Section Flowcharts  
Problem:1

Normal Delivery

Sorting Regular   
Packages

Package = input

Fragile Packages

Urgent Delivery Packages

False

False

True True  
  
  
  
  
  
  
  
Problem 2:

Urgent/Fast Delivery

B == item

NUM = INPUT

Repeat

Until

False

Despense item

True

IF amount >= price

False True

PRINT”Insufficient funds”

**SECTION PSEUDUOCODE  
  
Q:1 Pseudocode for smallest number.**Step 1: fixing three variables a, b, c   
SET a = <1>  
SET b = <2>   
SET c = <3>  
  
Step 2: Comparing variables to find the smallest   
IF a < b AND a < c THEN  
 SET smallest = a   
ELSE IF b < a AND b < c THEN  
 SET smallest = b   
ELSE SET smallest = c   
END IF  
  
Step 3: Output the smallest number   
PRINT "The smallest number is: " + smallest   
END  
  
**Q2: Basic Calculator Pseudocode  
  
Input**: Assign two numbers: ‘num1’ and ‘num2’ and an operator (\* for multiplication, / for division).  
**Output**: Result of the operator.  
  
Step 1:  
 Display ‘The first number’  
Read ‘num1’  
  
Step 2:   
Display ‘The second number’   
Read ‘num2’

Step 3:   
Display ‘Enter an operator (\* for multiplication, / for division):’  
Read ‘operator’

Step 4:  
 If ‘operator’ is ‘\*’  
Calculate ‘result = num1 \* num2’

Display: "The result is: " + ‘result’  
Else If ‘operator’ is ‘/’  
Check if ‘num2’ is not equal to 0:

If true, calculate ‘result = num1 / num2’

Display: "The result is: " + ‘result’

Else:

Display: "Error: Division by zero is not allowed."

Else:

Display: "Error: Invalid operator."  
  
END

**SECTION: ALGORITHM**

Q1: Algorithm for a Prime number.  
**Input**: ‘n’ is a positive integer.  
  
**Output**: If ‘n’ is a prime number ‘True’, else ‘False’.  
**Total calculations**:   
If n is less than or equal to 1, return False (since numbers less than 2 are not prime).  
  
set ‘A = 2’, the smallest prime number.  
  
If ‘n % A == 0’ return ‘False’   
Else   
increment ‘A’ by ‘1’

If no divisors are found in the previous step, return ‘True’  
(hence ‘n’ is a prime number)  
  
Q2: Algorithm for day number.  
  
**Input:** A day number n (where 1 ≤ n ≤ 365).

**Output:** The corresponding day of the week.

**Calculations:**  
 Create a list of days of the week in order: days\_of\_week = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"].  
  
Compute the index of the day of the week by taking the remainder of (n - 1) when divided by 7:

index = (n - 1) % 7  
  
Output: days\_of\_week[index].

Q3: Algorithm to find the Greatest Common Divisor.  
  
**Input:** Two positive integers,’ a’ and ‘b’.

**Output:** The Greatest Common Divisor (GCD) of ‘a’ and ‘b’.  
  
**Calculations:**

Assign ‘a’ to the larger of the two numbers.

Assign ‘b’ to the smaller of the two numbers.

Compute the remainder ‘r’ of ‘a’ divided by ‘b’ (i.e., r = a % b).

Assign the value of ‘b’ to ‘a’.

Assign the value of ‘r’ to ‘b’.  
  
When ‘b’ becomes 0, the value of ‘a’ is the GCD of the original pair of numbers.