Pseudocode - Project 2: Palpatine’s Delivery Service

Function Main

* Connect ifstream to shipments.txt
* char sWeight[20] = {};
* double cost[8], time[8];
* Loop – while !EOF
  + Read line into c-string
  + valid = verifyShipment(cstring)
  + if(valid == 1)
    - for < strlen(cstring)
      * if counter == 0
        + serviceCode = cstring[0]
      * if counter == 2
        + shipCode = cstring[2]
      * if counter == 4
        + delivCode = cstring[4]
    - strcat(weight, cstring+5)
    - weight = atof(sWeight);
    - calcShipment(serviceCode, shipCode, delivCode, weight, cost, time)
* shipReport(cost, time)

Function verifyShipment

* Parameters: shipLine
* Return: valid
* Logic
  + valid = 1, decimal = 0;
  + for loop < strlen(shipLine)
    - if counter == 0
      * if shipLine[0] != F || P || E
        + valid = 0;
        + break;
    - if counter == 1 or 3 or 5
      * if shipLine[counter] != “ “
        + valid = 0;
        + break;
    - if counter == 2 or 4
      * if shipLine[counter] != Y || V || E || M || J || S || U || N
        + valid = 0;
        + break;
    - if counter == 6
      * if isdigit(shipLine[6])
        + for counter = 7; counter < strlen(shipLine)

if shipLine[counter] == ‘.’ && decimal == 0

decimal = 1;

decimalLoc = counter;

* + - * + if isdigit(shipLine[decimalLoc+1])

valid = 0;

break;

* + - * + if decimal != 1

valid = 0;

break;

* + if shipLine[2] == shipLine[4]
    - valid == 0;
    - break;

Function calcShipment

* Parameters: service\_code, ship\_code, deliv\_code, shipWeight, shipCost[], delivTime[]
* Return: none
* Logic
  + if service\_code == ‘F’
    - cost = .25
    - speed = 45000
  + if service\_code == ‘P’
    - cost = .75
    - speed = 95000
  + if(service\_code == ‘E’
    - cost = 1.5
    - speed = 175000

Two arrays storing the distance from the sun and gravity factor with the planets corresponding the values stored at each element in accordance the order closer to farthest from sun i.e distance[0] and gravity[0] hold values for Mercury. And value i will control the values for shipping planet and j for delivery planet

* + if(ship\_code == ‘Y’
    - i = 0;
  + if(ship\_code == ‘V’
    - i = 1
  + if(ship\_code == ‘E’
    - i = 2
  + if(ship\_code == ‘M’
    - i = 3
  + if(ship\_code == ‘J’
    - i = 4
  + if(ship\_code == ‘S’
    - i = 5
  + if(ship\_code == ‘U’
    - i = 6
  + if(ship\_code == ‘N’
    - i = 7
  + if(deliv\_code == ‘Y’
    - j = 0;
  + if(deliv\_code == ‘V’
    - j = 1
  + if(deliv\_code == ‘E’
    - j = 2
  + if(deliv\_code == ‘M’
    - j = 3
  + if(deliv\_code == ‘J’
    - j = 4
  + if(deliv\_code == ‘S’
    - j = 5
  + if(deliv\_code == ‘U’
    - j = 6
  + if(deliv\_code == ‘N’
    - j = 7

shipWeight \*= gravity[j];

shipCost[i] += (shipWeight \* cost);

distBetween = abs(distance[i] – distance[j]);

delivTime[i] += (distBetween / speed);

Function shipReport

* Parameters: shipCost[] , shipTime[]
* Return: none
* Logic:
  + cout << “Planet Name” << setw(10) << “Total Cost” << setw(10) << “Total Time” << endl;
  + For loop < 8
    - cout << planetName[i] << setw(10) << shipCost[i] << setw(10) << shipTime[i] << endl;