


```
/* 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

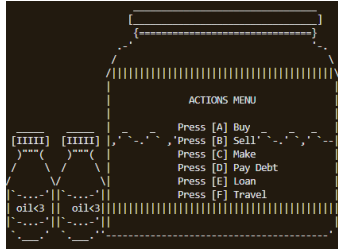
Function: void displayGameTitle()

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

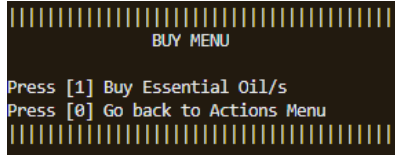
Function: void displayMainMenu()

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

Function: void displayActionsMenu()

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

Function: void displayBuyMenu()

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

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 within price range per bottle of:" Lemon	nLower = 100 nUpper = 150 nCount = 1	100 - 150	112	Pass
2	" "Lavender	nLower = 20 nUpper = 60 nCount = 1	20 - 60	35	Pass
3	" "Rosemary	nLower = 70 nUpper = 100 nCount = 1	70 - 100	81	Pass
4	" " Mint	nLower = 130 nUpper = 200 nCount = 1	130 - 200	148	Pass

Function: double getOperator()

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

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 between 101-200	nLower = 102 nUpper = 199 nCount = 1	random number between 101-200	196	Pass
2	Random number between 1-50	nLower = 2 nUpper = 49 nCount = 1	random number between 1-50	20	Pass

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 retained	nLower = 100 nUpper = 150 nCount = 1	108	108	Pass
2	nPriceBottleStep1 is added to nRandomFactor(18)	nLower = 100 nUpper = 150 nCount = 1	126	126	
3	nRandomFactor(18) is subtracted from nPriceBottleStep1	nLower = 100 nUpper = 150 nCount = 1	90	90	
4	nPriceBottleStep1 is multiplied to nRandomFactor(180)	nLower = 100 nUpper = 150 nCount = 1	19440	19440	

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 Player can Buy in a City is from 1 – 10 EO Bottle/s.	nLowerNoBottle = 1 nUpperNoBottle = 10 nCountNoBottle = 1	1 - 10	8	Pass

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 output/result	Actual output/result	Pass/Fail
1	Prints the Buy Essential Oils Menu	displayMenuBuyEO(118, 228, 338, 448, 18, 28, 108, 118)	Formatted Buy Essential Oils Menu	<pre> 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 </pre>	Pass

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	<pre> Press [1] Buy Essential Oil/s Press [0] Don't execute transaction. Go back to Actions Menu </pre>	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
1	Player has enough nCash	nCash = 1500 nBuyEOPrice = 1409	1	1	Pass
2	Player does not have enough nCash	nCash = 1500 nBuyEOPrice = 2280	0	0	Pass

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

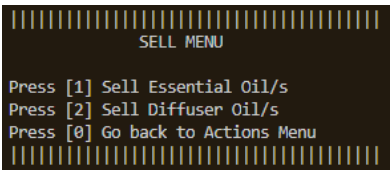
	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/ Fail
1	nSellPriceLemon will be updated because price generated is less than 1	*nSellPriceLemon = -100 *nSellPriceLavender = 18 *nSellPriceRosemary = 2 *nSellPriceMint = 200	nSellPriceLemon = 0 nSellPriceLavender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

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


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

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

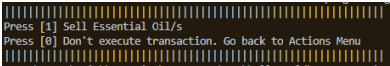
Function: void displaySellMenu()

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

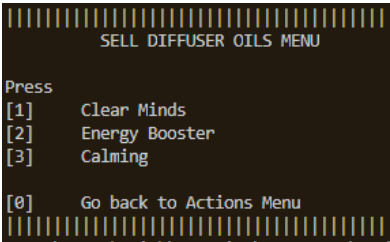
Function: void displaySellEOMenu()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell Essential Oils Menu	None	Formatted Sell Essential Oils Menu		Pass

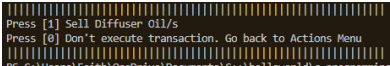
Function: void displaySellEO()

	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		Pass

Function: void displaySellDOMenu()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Sell Diffuser Oils Menu	None	Formatted Sell Diffuser Oils Menu		Pass

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		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 maximum number of Clear Minds the player can make	nStockLemonDrops = 10 nStockRosemaryDrops = 10 nStockMintDrops = 10	3	3	Pass

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

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

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 maximum number Calming the player can make	nStockLemonDrops = 10 nStockLavenderDrops = 10 nStockRosemaryDrops = 10	3	3	Pass

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 Price per unit of Clear Minds	nSellPriceLemon = 97 nSellPriceRosemary = 42 nSellPriceMint = 48	541	541	Pass

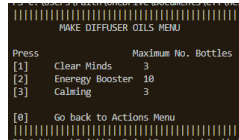
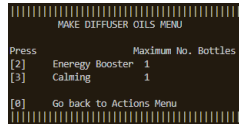
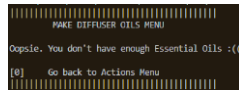
Function: int getSellPriceBottleEBooster(int nSellPriceLemon, int nSellPriceRosemary, int nSellPriceMint)

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

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 Price per unit of Calming	nSellPriceLemon = 97 nSellPriceLavender = 42 nSellPriceRosemary = 48	366	366	Pass

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

	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		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		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 input is valid)	nNoBottleMakeDO = 5 nNoBottleDO = 10	1	1	Pass
2	Returns 0 (player's input is invalid)	nNoBottleMakeDO = 8 nNoBottleDO = 5	0	0	Pass

Function: void displayConfirmMakeDO()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Make DO confirmation	none	Formatted Make DO confirmation choices	<pre> Press [1] Make Diffuser Oil/s Press [0] Don't make. Go back to Actions Menu </pre>	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 generated is within range of:" Manila	nLowerCity = 20 nUpperCity = 30 nCountCity = 1	20 - 30	28	Pass
2	" "Makati	nLowerCity = 80 nUpperCity = 100 nCountCity = 1	80 - 100	88	Pass
3	" "Alabang	nLowerCity = 70 nUpperCity = 90 nCountCity = 1	70 - 90	83	Pass
4	" "QC	nLowerCity = 40 nUpperCity = 60 nCountCity = 1	40 - 60	59	Pass

Function: `int getCityCharge(int nCity)`

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	"Make charge generated is within range of:" Manila	nCity = 1	20 - 30	28	Pass
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
1	Stock on Hand will be updated (player made 1 bottle of Clear Minds)	nNoBottleMakeDO = 1 *nStockLemonDrops = 10 *nStockRosemaryDrops = 10 *nStockMintDrops = 10	nNoBottleMakeDO = 1 *nStockLemonDrops = 7 *nStockRosemaryDrops = 8 *nStockMintDrops = 8	Same as expected results	Pass

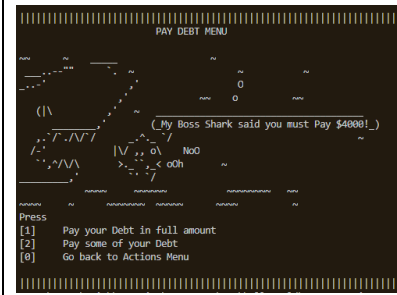
Function: `void updateEOStockEBooster(int nNoBottleMakeDO, int *nStockLemonDrops, int *nStockRosemaryDrops, int *nStockMintDrops)`

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

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

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

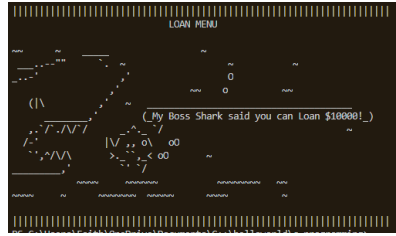
Function: void displayPayDebtMenu(int nDebt)

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

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

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


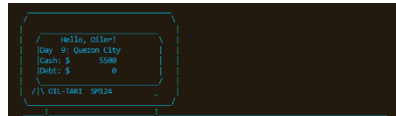
Function: void displayLoanMenu(int nLoan)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Loan Menu	nLoan = 10000	Formatted Loan Menu		Pass

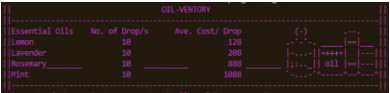
Function: int isValidAmountDebtLoan(int nAmountWishLoan, int nLoan)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Player entered a valid Amount to Loan	nAmountWishLoan = 8000 nLoan = 10000	1	1	Pass
2	Player entered an invalid Amount to Loan	nAmountWishLoan = 20000 nLoan = 10000	0	0	Pass

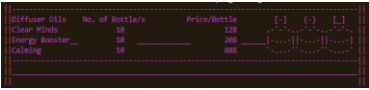
Function: void displayPerCityDay(int nDay, int nCity, int nCash, int nDebt)

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Display Per City/ Day (Manila)	nDay = 8 nCity = 1 nCash = 1500 nDebt = 4000	Formatted Display Per City/ Day		Pass
2	Prints the Display Per City/ Day (Quezon City)	nDay = 9 nCity = 4 nCash = 5500 nDebt = 0	Formatted Display Per City/ Day		Pass


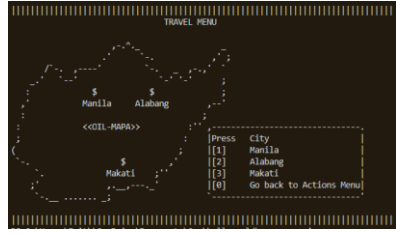
Function: void displayInventoryEO(int nStockLemonDrops, int nStockLavenderDrops, int nStockRosemaryDrops, int nStockMintDrops, int nSellPriceLemon, int nSellPriceLavender, int nSellPriceRosemary, int nSellPriceMint)

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

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
1	Prints the Diffuser Oils Inventory	nStockCMinds = 10 nStockEBooster = 10 nStockCalm = 10 nSellPriceBottleCMinds = 128 nSellPriceBottleEBooster = 208 nSellPriceBottleCalm = 888	Formatted Diffuser Oils Inventory		Pass

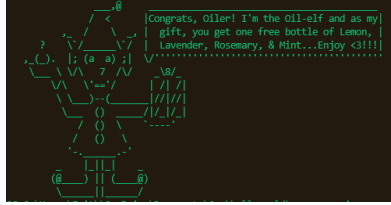
Function: void displayTravelMenu(int nCity)

	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		Pass
2	Prints the Travel Menu (Player is currently in Quezon City)	nCity = 4	Formatted Travel Menu		Pass

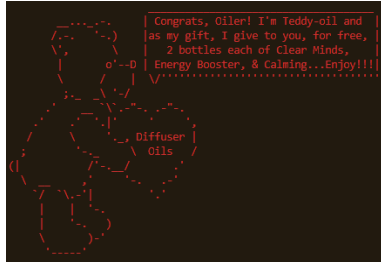
Function: int isValidChoiceTravelMenu(int nCity, int nChoiceTravelMenu)

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

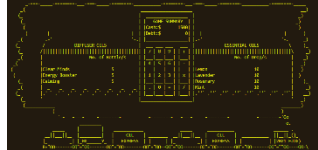
Function: void displayFreebieEO()

	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		Pass


Function: void displayFreebieDO()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Teddy-oil who gives free DO	None	Formatted DO freebies		Pass

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

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

Function: void displayExitMesage()

	Test Description	Input value/ parameters	Expected output/result	Actual output/result	Pass/Fail
1	Prints the Exit Message	None	Formatted Exit Message		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	\$	10000
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	<table><tr><td></td><td colspan="3">HALL OF OIL</td></tr><tr><td>RANK</td><td>NAMES</td><td colspan="2">SCORES/CASH</td></tr><tr><td>1</td><td>Sheila</td><td>\$</td><td>216336</td></tr><tr><td>2</td><td>Julia</td><td>\$</td><td>10600</td></tr><tr><td>3</td><td>Jaimmy</td><td>\$</td><td>10500</td></tr><tr><td>4</td><td>Robin</td><td>\$</td><td>10100</td></tr><tr><td>5</td><td>Ted</td><td>\$</td><td>10000</td></tr><tr><td>6</td><td>Jeff</td><td>\$</td><td>9000</td></tr><tr><td>7</td><td>Barney</td><td>\$</td><td>8800</td></tr><tr><td>8</td><td>John</td><td>\$</td><td>8800</td></tr><tr><td>9</td><td>Britta</td><td>\$</td><td>8000</td></tr><tr><td>10</td><td>Abed</td><td>\$</td><td>8000</td></tr></table>		HALL OF OIL			RANK	NAMES	SCORES/CASH		1	Sheila	\$	216336	2	Julia	\$	10600	3	Jaimmy	\$	10500	4	Robin	\$	10100	5	Ted	\$	10000	6	Jeff	\$	9000	7	Barney	\$	8800	8	John	\$	8800	9	Britta	\$	8000	10	Abed	\$	8000	Pass
	HALL OF OIL																																																				
RANK	NAMES	SCORES/CASH																																																			
1	Sheila	\$	216336																																																		
2	Julia	\$	10600																																																		
3	Jaimmy	\$	10500																																																		
4	Robin	\$	10100																																																		
5	Ted	\$	10000																																																		
6	Jeff	\$	9000																																																		
7	Barney	\$	8800																																																		
8	John	\$	8800																																																		
9	Britta	\$	8000																																																		
10	Abed	\$	8000																																																		
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	<table><tr><td></td><td colspan="3">HALL OF OIL</td></tr><tr><td>RANK</td><td>NAMES</td><td colspan="2">SCORES/CASH</td></tr><tr><td>1</td><td>Sheila</td><td>\$</td><td>216336</td></tr><tr><td>2</td><td>Julia</td><td>\$</td><td>10600</td></tr><tr><td>3</td><td>Jaimmy</td><td>\$</td><td>10500</td></tr><tr><td>4</td><td>Robin</td><td>\$</td><td>10100</td></tr><tr><td>5</td><td>Ted</td><td>\$</td><td>10000</td></tr><tr><td>6</td><td>Jeff</td><td>\$</td><td>9000</td></tr><tr><td>7</td><td>Barney</td><td>\$</td><td>8800</td></tr><tr><td>8</td><td>John</td><td>\$</td><td>8800</td></tr><tr><td>9</td><td>Britta</td><td>\$</td><td>8000</td></tr><tr><td>10</td><td>Abed</td><td>\$</td><td>8000</td></tr></table>		HALL OF OIL			RANK	NAMES	SCORES/CASH		1	Sheila	\$	216336	2	Julia	\$	10600	3	Jaimmy	\$	10500	4	Robin	\$	10100	5	Ted	\$	10000	6	Jeff	\$	9000	7	Barney	\$	8800	8	John	\$	8800	9	Britta	\$	8000	10	Abed	\$	8000	Pass
	HALL OF OIL																																																				
RANK	NAMES	SCORES/CASH																																																			
1	Sheila	\$	216336																																																		
2	Julia	\$	10600																																																		
3	Jaimmy	\$	10500																																																		
4	Robin	\$	10100																																																		
5	Ted	\$	10000																																																		
6	Jeff	\$	9000																																																		
7	Barney	\$	8800																																																		
8	John	\$	8800																																																		
9	Britta	\$	8000																																																		
10	Abed	\$	8000																																																		