**Q2:**

Product not available

PRINT Total

Is Total Price greater than Cash?

INPUT Cash

Want more Product?

Add Product Price to Total

INPUT Product

Is Product Available?

PRINT “Insufficient Funds”

No

Yes

Yes

No

Yes

No

Q1:

Yes

Yes

Deliver Package

Urgent Delivery Package

Urgent Fragile Package

Normal Fragile Package

Normal Delivery Package

Is package fragile

Is package fragile?

Handle Damage Package

Is package Urgent?

Is package Damaged?

Scan Package

Receive Package

Yes

No

Yes

Yes

No

No

No

Yes

No

No

Yes

**PSEUDOCODES**

Q1. Write pseudocode to find the smallest number among three given variables. Implement a decision-making structure to compare the variables

Ans:

START

INPUT num1

INPUT num2

INPUT num3

SET smallest to num1

IF num2 < smallest THEN

SET smallest to num2

ENDIF

IF num3 < smallest THEN

SET smallest to num3

ENDIF

OUTPUT smallest

END

Q2. Develop pseudocode for basic calculator that performs multiplication and division. The pseudocode should prompt the user for two numbers and an operator, then display the result of the operation

Ans:

START

INPUT num1

INPUT operator

INPUT num2

IF operator = “\*” THEN //Assuming the operator datatype is STRING, otherwise would have directly

SET answer to num1 \* num2 //wrote ---- answer = num1 operator num2

ELSE

IF operator = “/” THEN

SET answer to num1 / num2

ENDIF

ENDIF

OUTPUT answer

END

**ALGORITHMS**

Q1. Write an algorithm to determine whether a number is a prime number. The algorithm should iterate through possible divisors and determine if the number has any divisors other than 1 and itself

Ans:

1. Ask the user to enter **Number**
2. Set the **Count** to **2**
3. Set **Divisor** to **1**
4. Increment **Divisor**
5. If **Divisor** is equal to **Number**, Jump to **step** **9**
6. Set **Remainder** to the remainder obtained when (**Number** / **Divisor**)
7. If **Remainder** is **0**, Increment **Count** and Print **Divisor**
8. Jump to **step** **4**
9. If **Count** is **2**, Print “Number is prime number”
10. End Program

Q2. Create an algorithm that asks the user for a day number (1-365) and outputs the corresponding day of the week, assuming that January 1st is a Monday.

Ans: METHOD 1 (Long and Inefficient)

1. Ask the user to enter **Day** between 1-365
2. Set **Remainder** to (**Day** **MOD** **7**) //MOD is a function which returns the remainder when

//(Day / 7)

1. If **Remainder** is 1, Print Monday
2. If **Remainder** is 2, Print Tuesday
3. If **Remainder** is 3, Print Wednesday
4. If **Remainder** is 4, Print Thursday
5. If **Remainder** is 5, Print Friday
6. If **Remainder** is 6, Print Saturday
7. If **Remainder** is 7, Print Sunday
8. End Program

METHOD 2 (Shorter and Efficient)

1. Create a list named **Days**, containing days of the week starting with **Monday** and index **1**
2. Ask the user to enter **day** between 1-365
3. Set **Remainder** to **day** MOD **7**
4. Print **Days**[**Remainder**] //prints the day at index **Remainder** present in the list **Day**

Q3. Develop an algorithm for a program that takes two numbers as input and finds the Greatest Common Divisor (GCD) of the two numbers using the Euclidean algorithm.

Ans:

1. Ask user for the two number as inputs
2. Pick out the greater number
3. Divide the greater number by smaller number
4. If the remainder is zero, Output the divisor or preceding remainder (this will be GCD). Jump to step 6
5. If the remainder is not zero, divide the divisor by the remainder. Go to step 4
6. End the Program