1. You are working in a logistics company responsible for delivering packages. Design a flowchart to manage the process of receiving, sorting, and delivering packages. Include decision structures for handling fragile items and urgent deliveries.

IS THE PACKAGE URGENT?

IS THE PACKAGE FRAGILE?

PRINT PACKAGE DELIVERED

YES

NO

NO

YES

ASSIGN PACKAGE TO STANDARD DELIVERY TEAM.

ASSIGN PACKAGE TO URGENT DELIVERY TEAM

PROCEED TO STANDARD HANDLING.

PROCEED TO FRAGILE HANDLING. LABEL AS FRAGILE. HANDLE WITH CARE.

START

SORTING PACKAGE

RECEIVE PACKAGE

2. Imagine you are automating the process of a vending machine. Create a flowchart that includes decision points for user input, selecting products, accepting payment, and dispensing the correct item. Include error-handling for invalid inputs and insufficient funds.

“OUT OF STOCK”

“INVALID SELECTION”

YES

NO

A

ACCEPT PAYMENT

IS PRODUCT IN STOCK?

CHECK INVENTORY

IS PRODUCT SELECTION VALID?

SELECT PRODUCT

START

DISPLAY WELCOME

A



“INSUFFICIENT FUNDS”

END

RETURN CHANGE (IF ANY)

DISPENSE ITEM

IS PAYMENT SUFFICIENT?

Pseudocode

1. Write pseudocode to find the smallest number among three given variables. Implement a

decision-making structure to compare the variables.

1. Start
2. // Input/Output
3. INPUT num1
4. INPUT num2
5. INPUT num3
6. // Process Steps
7. Print num1, num2 , num3
8. 13 // Conditional Statements
9. IF (num1<num2&&num1<num3) THEN
10. PRINT "The smallest number is num1"
11. ELSE IF (num2<num1&&num2<num3) THEN
12. PRINT " The smallest number is num2”
13. ELSE (num3<num1&&num3<num2) THEN
14. PRINT " The smallest number is num3”
15. End

3. Develop pseudocode for a 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.

1. Start
2. Print “Basic Calculator”
3. Input num1
4. Input num2
5. Print “Enter the operator (\*, /):”
6. Read operator
7. If (operator==\*) THEN
8. Result=num1\*num2
9. Else (operator==/) THEN
10. Result=num1/num2
11. Print “Result”
12. End

ALGORITHM

1. 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.
2. Start
3. Ask the user to enter a “n” value to check prime or not
4. Set I =1, count=0.
5. If I<=n, if true go to step 5 else goes to step 8
6. Check the condition n % I==0 if true then evaluate step 6, false go to step 7
7. Set count=count+1
8. I = I+1 go to step 4
9. Check count, if count=2 display prime, if not it is not prime
10. Stop
11. 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.

1. Start
2. Ask the user to enter “Day number (1-365)”
3. Read Day number
4. Set Day of week= (Day number - 1) % 7
5. Day of week = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
6. Display the corresponding day of week
7. Stop
8. 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.
9. Start
10. Ask the use to enter number “a”
11. Ask the use to enter number “b”
12. Read a and b
13. While b! = 0
14. r = a % b
15. a=b
16. b=r
17. Return a as GCD
18. Stop