Adolfo Sanpedro Gante

Feb 4th, 2021

Homework 1 : Algorithms

Vending Machine Change:

1. Insert a $1.00 into the machine
2. List all items and prices
3. Prompt user to choose an item and dispense item
4. After user chooses an item calculate the difference ( change = $1.00 - item price)
5. What is the largest coin that can “fit” into “change”?
6. Declare variables int Quarter, Dime, Nickel, Penny
7. If change is larger than 25/10/5/1 cents do (change = change - 25/10/5/1 cents)
8. Update coin variables if their respective if statement is true
9. When change reaches “0” dispense coins bases on coin variables\

Calculating Pi:

1. Declare Pi and denom (denominator) as doubles
2. Rewrite the formula to pi = 4\*( - (1/n) + (1/n+2) - … )+4
3. Declare piCheck used to check whether or not pi is equal to piCheck
4. Create a loop to iterate the above formula many times
5. Split the formula into two parts in the loop each part updates pi
6. The two part are the negative (1/n) and positive (1/n)
7. Make sure to update n by 2 after each part
8. Save the value of pi to piCheck
9. The loop should continue until pi = piCheck
10. Finally print pi to the six digits.

Robot Mower:

1. The lawn mower will have three functions
2. The mower will constantly check if there's a red square in front of it
3. If it detects a red square it will (turn right, check square, move forward, check square turn right, check square move forward, check square)
4. If it detects another red square it will (turn left, check square, move forward, check square, turn left, check square, move forward, check square)
5. There will be count (ex. Count++; )to keep track of the two previous functions so that they always go right after each other depending if (count) is odd or even
6. If the mower detects two red squares consecutively it will turn left or right for a third time depending on the last function and update the variable int (corners) by 1
7. When variable corners equals 4 the program will terminate
8. There is no need for the mower to know its orientation at the beginning

Horse Movers:

1. Variables needed
   1. Number of Horses (horseNum)
   2. Average weight of all horses (avgWeight)
   3. Current weight of the horse on the scale (tempWeight)
   4. Current name of the horse on the scale (tempName)
   5. Name of lightest horse (lightName)
   6. Weight of the lightest horse (lightWeight)
   7. Name of heaviest horse (heavyName)
   8. Weight of the heaviest horse (heavyWeight)
2. Weight one horse at a time
3. After every horse add 1 to horseNum
4. After every horse save the weight to (tempWeight) and add half the (tempWeight) to (avgWeight)
5. When the first horse is weighed save the name to lightest horse to (lightName)
6. When the second horse is weighed compare (tempWeight) to (lightWeight)
7. If (tempWeight) is lighter than (lightWeight) save (tempWeight) to (lightWeight) then save (tempName) to (lightName)
8. If (tempWeight) is heavier than (lightWeight) save (tempWeight) to (heavyWeight) then save (tempName) to (heavyName)
9. Continue to compare (tempWeight) to (lightWeight) and (heavyWeight) and re-write (lightName) and (heavyName) accordingly to the new lightest weight and heaviest weight.
10. The program then prints horseNum, avgWeight, lightName/Weight, heavyName/Weight and terminates when there are no more horses.