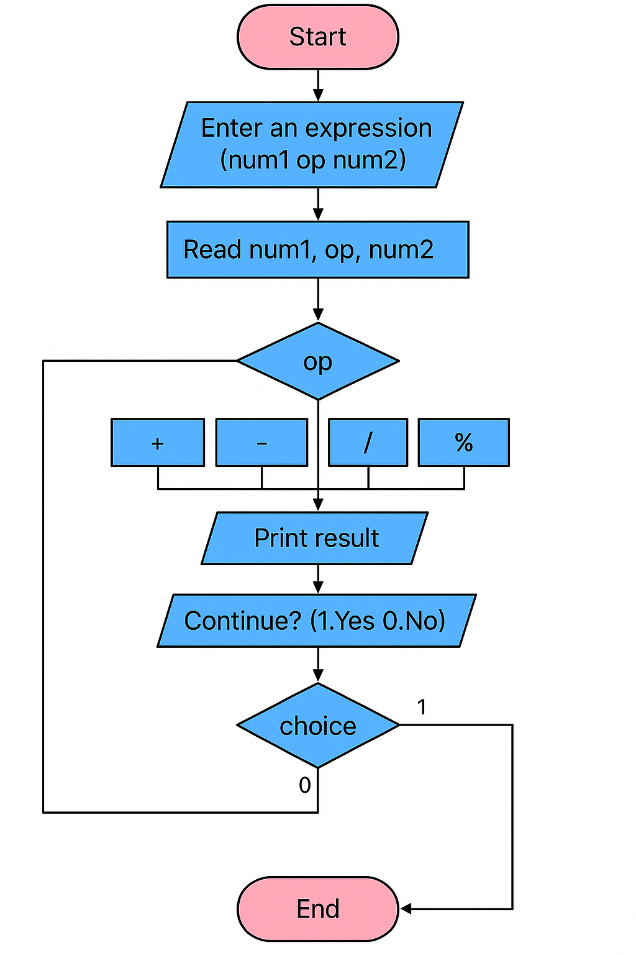
**1. Simple Calculator Program**

**Algorithm:**

1. Start
2. Declare num1, num2, choice, and op
3. Loop until the user chooses to quit:
   * Prompt user to enter an expression (num1 op num2)
   * Read the input values
   * Perform operation based on op:
     + + → Addition
     + - → Subtraction
     + \* → Multiplication
     + / → Division
     + % → Modulus
   * Print the result
   * Ask the user if they want to continue
4. If the user enters 0, exit the loop.
5. End

**Flowchart:**

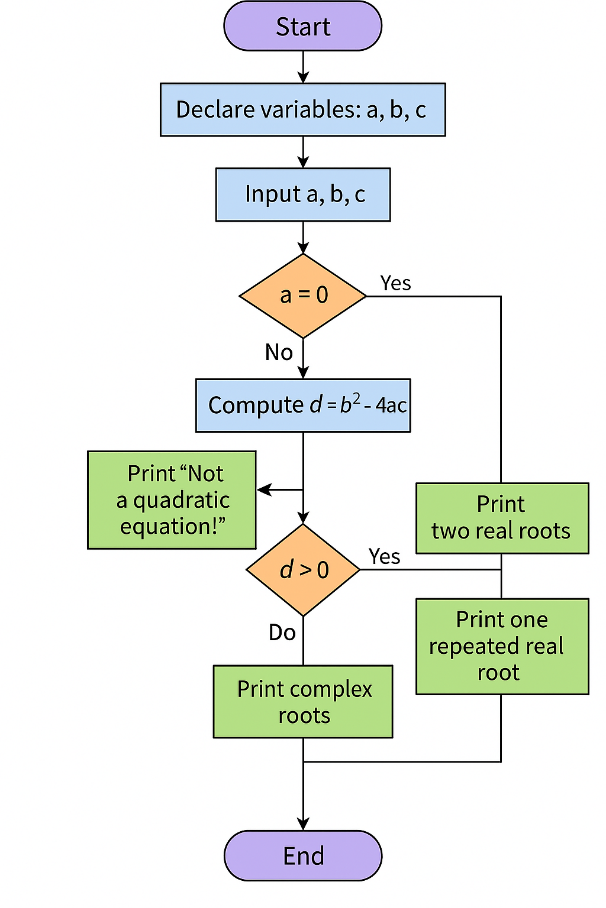
****

**2. Quadratic Equation Solver**

**Algorithm:**

1. Start
2. Declare variables: a, b, c, d
3. Prompt user to enter coefficients a, b, c
4. Read input values
5. If a == 0, print "Not a quadratic equation!" and exit
6. Compute the discriminant: d = b² - 4ac
7. If d > 0, compute and print two real roots
8. If d == 0, compute and print one repeated real root
9. If d < 0, compute and print complex roots
10. End

**Flowchart:**



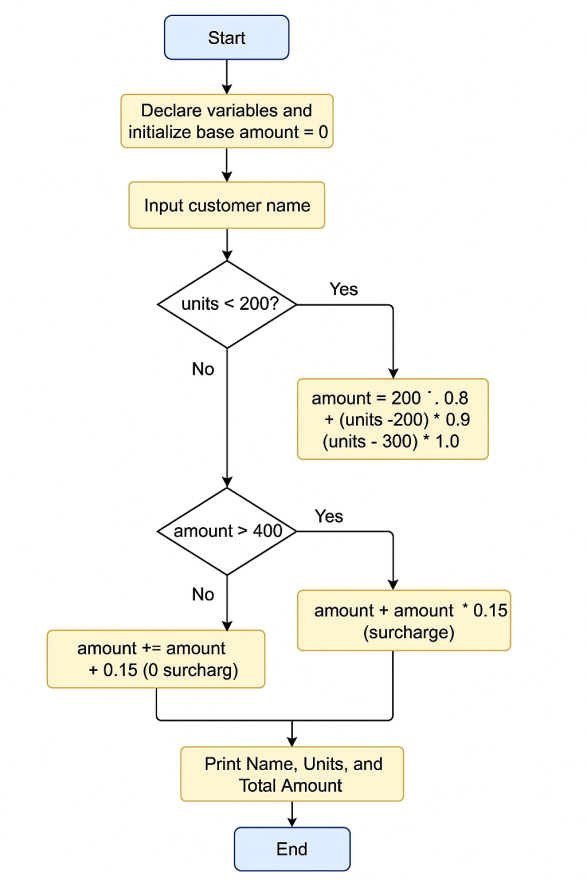
**3 .Electricity Bill**

An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges

**Algorithm: Electricity Bill Calculation**

1. **Start**
2. Declare variables: name, units, bill
3. Set bill = 100 (minimum meter charge)
4. Prompt the user to enter name
5. Read the name
6. Prompt the user to enter units
7. Read the units
8. **If** units <= 200  
   → bill += units \* 0.80
9. **Else if** units <= 300  
   → bill += 200 \* 0.80 + (units - 200) \* 0.90
10. **Else**  
    → bill += 200 \* 0.80 + 100 \* 0.90 + (units - 300) \* 1.00
11. **If** bill > 400  
    → bill += bill \* 0.15 (add 15% surcharge)
12. Display the customer name, units, and final bill
13. **End**

**Flowchart:**



**4. Write a C Program to display the following by reading the number of rows as input,**

**1**

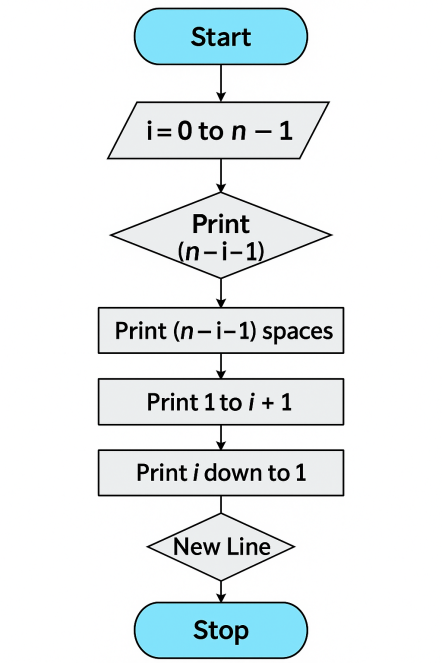
**1 2 1**

**1 2 3 2 1**

**1 2 3 4 3 2 1 --------------------------- nth row**

**Algorithm:**

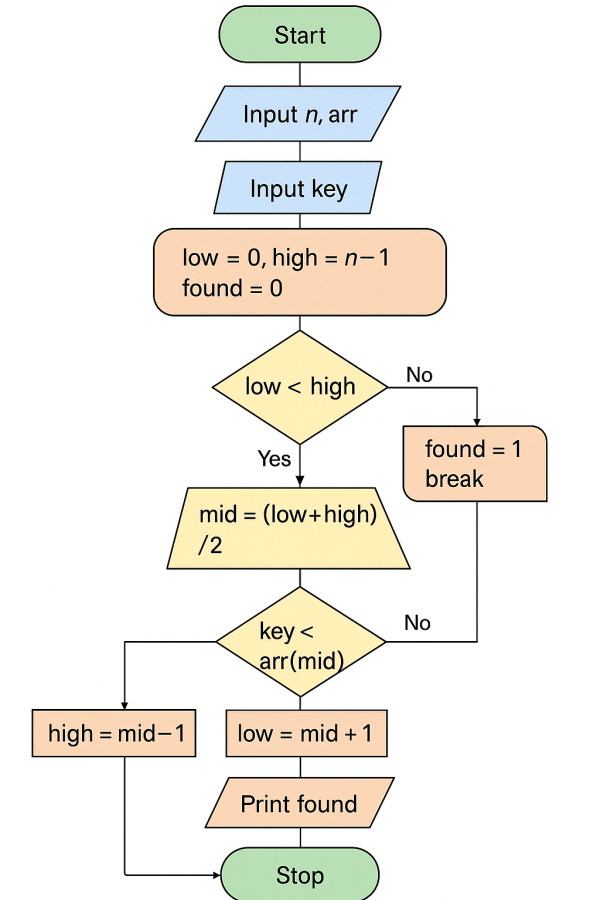
1. **Start**
2. **Input** number of rows n
3. **Repeat for** i = 0 to n - 1 (row index):
   * Print (n - i - 1) spaces
   * Print numbers from 1 to i + 1
   * Print numbers from i down to 1
   * Move to the next line
4. **Stop**



**5. Implement Binary Search on Integers.**

**Algorithm:**

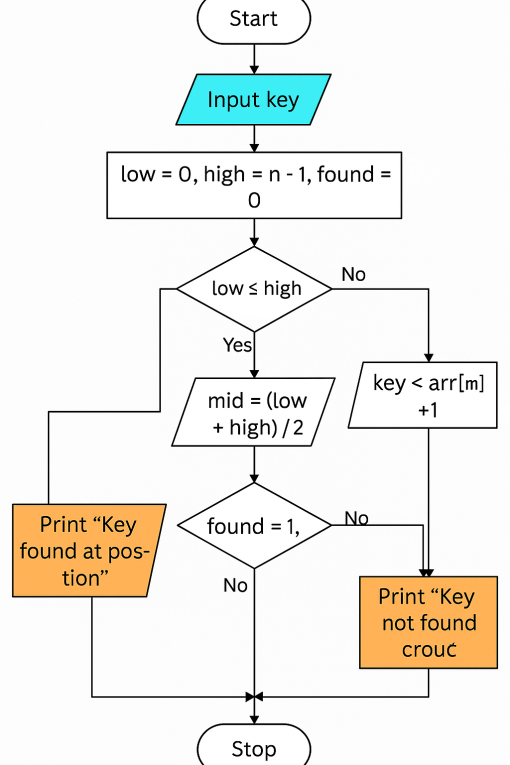
1. Start
2. Input the number of elements (n)
3. Input array elements in ascending order
4. Input key to be searched
5. Initialize low = 0, high = n - 1, found = 0
6. Repeat while low <= high
   * Calculate mid = (low + high)/2
   * If arr[mid] == key, set found = 1, break loop
   * Else if key < arr[mid], set high = mid - 1
   * Else set low = mid + 1
7. If found == 1, print position
8. Else, print "Key not found"
9. Stop



1. **Implement Matrix multiplication and validate the rules of multiplication.**

**Algorithm:**

1. Start
2. Input dimensions of Matrix 1 (m × n)
3. Input dimensions of Matrix 2 (p × q)
4. If n != p, print "Multiplication not possible", exit
5. Input elements of Matrix 1
6. Input elements of Matrix 2
7. Initialize result matrix to 0
8. For each row i in Matrix 1
   * For each column j in Matrix 2
     + For each element k, compute result[i][j] += mat1[i][k] \* mat2[k][j]
9. Print result matrix
10. Stop

****