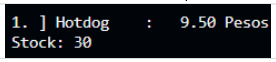
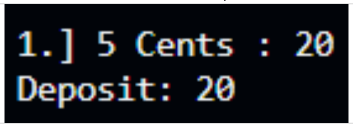
getOrder()	29	check flow if fMoney < fTotalPrice and decided to cancel the item	aStock[1] = 20;
			nSelection = 1
			fMoney = 0;
				1.] Hotdog
				20 stocks
				0 order
					1.] Hotdog
					20 stocks
					0 order
						Pass

getOrder()	30	check flow if fMoney < fTotalPrice and decided put invalid input	aStock[1] = 20;
			nSelection = 1
			fMoney = 0;
			
			nMultiplier = -1
				ERROR: Invalid Input	ERROR: Invalid Input	Pass

getOrder()	31	check flow if user decided to cancel order	aStock[1] = 20;
			nSelection = 0
			fMoney = 0;
				Order canceled.	Order canceled.	Pass

getOrder()	32	check flow if user decided to confirm order and the change can be given in one type of denomination	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0}; aOrder[1] = 2 aOrder[6] = 1 aOrder[7] = 1 aOrder[8] = 1 fMoney = 100	fTotalPrice = 60 100 - 60 = 40 2 20-peso bills	Recipt: 1.] Hotdog: 2 orders Price: 19.00 2.] Tocino: 1 orders Price: 18.00 3.] Rice: 1 orders Price: 15.00 4.] Egg: 1 orders Price: 8.00 Current Money: 100.00 Pesos Total Price: 60.00 Pesos Money: 100.00 Price: 60.00 Change: 40.00 Denominations given: 1.] 20 Pesos: 2	Pass

getOrder()	33	check flow if user decided to confirm order and the change has to be broken down into smaller denominations	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0}; aOrder[1] = 2 aOrder[7] = 1 aOrder[8] = 1 fMoney = 100	fTotalPrice = 42 100 - 42 = 58 1 50-peso bill 1 5-peso coin 3 1-peso coins	Recipt: 1.] Hotdog: 2 orders Price: 19.00 2.] Rice: 1 orders Price: 15.00 3.] Egg: 1 orders Price: 8.00 Current Money: 100.00 Pesos Total Price: 42.00 Pesos Money: 100.00 Price: 42.00 Change: 58.00 Denominations given: 1.] 1 Peso: 3 2.] 5 Pesos: 1 3.] 50 Pesos: 1 Given change: 58.00	Pass
getOrder()	34	check flow if user decided to continue ordering	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0}; aOrder[1] = 2 aOrder[7] = 1 aOrder[8] = 1 fMoney = 100	loops back to displayItems()	loops back to displayItems()	Pass
getOrder()	35	check flow if user decided to confirm order but entered an invalid input	aDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0}; aOrder[1] = 2 aOrder[7] = 1 aOrder[8] = 1 fMoney = 100	invalid input loops back to confirming order	ERROR: Invalid Input Confirm order? Press 0 to cancel the order Press 1 to confirm the order Press 2 to continue ordering	Pass
vendingFeatures()	36	check flow if fMoney > 0	aDenomUser[9] = 1	breaks the while condition to go to getOrder()	breaks the while condition to go to getOrder()	Pass
vendingFeatures()	37	check flow if fMoney == 0 and decided to continue order	aDenomUser[] = 0	ask first for denomination user inputs 0 ask if they still want to order user inputs 1 loops back to asking denomination	ask first for denomination user inputs 0 ask if they still want to order user inputs 1 loops back to asking denomination	Pass
vendingFeatures()	38	check flow if fMoney == 0 and decided to cancel order	aDenomUser[] = 0	ask first for denomination user inputs 0 ask if they still want to order user inputs 0 breaks loop	ask first for denomination user inputs 0 ask if they still want to order user inputs 0 breaks loop	Pass
vendingFeatures()	39	check flow if fMoney == 0 and decided put invalid input	aDenomUser[] = 0	ask first for denomination user inputs 0 ask if they still want to order user inputs -1 gives error message and loops back to asking if they want to order	ERROR: Invalid Input No denominations placed, would you like to cancel the order? Press 1 to continue the order Press 0 to cancel the order	Pass
maintenanceFeatures()	40	if user picks 1	nMenu = 1	moves to invenrtoryFeatures()	1. View Inventory 2. Modify Price 3. Restock Inventory 4. Back to Maintenance Features	Pass
maintenanceFeatures()	41	if user picks 2	nMenu = 2	moves to cashRegisterFeatures()	1. View Cash Register 2. Restock Cash Register 3. Cash Out 4. Back to Maintenance Features	Pass
maintenanceFeatures()	42	if user picks 3	nMenu = 3	breaks loop	breaks loop	Pass
maintenanceFeatures()	43	if user picks invalid number	nMenu = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass
inventoryFeatures()	44	if user picks 1	nMenu = 1	displayItems	displayItems	Pass
inventoryFeatures()	45	if user picks 2	nMenu = 2	changePrice	changePrice	Pass
inventoryFeatures()	46	if user picks 3	nMenu = 3	restockInventory	restockInventory	Pass
inventoryFeatures()	47	if user picks 4	nMenu = 4	breaks loop	breaks loop	Pass
inventoryFeatures()	48	if user picks invalid number	nMenu = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass
changePrice()	49	if fMoney is non-negative and non-zero	nSelection = 1 fMoney = 10	aItemPrices[1] = 9.5 Hotdog = 10 Pesos	1.] Hotdog : 10.00 Pesos	Pass
changePrice()	50	if fMoney is zero	nSelection = 1 fMoney = 0	Price value of zero detected. Price changed cancelled	Price was set to 0. Price change canceled.	Pass

changePrice()	51	if fMoney is negative	nSelection = 1 fMoney = -10	invalid input	ERROR: Invalid Input	Pass
changePrice()	52	if nSelection is -1	nSelection = -1	reset to default prices	Item prices resetted	Pass
changePrice()	53	if nSelection is less than -1	nSelection = -2	invalid input	ERROR: Invalid Input	Pass
restockInventory()	54	if restock number is positive int	nSelection = 1 nRestock = 10 aStock[1] = 20	Hotdog stocks = 30	 1.] Hotdog : 9.50 Pesos Stock: 30	Pass
restockInventory()	55	if restock number is negative	nSelection = 1 nRestock = -10 aStock[1] = 20	invalid input	ERROR: Invalid Input	Pass
restockInventory()	56	if restock number is zero	nSelection = 1 nRestock = 0 aStock[1] = 30	Hotdog stocks = 30	 1.] Hotdog : 9.50 Pesos Stock: 30	Pass
cashRegisterFeatures()	57	if user picks 1	nMenu = 1	displayDenominations	displayDenominations	Pass
cashRegisterFeatures()	58	if user picks 2	nMenu = 2	acceptDenominations	acceptDenominations	Pass
cashRegisterFeatures()	59	if user picks 3	nMenu = 3	cashOut	cashOut	Pass
cashRegisterFeatures()	60	if user picks 4	nMenu = 4	breaks loop	breaks loop	Pass
cashRegisterFeatures()	61	if user picks invalid number	nMenu = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass
cashOut()	62	if user picks 1	nMenu = 1	denominationCashOut()	denominationCashOut()	Pass
cashOut()	63	if user picks 2	nMenu = 2	valueCashOut()	valueCashOut()	Pass
cashOut()	64	if user picks 3	nMenu = 3	breaks loop	breaks loop	
cashOut()	65	if user picks invalid number	nMenu = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass
denominationCashOut()	66	if the user decides to pick a denomination that has a non-zero stock and the user decided to want to deposit denominations less than or equal to the current denomination stock	aDenomCashier[1] = 20 nSelection = 1 nMultiplier = 20	20 25-cent coins deposit	 1.] 5 Cents : 20 Deposit: 20	Pass
denominationCashOut()	67	if the user decides to pick a denomination that has a non-zero stock and the user decided to want to deposit denominations more than the current denomination stock	aDenomCashier[2] = 20 nSelection = 2 nMultiplier = 30	Error: not enough denominations	Error: not enough denominations	Pass
denominationCashOut()	68	if the user decides to pick a denomination that has zero stock	aDenomCashier[11] = 0 nSelection = 11 nMultiplier = 1	Error: not enough denominations	Error: not enough denominations	Pass
denominationCashOut()	69	if the user picks an invalid denomination	nSelection = -10	ERROR: Invalid Input	ERROR: Invalid Input	Pass
denominationCashOut()	70	if the user picks an invalid multiplier	aDenomCashier[2] = 20 nSelection = 2 nMultiplier = -10	ERROR: Invalid Input	ERROR: Invalid Input	Pass
denominationCashOut()	71	if the user decides to have nSelection = 0 after picking valid denomination deposits	aDenomCashOut[1] = 20 aDenomCashier[1] = 20	1.] 5 cents: 20 Deposited cash: 1 peso	Deposited denominations: 1.] 5 Cents: 20 Deposited cash: 1.00	Pass
denominationCashOut()	72	if the user decides to have nSelection = 0 without depositing anything	nSelection = 0	No denominations inputted. Cash out canceled.	No denominations inputted. Cash out canceled.	Pass
denominationCashOut()	73	if the user decides to cancel depositing denominations	aDenomCashOut[1] = 20 nSelection = -1	Cash out canceled	Cash out canceled	Pass
valueCashOut()	74	if the value can be deposited by atleast the smallest denomination unit	fMoney = 0.05 nDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	1 5-cent coin 0.05 pesos deposited	Denominations given: 1.] 5 Cents: 1 Total deposit: 0.05	Pass
valueCashOut()	75	if the value is less than the smallest denomination unit	fMoney = 0.04 nDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	ERROR: Not dispensable by denominations	ERROR: Not dispensable by denominations	Pass
valueCashOut()	76	if the value's decimal places is not divisible by the cent values	fMoney = 7.27 nDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	5-peso coins = 1 1-peso coins = 2 25-cents coins = 1	Denominations given: 1.] 25 Cents: 1 2.] 1 Peso: 2 3.] 5 Pesos: 1 Total deposit: 7.25 Warning: Unable to dispense exact stated amount.	Pass
valueCashOut()	77	if the value is larger than the current total money	fMoney = 1000 nDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	ERROR: Not enough money	ERROR: Not enough money	Pass

valueCashOut()	78	if the value is invalid	fMoney = -100 nDenomCashier[] = {0, 20, 20, 20, 10, 10, 10, 5, 4, 5, 0, 0};	ERROR: Invalid Input	ERROR: Invalid Input	Pass
main()	79	if nMenu = 1	nMenu = 1	vendingFeatures()	vendingFeatures()	Pass
main()	80	if nMenu = 2 and the password is correct	nMenu = 2 nPassword = 123456	Welcome, User!	Welcome, User!	Pass
main()	81	if nMenu = 2 and the password is wrong	nMenu = 2 nPassword = 123455	ERROR: Incorrect password	ERROR: Incorrect password	Pass
main()	82	if nMenu = 3 and the password is correct	nMenu = 3 nPassword = 123456	Shutting down...	Shutting down...	Pass
main()	83	if nMenu = 3 and the password is wrong	nMenu = 3 nPassword = 123455	ERROR: Incorrect password	ERROR: Incorrect password	Pass
main()	84	if nMenu is an invalid value	nMenu = -1	ERROR: Invalid Input	ERROR: Invalid Input	Pass