/* 12037893 Griffin(S12A)

Oil-Bularyo_Functions_Test Cases*/

Function: void changeTextColor(int nColor)

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Text will be in Green	2	Test!	Test!	Pass
2	Text will be in Magenta	5	Test!	Test!	Pass

Function: void displayGameTitle()

		Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
:	1	Prints the Game Title	None	Formatted Game Title	10 10 10 10 10 10 10 10	Pass

Function: void displayMainMenu()

	Test Description	Input value/	Expected	Actual output/result	Pass/Fail
		parameters	output/result		
1	Prints the Main Menu	None	Formatted Main Menu	in the man the	Pass

Function: void displayActionsMenu()

	Test Description	Input value/	Expected	Actual output/result	Pass/Fail
1	Prints the Actions Menu	None	output/result Formatted Actions Menu	ACTIONS MEMU	Pass

Function: void displayBuyMenu()

	Test	Input value/	Expected output/result	Actual output/result	Pass/Fail
	Description	parameters			
1	Prints the	None			Pass
	Buy Menu		BUY MENU	BUY MENU	
				Press [1] Buy Essential Oil/s	
			Press [1] Buy Essential Oil/s	Press [0] Go back to Actions Menu	
			Press [0] Go back to Actions Menu		

Function: int getPriceRangePerBottleEOStep1(int nLower, int nUpper, int nCount)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	"Price generated is	nLower = 100	100 - 150	112	Pass
	within price range per	nUpper = 150			
	bottle of:" Lemon	nCount = 1			
2	" "Lavender	nLower = 20	20 - 60	35	Pass
		nUpper = 60			
		nCount = 1			
3	" "Rosemary	nLower = 70	70 - 100	81	Pass
		nUpper = 100			
		nCount = 1			
4	" " Mint	nLower = 130	130 - 200	148	Pass
		nUpper = 200			
		nCount = 1			

Function: double getOperator()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	random number	None	random number (double)	0.416089	Pass
	between 0 and 1		between 0 and 1		
	(double)				

Function: int getFactor(int nLower, int nUpper, int nCount)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Random number	nLower = 102	random number between	196	Pass
	between 101-200	nUpper = 199	101-200		
		nCount = 1			
2	Random number	nLower = 2	random number between	20	Pass
	between 1-50	nUpper = 49	1-50		
		nCount = 1			

Function: int getPricePerBottleEOStep2(int nLower, int nUpper, int nCount)

Let nPriceBottleStep1 = getPriceRangePerBottleEOStep1(int nLower, int nUpper, int nCount)

Let nRandomFactor = getFactor(int nLower, int nUpper, int nCount)

To test this function, nPriceBottleStep1 = 108, nRandomFactor = 18, and nRandomFactor = 180 was used

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	nPriceBottleStep1 is	nLower = 100	108	108	Pass
	retained	nUpper = 150			
		nCount = 1			
2	nPriceBottleStep1 is	nLower = 100	126	126	
	added to	nUpper = 150			
	nRandomFactor(18)	nCount = 1			
3	nRandomFactor(18) is	nLower = 100	90	90	
	subtracted from	nUpper = 150			
	nPriceBottleStep1	nCount = 1			
4	nPriceBottleStep1 is	nLower = 100	19440	19440	
	multiplied to	nUpper = 150			
	nRandomFactor(180)	nCount = 1			

Function: int getNoEOBottles(int nLowerNoBottle, int nUpperNoBottle, int nCountNoBottle)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	No. of EO Bottles that the	nLowerNoBottle = 1	1 - 10	8	Pass
	Player can Buy in a City is	nUpperNoBottle = 10			
	from 1 – 10 EO Bottle/s.	nCountNoBottle = 1			

Function: void displayMenuBuyEO(int nPriceBottleLemon, int nPriceBottleLavender, int nPriceBottleRosemary, int nPriceBottleMint, int

nNoBottleLemon, int nNoBottleLavender, int nNoBottleRosemary, int nNoBottleMint)

	Test Description	Input value/ parameters	Expected	Actual output/result	Pass/Fail
			output/result		
1	Prints the Buy Essential	displayMenuBuyEO(118,	Formatted Buy		Pass
	Oils Menu	228, 338, 448, 18, 28, 108, 118)	Essential Oils Menu	BUY ESSENTIAL OILS MENU Press EO Available Bottles Price/Bottle [1] Lemon 18 118 [2] Lavender 28 228 [3] Rosemary 108 338 [4] Mint 118 448	
				[0] Go back to Actions Menu	

Function: void displayBuyEO()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Buy EO confirmation	None	Formatted Buy EO confirmation	Press [0] Buy Essential Oil/s Press [0] Don't execute transaction. Go back to Actions Menu	Pass

Function: int isBuyEOValid(int nCash, int nBuyEOPrice)

This function was tested using printf().

Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
Player has enough	nCash = 1500	1	1	Pass
nCash	nBuyEOPrice = 1409			
Player does not have	nCash = 1500	0	0	Pass
enough nCash	nBuyEOPrice = 2280			

Function: void sellPriceChecker(int *nSellPriceLemon, int *nSellPriceLavender, int *nSellPriceRosemary, int *nSellPriceMint)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/ Fail
1	nSellPriceLemon	*nSellPriceLemon = -100	nSellPriceLemon = 0	nSellPriceLemon = 0	Pass
	will be updated	*nSellPriceLavender = 18	nSellPriceLavender = 18	nSellPriceLavender = 18	
	because price	*nSellPriceRosemary = 2	nSellPriceRosemary = 2	nSellPriceRosemary = 2	
	generated is less	*nSellPriceMint = 200	nSellPriceMint = 200	nSellPriceMint = 200	
	than 1				
2	nSellPriceLavender	*nSellPriceLemon = 12	nSellPriceLemon = 12	nSellPriceLemon = 12	Pass
	will be updated	*nSellPriceLavender = -18	nSellPriceLavender = 0	nSellPriceLavender = 0	
	because price	*nSellPriceRosemary = 22	SellPriceRosemary = 22	SellPriceRosemary = 22	
	generated is less	*nSellPriceMint = 24	nSellPriceMint = 24	nSellPriceMint = 24	
	than 1				
3	nSellPriceRosemary	*nSellPriceLemon = 2000	nSellPriceLemon = 2000	nSellPriceLemon = 2000	Pass
	will be updated	*nSellPriceLavender = 8	nSellPriceLavender = 8	nSellPriceLavender = 8	
	because price	*nSellPriceRosemary = -12	nSellPriceRosemary = 0	nSellPriceRosemary = 0	
	generated is less	*nSellPriceMint = 24	nSellPriceMint = 24	nSellPriceMint = 24	
	than 1				
4	nSellPriceMint will	*nSellPriceLemon = 200	nSellPriceLemon = 200	nSellPriceLemon = 200	Pass
	be updated	*nSellPriceLavender = 80	nSellPriceLavender = 80	nSellPriceLavender = 80	
	because price	*nSellPriceRosemary = 88	nSellPriceRosemary = 88	nSellPriceRosemary = 88	
	generated is less	*nSellPriceMint = -1208	nSellPriceMint = 0	nSellPriceMint = 0	
	than 1				
5	No Sell Price will be	*nSellPriceLemon = 200	nSellPriceLemon = 200	nSellPriceLemon = 200	Pass
	updated	*nSellPriceLavender = 80	nSellPriceLavender = 80	nSellPriceLavender = 80	
		*nSellPriceRosemary = 88	nSellPriceRosemary = 88	nSellPriceRosemary = 88	
		*nSellPriceMint = 128	nSellPriceMint = 128	nSellPriceMint = 128	

Function: void bottlePriceChecker(int *nPriceBottleLemon, int *nNoBottleLemon, int *nPriceBottleLavender, int *nNoBottleLavender, int *nPriceBottleRosemary, int *nPriceBottleMint, int *nNoBottleMint)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/ Fail
1	No Buy Price and	*nPriceBottleLemon = 183	nPriceBottleLemon = 183	nPriceBottleLemon = 183	Pass
	No. of available EO	*nNoBottleLemon = 9	nNoBottleLemon = 9	nNoBottleLemon = 9	
	Bottle/s in a City	*nPriceBottleLavender = 19	nPriceBottleLavender = 19	nPriceBottleLavender = 19	
	will be updated	*nNoBottleLavender = 8	nNoBottleLavender = 8	nNoBottleLavender = 8	
		*nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	
		*nNoBottleRosemary = 8	nNoBottleRosemary = 8	nNoBottleRosemary = 8	
		*nPriceBottleMint = 113	nPriceBottleMint = 113	nPriceBottleMint = 113	
		*nNoBottleMint = 4	nNoBottleMint = 4	nNoBottleMint = 4	
2	nPriceBottleLemon	*nPriceBottleLemon = -183	nPriceBottleLemon = 0	nPriceBottleLemon = 0	Pass
	and	*nNoBottleLemon = 9	nNoBottleLemon = 0	nNoBottleLemon = 0	
	nNoBottleLemon	*nPriceBottleLavender = 19	nPriceBottleLavender = 19	nPriceBottleLavender = 19	
	will be updated	*nNoBottleLavender = 8	nNoBottleLavender = 8	nNoBottleLavender = 8	
	because price	*nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	
	generated is less	*nNoBottleRosemary = 8	nNoBottleRosemary = 8	nNoBottleRosemary = 8	
	than 1	*nPriceBottleMint = 113	nPriceBottleMint = 113	nPriceBottleMint = 113	
		*nNoBottleMint = 4	nNoBottleMint = 4	nNoBottleMint = 4	
3	nPriceBottleLavend	*nPriceBottleLemon = 183	nPriceBottleLemon = 188	nPriceBottleLemon = 188	Pass
	er and	*nNoBottleLemon = 9	nNoBottleLemon = 9	nNoBottleLemon = 9	
	nNoBottleLavender	*nPriceBottleLavender = -19	nPriceBottleLavender = 0	nPriceBottleLavender = 0	
	will be updated	*nNoBottleLavender = 8	nNoBottleLavender = 0	nNoBottleLavender = 0	
	because price	*nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	
	generated is less	*nNoBottleRosemary = 8	nNoBottleRosemary = 8	nNoBottleRosemary = 8	
	than 1	*nPriceBottleMint = 113	nPriceBottleMint = 113	nPriceBottleMint = 113	
		*nNoBottleMint = 4	nNoBottleMint = 4	nNoBottleMint = 4	
4	nPriceBottleRosem	*nPriceBottleLemon = 183	nPriceBottleLemon = 188	nPriceBottleLemon = 188	Pass
	ary and	*nNoBottleLemon = 9	nNoBottleLemon = 9	nNoBottleLemon = 9	
	nNoBottleRosemar	*nPriceBottleLavender = 19	nPriceBottleLavender = 19	nPriceBottleLavender = 19	
	y will be updated	*nNoBottleLavender = 8	nNoBottleLavender = 8	nNoBottleLavender = 8	
	because price	*nPriceBottleRosemary = -46	nPriceBottleRosemary = 0	nPriceBottleRosemary = 0	
	generated is less	*nNoBottleRosemary = 8	nNoBottleRosemary = 0	nNoBottleRosemary = 0	
	than 1	*nPriceBottleMint = 113	nPriceBottleMint = 113	nPriceBottleMint = 113	

		*nNoBottleMint = 4	nNoBottleMint = 4	nNoBottleMint = 4	
-,	nPriceBottleMint	*nPriceBottleLemon = 183	nPriceBottleLemon = 188	nPriceBottleLemon = 188	Pass
	and nNoBottleMint	*nNoBottleLemon = 9	nNoBottleLemon = 9	nNoBottleLemon = 9	
	will be updated	*nPriceBottleLavender = 19	nPriceBottleLavender = 19	nPriceBottleLavender = 19	
	because price	*nNoBottleLavender = 8	nNoBottleLavender = 8	nNoBottleLavender = 8	
	generated is less	*nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	nPriceBottleRosemary = 46	
	than 1	*nNoBottleRosemary = 8	nNoBottleRosemary = 8	nNoBottleRosemary = 8	
		*nPriceBottleMint = -113	nPriceBottleMint = 0	nPriceBottleMint = 0	
		*nNoBottleMint = 4	nNoBottleMint = 0	nNoBottleMint = 0	

Function: void displaySellMenu()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell	None	Formatted Sell Menu		Pass
	Menu			SELL MENU	
				Press [1] Sell Essential Oil/s	
				Press [2] Sell Diffuser Oil/s Press [0] Go back to Actions Menu	

Function: void displaySellEOMenu()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell	None	Formatted Sell Essential		Pass
	Essential Oils Menu		Oils Menu	SELL ESSENTIAL OILS MENU	
				Press	
				[1] Lemon [2] Lavender .	
				[3] Rosemary [4] Mint	
				[0] Go back to Actions Menu	
				[0] GO BACK TO ACCIONS MENU	
				ne o tu te into n i to	

Function: void displaySellEO()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell EO confirmation	None	Formatted Sell EO confirmation choices	Press [1] Sell Essential Oil/s Press [0] Don't execute transaction. Go back to Actions Menu	Pass

Function: void displaySellDOMenu()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell	None	Formatted Sell Diffuser Oils		Pass
	Diffuser Oils Menu		Menu	SELL DIFFUSER OILS MENU	
				Press	
				[1] Clear Minds [2] Energy Booster	
				[3] Calming	
				[0] Go back to Actions Menu	

Function: void displaySellDOConfirmation()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell DO confirmation	none	Formatted Sell DO confirmation choices	Press [3] Sell Diffuser Oil/S Press [9] Don't execute transaction. Go back to Actions Menu	Pass

Function: int getMaxNoBottleCMinds(int nStockLemonDrops, int nStockRosemaryDrops, int nStockMintDrops)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Outputs the	nStockLemonDrops = 10	3	3	Pass
	maximum number	nStockRosemaryDrops = 10			
	of Clear Minds the	nStockMintDrops = 10			
	player can make				

Function: int getMaxNoBottleEBooster(int nStockLemonDrops, int nStockRosemaryDrops, int nStockMintDrops)

Test Descript	ion Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1 Outputs the	nStockLemonDrops = 10	10	10	Pass
maximum num	ber nStockRosemaryDrops = 10			
of Energy Boos	ter nStockMintDrops = 10			
the player can				
make				

Function: int getMaxNoBottleCalm(int nStockLemonDrops, int nStockLavenderDrops, int nStockRosemaryDrops)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Outputs the	nStockLemonDrops = 10	3	3	Pass
	maximum number	nStockLavenderDrops = 10			
	Calming the player	nStockRosemaryDrops = 10			
	can make				

Function: int getSellPriceBottleCMinds(int nSellPriceLemon, int nSellPriceRosemary, int nSellPriceMint)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Outputs the sell	nSellPriceLemon = 97	541	541	Pass
	Price per unit of	nSellPriceRosemary = 42			
	Clear Minds	nSellPriceMint = 48			

Function: int getSellPriceBottleEBooster(int nSellPriceLemon, int nSellPriceRosemary, int nSellPriceMint)

Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1 Outputs the sell	nSellPriceLemon = 97	215	215	Pass
Price per unit of	nSellPriceRosemary = 42			
Energy Booster	nSellPriceMint = 48			

Function: int getSellPriceBottleCalm(int nSellPriceLemon, int nSellPriceLavender, int nSellPriceRosemary)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Outputs the sell	nSellPriceLemon = 97	366	366	Pass
	Price per unit of	nSellPriceLavender = 42			
	Calming	nSellPriceRosemary = 48			

Function: void displayMakeDOMenu(int nStockLemonDrops, int nStockLavenderDrops, int nStockRosemaryDrops, int nStockMintDrops, int nNoBottleCMinds, int nNoBottleEBooster, int nNoBottleCalm)

Ī		Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
	1	Prints the Make Diffuser Oils Menu (Player can make Clear Minds, Energy Booster, & Calming)	nStockLemonDrops = 10 nStockLavenderDrops = 10 nStockRosemaryDrops = 10 nStockMintDrops = 10 nNoBottleCMinds = 3 nNoBottleEBooster = 10 nNoBottleCalm = 3	Formatted Make DO Menu	Actual output/result	Pass
	2	Prints the Make Diffuser Oils Menu (Player can make Energy Booster, & Calming)	nStockLemonDrops = 1 nStockLavenderDrops = 3 nStockRosemaryDrops = 2 nStockMintDrops = 1 nNoBottleCMinds = 0 nNoBottleEBooster = 1 nNoBottleCalm = 1	Formatted Make DO Menu		Pass
	3	Prints the Make Diffuser Oils Menu (Player cannot make any Diffuser Oil/s)	nStockLemonDrops = 0 nStockLavenderDrops = 0 nStockRosemaryDrops = 0 nStockMintDrops = 0 nNoBottleCMinds = 0 nNoBottleEBooster = 0 nNoBottleCalm = 0	Formatted Make DO Menu	Oxpsic. You don't have enough Essential Oils :(([0] Go back to Actions Menu	Pass

Function: int isMakeDOValid(int nNoBottleMakeDO, int nNoBottleDO)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Returns 1 (player's	nNoBottleMakeDO = 5	1	1	Pass
	input is valid)	nNoBottleDO = 10			
2	Returns 0 (player's	nNoBottleMakeDO = 8	0	0	Pass
	input is invalid)	nNoBottleDO = 5			

Function: void displayConfirmMakeDO()

Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
Prints the Make DO confirmation	none	Formatted Make DO confirmation choices	Press [1] Make Diffuser Oil/s Press [0] Don't make. Go back to Actions Menu	Pass

Function: int getRandCityCharge (int nLowerCity, int nUpperCity, int nCountCity)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	"Make charge	nLowerCity = 20	20 - 30	28	Pass
	generated is within	nUpperCity = 30			
	range of:" Manila	nCountCity = 1			
2	" "Makati	nLowerCity = 80	80 - 100	88	Pass
		nUpperCity = 100			
		nCountCity = 1			
3	" "Alabang	nLowerCity = 70	70 - 90	83	Pass
		nUpperCity = 90			
		nCountCity = 1			
4	" "QC	nLowerCity = 40	40 - 60	59	Pass
		nUpperCity = 60			
		nCountCity = 1			

Function: int getCityCharge(int nCity)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	"Make charge	nCity = 1	20 - 30	28	Pass
	generated is within				
	range of:" Manila				
2	" "Makati	nCity = 3	80 - 100	88	Pass
3	" "Alabang	nCity = 2	70 - 90	83	Pass
4	" "QC	nCity = 4	40 - 60	59	Pass

Function: void updateEOStockCMinds(int nNoBottleMakeDO, int *nStockLemonDrops, int *nStockRosemaryDrops, int *nStockMintDrops)

		Test Description	Input value/ parameters	Expected output/result	Actual output/ result	Pass/ Fail
ĺ	1	Stock on Hand will	nNoBottleMakeDO = 1	nNoBottleMakeDO = 1	Same as expected results	Pass
		be updated (player	*nStockLemonDrops = 10	*nStockLemonDrops = 7		
		made 1 bottle of	*nStockRosemaryDrops = 10	*nStockRosemaryDrops = 8		
		Clear Minds)	*nStockMintDrops = 10	*nStockMintDrops = 8		

Function: void updateEOStockEBooster(int nNoBottleMakeDO, int *nStockLemonDrops, int *nStockRosemaryDrops, int *nStockMintDrops)

	Test Description	Input value/ parameters	Expected output/result	Actual output/	Pass/
				result	Fail
1 9	Stock on Hand will	nNoBottleMakeDO = 1	nNoBottleMakeDO = 1	Same as expected results	Pass
Ł	be updated (player	*nStockLemonDrops = 10	*nStockLemonDrops = 9		
r	made 1 bottle of	*nStockRosemaryDrops = 10	*nStockRosemaryDrops = 9		
E	Energy Booster	*nStockMintDrops = 10	*nStockMintDrops = 9		
ſ	Minds)				

Function: void updateEOStockCalm(int nNoBottleMakeDO, int *nStockLemonDrops, int *nStockLavenderDrops, int *nStockRosemaryDrops)

	Test Description	Input value/ parameters	Expected output/result	Actual output/ result	Pass/ Fail
	Charles Handell	AND POUL MADE DO 1	AND DANIE NACE DO 1	100000	1 9.11
1	Stock on Hand will	nNoBottleMakeDO = 1	nNoBottleMakeDO = 1	Same as expected results	Pass
	be updated (player	*nStockLemonDrops = 10	*nStockLemonDrops = 9		
	made 1 bottle of	*nStockRosemaryDrops = 10	*nStockLavenderDrops = 7		
	Calming)	*nStockMintDrops = 10	*nStockRosemaryDrops = 8		

Function: void displayPayDebtMenu(int nDoubt)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Pay Debt	nDebt = 4000	Formatted Pay Debt Menu		Pass
	Menu			PAY OBBT MENU (\	

Function: int isValidAmountPayDebt(int nAmountPayDebt, int nDebt, int nCash)

	Test Description	Input value/ parameters	Expected output/result	Actual output/	Pass/
				result	Fail
1	Player has enough	nAmountPayDebt = 3000	1	1	Pass
	Cash to Pay specified	nDebt = 4000			
	Amount of Debt to	nCash = 8000			
	Pay				
2	Player does not have	nAmountPayDebt = 9000	0	0	Pass
	enough Cash to Pay	nDebt = 10000			
	specified Amount of	nCash = 8000			
	Debt to Pay				

Function: void displayLoanMenu(int nLoan)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Loan Menu	nLoan = 10000	Formatted Loan Menu	LOAN MENU (I) (I) (IV) (IV	Pass

Function: int isValidAmountDebtLoan(int nAmountWishLoan, int nLoan)

Test Description Input	t value/ parameters	Expected output/result	Actual output/ result	Pass/ Fail
,	untWishLoan = 8000 = 10000	1	1	Pass
'	untWishLoan = 20000 = 10000	0	0	Pass

Function: void displayPerCityDay(int nDay, int nCity, int nCash, int nDebt)

		Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
	1	Prints the Display Per	nDay = 8	Formatted Display Per City/		Pass
		City/ Day (Manila)	nCity = 1	Day	/ Hello, Oller! \ Oay 8: Maxila	
			nCash = 1500		Cellshi	
			nDebt = 4000		1	
Į						
	2	Prints the Display Per	nDay = 9	Formatted Display Per City/		Pass
		City/ Day (Quezon	nCity = 4	Day	Hello, offer!	
		City)	nCash = 5500		(cash: \$ 5500	
			nDebt = 0		/ \ \text{OIL-1/ARL SPIZA}	

Function: void displayInventoryEO(int nStockLemonDrops, int nStockLavenderDrops, int nStockRosemaryDrops, int nStockMintDrops, int nSellPriceLemon, int nSellPriceLavender, int nSellPriceRosemary, int nSellPriceMint)

Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
Prints the Essential	nStockLemonDrops = 10	Formatted Essential Oils	II QIL-VENTORY II	Pass
Oils Inventory	nStockLavenderDrops = 10	Inventory		
	nStockRosemaryDrops = 10		Lavender	
	nStockMintDrops = 10		"	
	nSellPriceLemon = 128			
	nSellPriceLavender = 208			
	nSellPriceRosemary = 888			
	nSellPriceMint = 1088			

Function: void displayInventoryDO(int nStockCMinds, int nStockEBooster, int nStockCalm, int nSellPriceBottleCMinds, int nSellPriceBottleEBooster, int nSellPriceBottleCalm)

Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
Prints the Diffuser	nStockCMinds = 10	Formatted Diffuser Oils		Pass
Oils Inventory	nStockEBooster = 10	Inventory		
	nStockCalm = 10			
	nSellPriceBottleCMinds =		ii i	
	128			
	nSellPriceBottleEBooster =			
	208			
	nSellPriceBottleCalm = 888			

Function: void displayTravelMenu(int nCity)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Travel Menu (Player is currently in Alabang)	nCity = 2	Formatted Travel Menu	TRAVEL MENU TRAVEL MENU Portila Outron City Press City Press City I Menila Mekeri	Pass
2	Prints the Travel Menu (Player is currently in Quezon City)	nCity = 4	Formatted Travel Menu	TRACE PEND S S Mentla Alabang (CGIL-NUMN)> press City	Pass

Function: int isValidChoiceTravelMenu(int nCity, int nChoiceTravelMenu)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Travel Menu choice	nCity = 1 (Manila)	1	1	Pass
	is valid	nChoiceTravelMenu = 3			
		(Makati)			
2	Travel Menu choice	nCity = 2 (Alabang)	0	0	Pass
	is invalid	nChoiceTravelMenu = 2			
		(Alabang)			

Function: void displayFreebieEO()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Oil-elf who gives free Essential Oils	None	Formatted EO freebies	Comprats, Oiler! I'm the Oil-elf and as my gift, you get one free bottle of Lemon,	Pass

Function: void displayFreebieDO()

		Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
Î	1	Prints the Teddy-oil	None	Formatted DO freebies		Pass
		who gives free DO			Congrats, Oiler! I'm Teddy-oil and	

Function: void displayGameSummary (int nCash, int nDebt, int nStockLemonDrops, int nStockLavenderDrops, int nStockRosemaryDrops, int nStockColmids, int nStockEBooster, int nStockCalm)

Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1 Prints the Game	nCash = 1500, nDebt = 0	Formatted Game Summary		Pass
Summary	nStockLemonDrops = 10			
	nStockLavenderDrops = 10			
	nStockRosemaryDrops = 10		Amunimization of the second of	
	nStockMintDrops = 10		E Company of the comp	
	nStockCMinds =5			
	nStockEBooster = 5			
	nStockCalm = 5			

Function: void displayExitMesage()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
	Prints the Exit Message	None	Formatted Exit Message	The property of the property o	Pass

Function: void updateLeaderBoard(int nCash)

			•	
	HALL OF OIL			
RANK	NAMES		SCORES/CASH	
1	Sheila	\$	216336	
2	Julia	\$	10600	
3	Jaimmy	\$	10500	
4	Robin	\$	10100	
5	Ted \$ 1	1006	3 0	
6	Jeff	\$	9000	
7	Barney	\$	8800	
8	John	\$	8800	
9	Britta	\$	8000	
10	Travis	\$	7080	

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Final nCash of Player enters the Leader Board	nCash = 8000	Player will enter name (Abed) and Leader Board will be updated	HALL OF OIL RANK NAMES SCORES/CASH 1	Pass
2	Final nCash of Player does not enter the Leader Board	nCash = 7000	Player will not enter their name and Leader Board will not be updated	HALL OF OIL RANK NAMES SCORES/CASH Sheila \$216336 2	Pass