PSEUDOCODE

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

Start

INPUT “num1, num2, num3”

If

num1< num2 AND num1 < num3

Then

PRINT “num1 is smallest”

ELSEIF

num2 < num3 AND num2 < num1

Then

PRINT “num2 is smallest”

ELSE

PRINT “num3 is smallest”

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

Input “num1, num2”

Print “ enter operator”

If

operator== “\*”

then product = num1\*num2

print “product”

elseif

operator == “/”

then product = num1/num2

print “product”

else

print “invalid input”

end

FLOWCHART

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.

PRINT “please enter the package”

PRINT “is the package fragile”

Is package fragile?

Fragile delivery

yes

no

PRINT “is the package urgent?”

no

If package ==urgent

yes

Yyesyes

Urgent delivery

Normal delivery

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.

INPUT “product”

If product ==num

no

yes

Select product

PRINT “enter product price”

INPUT price

If price < product price

no yes

PRINT “product”

PRINT” insufficient funds"

ALGORITHMS

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.

1. Start
2. Ask the user to enter a number “n”
3. If n <=1, then it is not a prime number
4. If number is 2 or 3, it is a prime number
5. If number >3 and is divisible by 2 0r 3, then it is not a prime number.
6. Iterate for loop, initialize i=3.
7. If n is divided by i and it gives remainder 0, go to step 10.
8. If no, increment i=i+1 until i<n, go to step 7.
9. If number n=i, go to step 11.
10. The number “n” is not a prime number.
11. The number “n” is a prime number.
12. End

2. 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. Input a day number
3. If day number<1 or day number >365
4. Print that the number in invalid
5. Days of week= {“Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”, “Sunday”}
6. Divide the day number by 7
7. Note down the remainder
8. Print the day of the week corresponding to the remainder
9. If remainder==0, then day is Sunday
10. Print day of the week
11. End

3. 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.

1. Start
2. Ask the user to enter two positive integers
3. If the integers are negative, print “invalid input”.
4. Input a, b
5. If a<b, then swap values of a and b.
6. Divide a by b, then calculate the remainder “r”
7. Replace a by b and b by remainder “r”
8. Repeat from step 6 until b=0,
9. if b=0, greatest common divisor (GCD) = a
10. print GCD
11. end