* FLOWCHART:

1. You are working in a logistics company responsible for delivering packages. Design a flowchart.

START

PRINT “NOT AVAILABLE”

END

PRINT “DELIVER”

SORT THE ITEM

IF PROD

AVAILABLE

RECEIVE THE PRODUCT

FALSE

PRINT “DELIEVER WITH CARE”

PRINT “DELIVER URGENTLY”

IF URGENT DELIVERY

IF ITEM IS FRAGILE

FALSE FALSE

TRUE TRUE

1. 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.

START

INPUT ITEM

INPUT PAYMENT

IF ITEM AVAILABLE

ITEM 03

ITEM 02

ITEM 01

SELECT ITEM

FALSE

FALSE FALSE

TRUE TRUE TRUE

NO

YES

INPUT PAYMENT

IF THE PAYMENT IS SUFFICIENT

YES NO

ACCEPT AND PROCEED

DISPENSE AND RETURN THE EXTRA AMOUNT

END

INSUFFICIENT AMOUNT

* PSEUDO CODE:

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

START

// INPUT

INPUT NUMBER 1

INPUT NUMBER 2

INPUT NUMBER 3

// INITIONALIZATION

SET NUMBER 1 TO ‘a’

SET NUMBER 2 TO ‘b’

SET NUMBER 3 TO ‘c’

// CONDITIONAL STATEMENTS

IF a>b && a >c THEN

PRINT “NUMBER 1 IS GREATER”

ELSE IF b>a && b >c THEN

PRINT “NUMBER 2 IS GREATER”

ELSE IF c>a && c >b THEN

PRINT “NUMBER 3 IS GREATER”

ELSE

PRINT “ INVALID INPUT”

END

1. 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.

START

//OUTPUT

PRINT “ENTER TWO NUMBERS”

PRINT “ENTER THE OPERATOR(\*,/)”

//INTIALIZATION

SET NUMBER 1 TO ‘a’ , NUMBER 2 TO ‘b’, OPERATOR TO ‘OP’

SET PRODUCT TO 0, DIVIDE TO 0

// PROCESS STEPS

SET PRODUCT TO a\*b

SET DIVIDE TO a/b

// CONDITIONAL STATEMENTS

IF OP=\* THEN

PRINT THE VALUE STORED IN PRODUCT

ELSE IF OP=/ THEN

PRINT THE VALUE STORED IN DIVIDE

ELSE

PRINT “ THE INPUT IS INVALID”

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. ASK THE USER TO ENTER A NUMBER.
3. ITERATE ‘FOR LOOP’.
4. SET VARIABLE ‘i’ TO 1.
5. CHECK IF ‘i’ IS SMALLER OR EQUAL TO THE INPUT NUMBER. (i<=n)
6. IF THE CONDITION IS FALSE THEN,
7. CONTINUE ITERATION TILL THE CONDITION IS TRUE.
8. IF THE CONDITION IS TRUE THEN,
9. CHECK IF THE REMAINDER OF INPUT NUMBER AND ‘i’ IS 0. (n%i==0)
10. IF THE CONDITION IS TRUE THEN,
11. DISPLAY “THE NUMBER IS A PRIME NUMBER”
12. IF THE CONDITION IS NOT TRUE THEN,
13. DISPLAY “THE NUMBER IS NOT A PRIME NUMBER”
14. 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.
15. ASK THE USER TO ENTER A NUMBER(1-365).
16. SET AN INDEX.

{MONDAY = 1,TUESDAY = 2,Wednesday = 3,Thursday = 4,Friday = 5 ,Saturday =6 ,Sunday = 7}

1. SET DAY TO ( NUMBER%7)
2. THE NUMBER STORED IN DAY WILL BE CORRESPONDING WITH THE INDEX.
3. DISPLAY THE DAY IN CORRESPONDACE WITH THE NUMBER IN THE INDEX.
4. 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.
5. Ask The user to enter two numbers ‘a’ and ‘b’.
6. Check if ‘b’ is zero:
7. If true, return ‘a’ as the GCD.
8. If false, proceed to the next step.
9. Replace ‘a’ with ‘b’ and ‘b’ with the remainder of ‘a’ divided by ‘b’.
10. Repeat the process until ‘b’ becomes zero.
11. When b=0, the value of ‘a’ is the GCD
12. Display ‘a’ as the GCD.