CCPROG1 Term 1, AY 2021 – 2022

Test Script Document

Name	Francisco Emmanuel T. Dumas
Section	S17B

Test Scripts

There should be at least 3 distinct test classes (as indicated in the description) per function. There is no need to test functions which are only for screen design.

Function Name: <Merchant>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	The variable nRand is equal to 0, and the user has 200 GD.	<cConfirm> <Y>	Ask the user if they want to increase their nCapacity.	Ask the user if they want to increase their nCapacity.	P
2	The variable nRand is equal to 5, and the user has 100 GD.	<cConfirm> <y>	Tells the user that they don't have enough fGD, and then asks for input again.	Tells the user that they don't have enough fGD, and then asks for input again.	P
3	The variable nRand is equal to 9, and the user has 300 GD.	<cConfirm> <N>	The function ends	The function ends	P
4	The variable nRand is equal to 10, and the user has 200 GD	N/A	The function ends.	The function ends.	P

Function Name: <DisplayItem>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nItemNo is equal to 4	<nItemNo> <4>	Program prints Intoxicating Perfume.	Program prints Intoxicating Perfume.	P
2	nItemNo is equal to 1	<nItemNo> <1>	Program prints Sweet Beet.	Program prints Sweet Beet.	P
3	nItemNo is equal to 8	<nItemNo> <8>	Program prints Valyrian Steel	Program prints Valyrian Steel	P

Function Name: <DisplayCurrentCity>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nCityNum is equal to 2	<nCityNum> <2>	Program runs the Lys function.	Program runs the Lys function.	P
2	nCityNum is equal to 6	<nCityNum> <6>	Program runs the Volantis function.	Program runs the Volantis function.	P
3	nCityNum is equal to 3	<nCityNum> <3>	Program runs the Myr function.	Program runs the Myr function.	P

Function Name: <SellScreen>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	User has 5 [1] nSweetBeet and is currently in Winterfell.	<nItemNo, nSell> <1, 5> <cConfirm> <Y>	The program asks for validation,n SweetBeet is deducted by 5 and fGD is increased by 900.	The program asks for validation,n SweetBeet is deducted by 5 and fGD is increased by 900.	P
2	User has 8 [5] nPaleAmberWine and is currently in Pentos.	<nItemNo, nSell> <3, 8>	The program prints “Insufficient number of Intricate Lace,” then asks for input again.	The program prints “Insufficient number of Intricate Lace,” then asks for input again.	P
3	User has 20 [2] nTimbert and is currently in Volantis.	<nItemNo, nSell> <0, 0>	The function ends, and the function Options is displayed.	The function ends, and the function Options is displayed.	P
4	User has 3 [6] nMyrishEye and is currently in Lys.	<nItemNo, nSell> <10, 9>	The program prints “Invalid Input,” then asks for input again.	The program prints “Invalid Input,” then asks for input again.	P

Function Name: <BuyScreen>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	User has 2000 nGD, 0 items in their Total Inventory, has 50 nCapacity, and is currently in Lys.	<nItemNo, nBuy> <1, 5> <cConfirm> <Y>	The program asks for validation,n SweetBeet is increased by 5 and fGD is decreased by 915.	The program asks for validation, nSweetBe et is increased by 5 and fGD is decrease d by 915.	P
2	User has 10,000 nGD, 49 items in their Total Inventory, has 50 nCapacity, and is currently in Volantis.	<nItemNo, nBuy> <6, 10>	The program prints “Exceeds max capacity,” then asks for input again.	The program prints “Exceeds max capacity,” then asks for input again.	P
3	User has 500 nGD, 20 items in their Total Inventory, has 100 nCapacity, and is currently in Pentos.	<nItemNo, nBuy> <0, 0>	The function ends, and the function Options is displayed.	The function ends, and the function Options is displayed.	P
4	User has 3000 nGD, 65 items in their Total Inventory, has 100 nCapacity, and is currently in Myr.	<nItemNo, nBuy> <100, 100>	The program prints “Invalid Input,” then asks for input again.	The program prints “Invalid Input,” then asks for input again.	P

Function Name: <WheelHouse>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nDayOver is equal to 0 and the user has 10 nDays left	<cChoice> <Y>	nDayOver is equal to 1 and nDays is decreased by 1, becoming 9 nDays.	nDayOver is equal to 1 and nDays is decreased by 1, becoming 9 nDays.	P
2	nDayOver is equal to 0 and the user has 7 nDays left	<cChoice> <n>	nDayOver is equal to 0 and the function Options is displayed.	nDayOver is equal to 0 and the function Options is displayed.	P
3	nDayOver is equal to 0 and has 15 nDays left	<cChoice> <a>	The program asks for input again.	The program asks for input again.	P

Function Name: <IronBankScreen>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	fGD is 5000, fDebt is 0.00, and fSavings is 0.00.	<nBankChoice, nMoney> <1, 2000>	fGD is decreased by 2000, and fSavings is increased by 2000.	fGD is decreased by 2000, and fSavings is increased by 2000.	P
2	fGD is 300, fDebt is 0.00, and fSavings is 5000.	<nBankChoice, nMoney> <2, 5000>	fGD is increased by 5000, and fSavings is decreased by 5000.	fGD is increased by 5000, and fSavings is decreased by 5000.	P

3	fGD is 100, fDebt is 0.00, and fSavings is 0.00.	<nBankChoice, nMoney> <3, 2000>	fGD is increased by 2000, and fDebt is increased by 2000.	fGD is decreased by 2000, and fSavings is increased by 2000.	P
4	fGD is 3000, fDebt is 2000, and fSavings is 5000.	<nBankChoice, nMoney> <2, 6000>	The program prints “Insufficient funds,” and then asks for input again.	The program prints “Insufficient funds,” and then asks for input again.	P
5	fGD is 500, fDebt is 1000, and fSavings is 5000.	<nBankChoice, nMoney> <2, -500>	The program prints “invalid Input,” and then asks for input again.	The program prints “invalid Input,” and then asks for input again.	P

Function Name: <ChooseTradingPartners>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nCityNum is equal to 0.	<nTradingPartner> <5>	nCityNum is equal to 5, and the function Qohor is displayed.	nCityNum is equal to 5, and the function Qohor is displayed.	P
2	nCityNum is equal to 3.	<nTradingPartner> <4>	nCityNum is equal to 4, and the function Pentos is displayed.	nCityNum is equal to 4, and the function Pentos is displayed.	P
3	nCityNum is equal to 5.	<nTradingPartner> <9>	The program asks for input again.	The program asks for input again.	P

Function Name: <CurrentItem>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nCurrentItem is equal to 0, and nSweetBeet is equal to 5.	<nItemNo> <1>	nCurrentItem gets the current value of [1] nSweetBeet.nCurrentItem becomes 5.	nCurrentItem gets the current value of [1] nSweetBeet.nCurrentItem becomes 5.	P
2	nCurrentItem is equal to 6, and nTimber is equal to 15.	<nItemNo> <2>	nCurrentItem gets the current value of [2] nTimber.nCurrentItem becomes 15.	nCurrentItem gets the current value of [2] nTimber.nCurrentItem becomes 15.	P
3	nCurrentItem is equal to 8, and nMyrishEye is equal to 40.	<nItemNo> <6>	nCurrentItem gets the current value of [6] nMyrishEye.n	nCurrentItem gets the current value of [6] nMyrishEye.	P

			CurrentItem becomes 40.	nCurrentItem becomes 40.	

Function Name: <TotalInventory>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	nTotal is equal to 0, the user has 5 Item [1], 6 Item [2], 10 Item [3], 0 Item [4], 2 Item [5], 1 Item [6], 3 Item [7], 1 Item [8].	<nSweetBeet, nTimber, nIntricateLace, nIntoxicatingPerfume, nPaleAmberWine, nMyrishEye, nQohorikTapestry, nValyrianSteel> <5, 6, 10, 0, 2, 1, 3, 1>	nTotal will get the sum of all the items, which is nTotal 28.	nTotal will get the sum of all the items, which is nTotal 28.	P
2	nTotal is equal to 0, the user has 1 Item [1], 2 Item [2], 3 Item [3], 4 Item [4], 5 Item [5], 6 Item [6], 7 Item [7], 8 Item [8].	<nSweetBeet, nTimber, nIntricateLace, nIntoxicatingPerfume, nPaleAmberWine, nMyrishEye, nQohorikTapestry, nValyrianSteel> <1, 2, 3, 4, 5, 6, 7, 8>	nTotal will get the sum of all the items, which is nTotal 36.	nTotal will get the sum of all the items, which is nTotal 36.	P
3	nTotal is equal to 0, the user has 9 Item [1], 10 Item [2], 8 Item [3], 7 Item [4], 6 Item [5], 5 Item [6], 4 Item [7], 3 Item [8].	<nSweetBeet, nTimber, nIntricateLace, nIntoxicatingPerfume, nPaleAmberWine, nMyrishEye, nQohorikTapestry, nValyrianSteel> <9, 10, 8, 7, 6, 5, 4, 3>	nTotal will get the sum of all the items, which is nTotal 52.	nTotal will get the sum of all the items, which is nTotal 52.	P

Function Name: <Options>					
Test #	**Test Description**	**Sample Input**	**Expected Result**	**Actual Result**	**P/F**
1	cOptions has no value.	<cOptions> <B>	The screen clears and then the Buy screen function is displayed.	The screen clears and then the buy screen function is displayed.	P
2	cOptions has no value.	<cOptions> <s>	The screen clears and then the Sell screen function is displayed.	The screen clears and then the Sell screen function is displayed.	P
3	cOptions has no value.	<cOptions> <W>	The screen clears and then the WheelHouse function is displayed.	The screen clears and then the WheelHouse function is displayed.	P
4	cOptions has no value.	<cOptions> <i>	The screen clears and then the IronBank screen function is displayed.	The screen clears and then the IronBank screen function is displayed.	P
5	cOptions has no value, and nSentinel has a value of 0	<cOptions> <Q>	The screen clears and then nSentinel will become a value of 1.	The screen clears and then nSentinel will become a value of 1.	P