PIZZA PICKER: TEST CASE DOCUMENTS

CCPROG1 S14A Machine Project

By ERMITANO, Kate Justine U. [12073164]

To be submitted to Professor Danniel L. Alcantara / Professor John Alexander Santillana

FUNCTION: INT MAKEPIZZA

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Making a pizza	nPizzasMade = 1 nPizzas = 0	nPizzas = 1	nPizzas = 1	PASS
Adding a pizza to the total no. of pizzas	nPizzasMade = 1 nPizzas = 68	nPizzas = 69	nPizzas = 69	PASS
Adding a pizza to the total no. of pizzas (+ cooks' pizzas)	nPizzasMade = 1 nPizzas = 28 fPizo = 100.00 nCooks = 5	nPizzas = 34 fPizo = 85.00 nCooks = 5	nPizzas = 34 fPizo = 85.00 nCooks = 5	PASS
Adding a pizza to the total no. of pizzas (+ cooks' pizzas + customers buying)	Day 24 nPizzasMade = 1 nPizzas = 8 fPizo = 100.00 nCooks = 5 nCustomers = 7	Day 25 nPizzas = 9 fPizo = 95.00 nCooks = 5 nCustomers = 7	Day 25 nPizzas = 9 fPizo = 95.00 nCooks = 5 nCustomers = 7	PASS

FUNCTION: INT MAKEPIZZA

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Adding a pizza to the total no. of pizzas (+ cooks' pizzas + customers buying)	Day 16 nPizzasMade = 1 nPizzas = 4 fPizo = 10.00 nCooks = 3 nCustomers = 4	Day 17 nPizzas = 5 fPizo = 10.00 nCooks = 3 nCustomers = 4	Day 17 nPizzas = 5 fPizo = 10.00 nCooks = 3 nCustomers = 4	PASS
Adding a pizza to the total no. of pizzas (+ cooks' pizzas + customers buying)	Day 6 nPizzasMade = 1 nPizzas = 2 fPizo = 5.00 nCooks = 2 nCustomers = 3	Day 7 nPizzas = 2 fPizo = 8.00 nCooks = 2 nCustomers = 3	Day 7 nPizzas = 2 fPizo = 8.00 nCooks = 2 nCustomers = 3	PASS

FUNCTION: INT SELLPIZZA

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Selling a pizza	*nAmountofPizza = 1 *fPizo = 0.00 *fPizzaPrice = 2.00 nPizzasSold = 1	nPizzas = 0 fPizo = 2.00	nPizzas = 0 fPizo = 2.00	PASS
Subtracting a pizza from the total no. of pizzas	*nAmountofPizza = = 35 *fPizo = 10.00 *fPizzaPrice = 2.00 nPizzasSold = 1	nPizzas = 34 fPizo = 12.00	nPizzas = 34 fPizo = 12.00	PASS
Selling a pizza without pizzas	*nAmountofPizza = 0 *fPizo = 0.00 *fPizzaPrice = 2.00 nPizzasSold = 1	“You cannot sell a pizza without pizzas!” nPizzas = 0 fPizo = 0.00	“You cannot sell a pizza without pizzas!” nPizzas = 0 fPizo = 0.00	PASS

NOTE: *AmountofPizza = nPizzas

FUNCTION: INT SELLPIZZA

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Subtracting a pizza from the total no. of pizzas (+ Customers buying)	Day 10 *nAmountofPizza = 69 *fPizo = 30.00 *fPizzaPrice = 2.00 nPizzasSold = 4 nCustomers = 7	Day 11 nPizzas = 65 fPizo = 38.00 nCustomers = 7	Day 11 nPizzas = 65 fPizo = 38.00 nCustomers = 7	PASS
Subtracting a pizza from the total no. of pizzas (+ Customers buying + Cooks' pizzas)	Day 10 *nAmountofPizza = 69 *fPizo = 30.00 *fPizzaPrice = 2.00 nPizzasSold = 4 nCooks = 5 nCustomers = 7	Day 11 nPizzas = 70 fPizo = 23.00 nCooks = 5 nCustomers = 7	Day 11 nPizzas = 70 fPizo = 23.00 nCooks = 5 nCustomers = 7	PASS

FUNCTION: INT HIRECOOKS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Hire a cook	nHiringCooks = 1 fPizo = 15.00 nCooks = 0	fPizo = 14.00 nCooks = 1	fPizo = 14.00 nCooks = 1	PASS
Hire another cook	nHiringCooks = 1 fPizo = 15.00 nCooks = 1	fPizo = 12.00 nCooks = 2	fPizo = 12.00 nCooks = 2	PASS
Hire another cook to a total of 2 cooks	nHiringCooks = 1 fPizo = 15.00 nCooks = 2	fPizo = 9.00 nCooks = 3	fPizo = 9.00 nCooks = 3	PASS
Hire another cook to a total of 3 cooks	nHiringCooks = 1 fPizo = 15.00 nCooks = 3	fPizo = 5.00 nCooks = 4	fPizo = 5.00 nCooks = 4	PASS
Hire another cook to a total of 4 cooks	nHiringCooks = 1 fPizo = 15.00 nCooks = 4	fPizo = 0.00 nCooks = 5	fPizo = 0.00 nCooks = 5	PASS

FUNCTION: INT HIRECOOKS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Hire a cook but with less pizo	nHiringCooks = 1 fPizo = 0.00 nCooks = 0	“1 cook(s) resigned.” fPizo = 0.00 nCooks = 0	“1 cook(s) resigned.” fPizo = 0.00 nCooks = 0	PASS
Hire another cook but with less pizo	nHiringCooks = 1 fPizo = 1.00 nCooks = 1	“1 cook(s) resigned.” fPizo = 0.00 nCooks = 1	“1 cook(s) resigned.” fPizo = 0.00 nCooks = 1	PASS
Wipe out all cooks (+ hired cook) with zero pizo	nHiringCooks = 1 fPizo = 0.00 nCooks = 4	“5 cook(s) resigned.” fPizo = 0.00 nCooks = 0	“5 cook(s) resigned.” fPizo = 0.00 nCooks = 0	PASS

FUNCTION: INT ATTRACTCUSTOMERS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Attract a customer with 69 pizzas	nAttractingCustomer = 1 nPizzas = 69 nPizo = 100.00 nCustomers = 0	nPizzas = 68 nPizo = 102.00 nCustomers = 1	nPizzas = 68 nPizo = 102.00 nCustomers = 1	PASS
Attract another customer with 69 pizzas	DAY 2 nAttractingCustomer = 1 nPizzas = 69 nPizo = 100.00 nCustomers = 1	DAY 3 nPizzas = 67 nPizo = 104.00 nCustomers = 2	DAY 3 nPizzas = 67 nPizo = 104.00 nCustomers = 2	PASS

FUNCTION: INT ATTRACTCUSTOMERS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Attract a customer but with no pizza	nAttractingCustomer = 1 nPizzas = 0 nCustomers = 0	“Customer 1 walked out” nPizzas = 0 nCustomers = 0	“Customer 1 walked out.” nPizzas = 0 nCustomers = 0	PASS
Attract another customer but with less pizza	Day 2 nAttractingCustomer = 1 nPizzas = 1 nCustomers = 1	Day 3 “Customer 2 walked out” nPizzas = 0 nCustomers = 1	Day 3 “Customer 2 walked out.” nPizzas = 0 nCustomers = 1	PASS
Attract another customer but with one pizza	Day 32 nAttractingCustomer = 1 nPizzas = 1 nCustomers = 9	Day 33 “Customer 2 walked out” nPizzas = 0 nCustomers = 1	Day 33 “Customer 2 walked out” nPizzas = 0 nCustomers = 1	PASS

FUNCTION: FLOAT RESEARCHPIZZAQUALITY

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Input a number under the total pizo	fResearchPrice = 69 *fPizo = 100.00 fPizzaPrice = 2.00 *ncounter = 0	fPizzo: 31.00 fPizzaPrice: 8.31	fPizzo: 31.00 fPizzaPrice: 8.31	PASS
Input a number equal the total pizo	fResearchPrice = 100 *fPizo = 100.00 fPizzaPrice = 2.00 *ncounter = 0	fPizzo: 0 fPizzaPrice: 10.00	fPizzo: 0.00 fPizzaPrice: 10.00	PASS
Input a number greater than the total pizo	fResearchPrice = 420 *fPizo = 100.00 fPizzaPrice = 2.00 *ncounter = 0	“You do not have enough pizo to do the research. Please input a different amount.”	“You do not have enough pizo to do the research. Please input a different amount.”	PASS

FUNCTION: FLOAT RESEARCHPIZZAQUALITY

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Input a negative number	fResearchPrice = -69 *fPizo = 100.00 fPizzaPrice = 2.00 *ncounter = 0	“INVALID INPUT Try again”	“INVALID INPUT Try again”	PASS
Input a number less than the square root of the old research price (see if the price lowers back down)	fResearchPrice = 8 *fPizo = 24.00 fPizzaPrice = 3.00 *ncounter = 0 fOldResearchPrice = 9.00	“You cannot lower your pizza price back down. Input a higher price.”	“You cannot lower your pizza price back down. Input a higher price.”	PASS
Input zero to cancel process	fResearchPrice = 0 *fPizo = 100.00 fPizzaPrice = 2.00 *ncounter = 0	fPizzo: 100.00 fPizzaPrice: 2.00	fPizzo: 100.00 fPizzaPrice: 2.00	PASS

FUNCTION: INT MANAGECOOKS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Making a pizza	DAY 6 Pizza(s): 1 Pizo(s): 5.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 1	DAY 7 Pizza(s): 2 Pizo(s): 6.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	DAY 7 Pizza(s): 2 Pizo(s): 6.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	PASS
Selling a pizza	DAY 6 Pizza(s): 1 Pizo(s): 5.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 2	DAY 7 Pizza(s): 0 Pizo(s): 8.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	DAY 7 Pizza(s): 0 Pizo(s): 8.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	PASS

FUNCTION: INT MANAGECOOKS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Hiring a cook	DAY 6 Pizza(s): 1 Pizo(s): 5.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 3	DAY 7 Pizza(s): 2 Pizo(s): 4.00 Cook(s): 2 Customer(s): 1 Pizza Price: 2.00	DAY 7 Pizza(s): 2 Pizo(s): 4.00 Cook(s): 2 Customer(s): 1 Pizza Price: 2.00	PASS
Hiring a cook but with less pizo	DAY 6 Pizza(s): 1 Pizo(s): 2.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 3	DAY 7 Pizza(s): 1 Pizo(s): 3.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	DAY 7 Pizza(s): 1 Pizo(s): 3.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	PASS

FUNCTION: INT MANAGECOOKS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Research pizza quality	DAY 6 Pizza(s): 1 Pizo(s): 10.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 5 fResearchPrice = 9.00	DAY 7 Pizza(s): 1 Pizo(s): 3.00 Cook(s): 1 Customer(s): 1 Pizza Price: 3.00	DAY 7 Pizza(s): 1 Pizo(s): 3.00 Cook(s): 1 Customer(s): 1 Pizza Price: 3.00	PASS
Skip a day	DAY 6 Pizza(s): 1 Pizo(s): 10.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00 nActiontoPerform = 6 nSkipDays = 1	DAY 7 Pizza(s): 1 Pizo(s): 11.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	DAY 7 Pizza(s): 1 Pizo(s): 11.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	PASS

FUNCTION: FLOAT HANDLECUSTOMERS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Making a pizza	DAY 2 Pizza(s): 3 Pizo(s): 3.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 1	DAY 3 Pizza(s): 4 Pizo(s): 4.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	DAY 3 Pizza(s): 4 Pizo(s): 4.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	PASS
Selling a pizza	DAY 2 Pizza(s): 3 Pizo(s): 3.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 2	DAY 3 Pizza(s): 2 Pizo(s): 6.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	DAY 3 Pizza(s): 2 Pizo(s): 6.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	PASS

FUNCTION: FLOAT HANDLECUSTOMERS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Attracting a customer	DAY 2 Pizza(s): 3 Pizo(s): 3.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 4	DAY 3 Pizza(s): 3 Pizo(s): 4.00 Cook(s): 2 Customer(s): 3 Pizza Price: 2.00	DAY 3 Pizza(s): 3 Pizo(s): 4.00 Cook(s): 2 Customer(s): 3 Pizza Price: 2.00	PASS
Attracting a customer but with less pizza	DAY 2 Pizza(s): 0 Pizo(s): 3.00 Cook(s): 1 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 4	“Customer 2 walked out” DAY 3 Pizza(s): 0 Pizo(s): 4.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	“Customer 2 walked out” DAY 3 Pizza(s): 0 Pizo(s): 4.00 Cook(s): 1 Customer(s): 1 Pizza Price: 2.00	PASS

FUNCTION: FLOAT HANDLECUSTOMERS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Research pizza quality	DAY 2 Pizza(s): 3 Pizo(s): 12.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 5 fResearchPrice = 9.00	DAY 3 Pizza(s): 3 Pizo(s): 6.00 Cook(s): 2 Customer(s): 2 Pizza Price: 3.00	DAY 3 Pizza(s): 3 Pizo(s): 6.00 Cook(s): 2 Customer(s): 2 Pizza Price: 3.00	PASS
Skip a day	DAY 2 Pizza(s): 3 Pizo(s): 3.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00 nActiontoPerform = 6 nSkipDays = 1	DAY 3 Pizza(s): 3 Pizo(s): 4.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	DAY 3 Pizza(s): 3 Pizo(s): 4.00 Cook(s): 2 Customer(s): 2 Pizza Price: 2.00	PASS

SKIPPING DAYS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Skip 3 Days	Day 1 nSkipDays = 3 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3 *ncounter = 0	Day 4 nPizzas = 14 fPizo = 61.00 nCooks = 3 nCustomers = 3	Day 4 nPizzas = 14 fPizo = 61 nCooks = 3 nCustomers = 3	PASS
Input zero	Day 1 nSkipDays = 0 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3 *ncounter = 0	Day 1 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3	Day 1 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3	PASS

SKIPPING DAYS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Input a negative integer	Day 1 nSkipDays = -69 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3 *ncounter = 0	“INVALID INPUT Try again”	“INVALID INPUT Try again”	PASS
Skip 5 Days	Day 1 nSkipDays = 5 nPizzas = 10 fPizo = 69.00 nCooks = 3 nCustomers = 3 *ncounter = 0	Day 6 nPizzas = 17 fPizo = 55.00 nCooks = 3 nCustomers = 3	Day 6 nPizzas = 17 fPizo = 55.00 nCooks = 3 nCustomers = 3	PASS

SKIPPING DAYS

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Skip 5 more days	Day 6 nSkipDays = 5 nPizzas = 17 fPizo = 55.00 nCooks = 3 nCustomers = 3 *ncounter = 0	Day 11 nPizzas = 22 fPizo = 45.00 nCooks = 3 nCustomers = 3	Day 11 nPizzas = 22 fPizo = 45.00 nCooks = 3 nCustomers = 3	PASS
Skip 10 more days	Day 11 nSkipDays = 10 nPizzas = 22 fPizo = 45.00 nCooks = 3 nCustomers = 3 *ncounter = 0	Day 21 nPizzas = 34 fPizo = 21.00 nCooks = 3 nCustomers = 3	Day 21 nPizzas = 34 fPizo = 21.00 nCooks = 3 nCustomers = 3	PASS

INT MAIN

TEST DESCRIPTION	INPUT VALUE/PARAMETERS	EXPECTED OUTPUT/RESULT	ACTUAL OUTPUT/RESULT	PASS OR FAIL
Input a positive integer between 1-7	nActiontoPerform = 7	“Thank you for playing Pizza Picker. Have a great day!”	“Thank you for playing Pizza Picker. Have a great day!”	PASS
Input an integer greater than 7	nActiontoPerform = 8	“INVALID INPUT Try again”	“INVALID INPUT Try again”	PASS
Input an integer less than one	nActiontoPerform = -1	“INVALID INPUT Try again”	“INVALID INPUT Try again”	PASS
Input 0	nActiontoPerform = 0	“INVALID INPUT Try again”	“INVALID INPUT Try again”	PASS
Input a character	nActiontoPerform = k	Infinite Looping	Infinite Looping	PASS