

24K-0810

PSEUDOCODES: LAB02

Q: Develop a pseudocode for a basic calculator that performs * and /

Start

Output "Enter first number:"

Input num1

Output "Enter second number:"

Input num2

Output "Enter operator (* or /):"

Input Operator

IF operator == *

Result = num1 * num2

Elseif ~~Output~~ operator == /

If ~~num1~~ num2 == 0

Result = "error"

Else

Result = num1 / num2

Output "Result"

End

Q: Pseudo code for subtraction using addition and complement.

Start

Input int(a)

input int(b)

ADD int(a) + int(-b)

Result = a + (-b)

Output "Result"

End

variable

Q: Write pseudocode to find the smallest number among 3 given ↑

- Start
- Input num1, num2, num3
- IF num1 \geq num2
 Smaller = num2
- Else Smaller = num1
- IF Smaller \geq num3
 Smallest = num3
- Else Smallest = Smaller
- Output "Smallest number:", Smallest
- Exit

Q: Algorithm that corresponds the day of week to a number.

- 1) Ask the user for a number (1-365). In case of leap year, ask for a number (1-366). Input = x.
- 2) Mark 1-7 numbers with days of week with 1 being Monday and 7 as Sunday. Display the day that corresponds to that number.
- 3) Start from Monday again after every 7 numbers.
- 4) For every input greater than 7, take $x \% 7 = \text{remainder}$.
The remainder (1-6) correspond to the days of the week.
- 5) If $x \% 7 = 0$, the day is 7 and is taken as Sunday.
- 6) Display the day to user.

Q: Write an algorithm to determine whether a number is prime or not.

- 1) ~~Take~~ the user for a number 'n'.
- 2) Check if number is less than 2. If yes, then display "not prime".
- 3) Iterate from n to the square root of n. $I \leftarrow 2$. $I \leftarrow I+1$
- 4) For each iteration check if $n \% I = 0$.
- 5) If modulus is zero, Display "not prime".
- 6) If the loop completes without finding a divisor, display "prime".