CPS – ALOGORITHIM’S

CB.SC.U4CSE25361

NAGA RITHWIK VALLALA

2025

CPS – ALGORITHIMS FOR THE GIVEN CODES

PROGRAME - 01

* ALGO- STATEMENT: Printing of an address
* INPUT: Name, Street, City, State, Postal code, Country
* OUTPUT: Street, City, State, Postal code, Country
* Step 0: START
* Step1: Taking the name of the user
* Step 2: Asking the user give his/her street name.
* Step 3: Asking the user give the city, state, postal code, country similarly.
* Step 4: Giving an address based on the given input neatly.
* Step 5: END

PROGRAME – 02

* ALGO-STATEMENT: Converting KM to Miles (vice versa)
* INPUT: Asking the user to give values for conversion and the type of conversion
* OUTPUT: Gives the output based on the given values and the type of conversion
* Step0: START
* Step1: creating a loop for providing the type of conversion to choose by the user.
* Step 2: Asking the user to Choose the type of converter
* Step3: If kilometers to miles conversion type of converter is taken then the following mathematical algorithm can be used

KM= Miles \*0.62137119

* Step 4: Printing the output based on the Obtained value.
* Step5: In case if they choose a type of converter where the mines are being converted into kilometers Then based on the following mathematical algorithm we can proceed with the further calculation

Miles = miles/0.2137119

* Step6: Tinting the output based on the obtained value by round figure it.
* Step7: Apart from these two converters if the user is trying to choose any type of converter, then we can print Please enter the perfect choice.
* Step8: END

PROGRAME-03

* **algo statement**: Temperature Converter (Fahrenheit to Celsius and vice versa)
* **Input**: User choice (a, b, or q), temperature value in Fahrenheit or Celsius
* **Output**: Converted temperature value
* **Steps**:
* **Step 1**: Start
* **Step 2**: Repeat the following steps until the user chooses to quit
* **Step 3**: Display menu:
* a) Fahrenheit to Celsius   b) Celsius to Fahrenheit   q) Quit
* **Step 4**: Read user input as choosing
* **Step 5**: If choosing is "q", display exit message and stop loop
* **Step 6**: If choosing is "a":
* a. Read temperature in Fahrenheit
* b. Convert to Celsius using formula: C = (F - 32) × 5/9
* c. round result to 3 decimal places
* d. Display result
* **Step 7**: Else if choosing is "b":
* a. Read temperature in Celsius
* b. Convert to Fahrenheit using formula: F = (C × 9/5) + 32
* c. round result to 3 decimal places
* d. Display result
* **Step 8**: Else, display error message: “please enter correct choice”
* **Step 9**: End

PROGRAME -04

* **algo statement**: Redesigned Calculator
* **Input**: Two integers (num1, num2), one mathematical operator (+, -, \*, /) or additional operator (//, %)
* **Output**: Result of the selected arithmetic operation
* **Steps**:
* **Step 1**: Start
* **Step 2**: Read num1
* **Step 3**: Read num2
* **Step 4**: Read Mathematical Operator
* **Step 5**: If Mathematical Operator is +, compute and display num1 + num2
* **Step 6**: Else if Mathematical Operator is -, compute and display num1 - num2
* **Step 7**: Else if Mathematical Operator is \*, compute and display num1 \* num2
* **Step 8**: Else if Mathematical Operator is /, compute and display num1 / num2
* **Step 9**: Else, read additional Operator
* **Step 10**: If additional Operator is //, compute and display num1 // num2
* **Step 11**: Else if additional Operator is %, compute and display num1 % num2
* **Step 12**: Else, display error message “please enter correct choice”
* **Step 13**: End

PROGRAME-05

* **algo statement**: Find the square root or cube of a number
* **Input**: A number num1, and user choice (a or b)
* **Output**: Square root or cube of num1
* **Steps:**
* **Step 1**: Start
* **Step 2**: Display menu:
* a) square root of a number   b) Cube of a number
* **Step 3**: Read user input as choosing
* **Step 4**: Read the number
* **Step 5**: If choosing is "a":
* a. Compute square root using √num1 = math.sqrt(num1)
* b. Display result
* **Step 6**: Else if choosing is "b":
* a. Compute cube using num1³ = math.pow(num,1/3)
* b. Display result
* **Step 7**: Else, display error message: “please enter correct choice”
* **Step 8**: End

PROGRAME-06

* **algo statement**: Find the biggest of two numbers
* **Input**: Two integers num1 and num2
* **Output**: The greater number or a message indicating equality
* **Steps**:
* **Step 1**: Start
* **Step 2**: Read num1
* **Step 3**: Read num2
* **Step 4**: If num1 > num2, display "num1 is greater than num2"
* **Step 5**: Else if num1 < num2, display "num1 is less than num2"
* **Step 6**: Else, display "Both numbers are equal"
* **Step 7**: End

PROGRAME-07

* **algo statement**: Find the roots of a quadratic equation
* **Input**: Coefficients a, b, and c
* **Output**: Roots of the equation ax2+bx+c=0ax^2 + bx + c = 0
* **Steps**:
* **Step 1**: Start
* **Step 2**: Read coefficient a
* **Step 3**: Read coefficient b
* **Step 4**: Read coefficient c
* **Step 5**: Compute discriminant D = b² - 4ac
* **Step 6**: If D > 0:
* a. Compute root1 = (-b + √D) / (2a)
* b. Compute root2 = (-b - √D) / (2a)
* c. Display both roots (real and distinct)
* **Step 7**: Else if D == 0:
* a. Compute root = -b / (2a)
* b. Display the root (real and equal)
* **Step 8**: Else (D < 0):
* a. Compute real\_part = -b / (2a)
* b. Compute imag\_part = √(-D) / (2a)
* c. Display complex roots:
* root1 = real\_part + imag\_part \* i       root2 = real\_part - imag\_part \* i
* **Step 9**: End