ALGORITHM / PROCEDURE

**Set 1:**

1. **Area of Triangle**

* Start
* Read base and height
* Compute area = 0.5 × base × height
* Print area
* End

1. **Area of Square Using Macro**

* Start
* Define macro: #define AREA(x) ((x)\*(x))
* Read side of square
* Use macro to compute area
* Print area
* End

**Set 2:**

1. **Temperature Conversion**

* Start
* Read temperature and unit (C or F)
* If unit is C, convert to F using: F = C × 9/5 + 32
* If unit is F, convert to C using: C = (F - 32) × 5/9
* Print result
* End

1. **File Copy Program**

* Start
* Create and open file1 for writing
* Take input until '@' is entered
* Create and open file2 for writing
* Copy contents from file1 to file2
* Display contents of file2
* End

**Set 3:**

1. **Find Largest of Three Numbers**

* Start
* Read three numbers
* Compare them using if-else
* Print the largest number
* End

1. **Store Student Data in Structure**

* Start
* Define structure for student with name, roll, marks
* Input details for 5 students
* Display the entered data
* End

**Set 4:**

1. **Weekday Using Switch-Case**

* Start
* Read weekday number (1–7)
* Use switch-case to print corresponding name
* End

1. **Find Top Performer by Salary**

* Start
* Define structure with ID, Name, Salary
* Read details for 3 employees
* Compare salaries and find max
* Print the details of highest paid employee
* End

**Set 5:**

1. **Armstrong Number**

* Start
* Read a number
* Find sum of cubes of its digits
* Compare sum with original number
* Print whether Armstrong or not
* End

1. **Linear Search Using Pointers**

* Start
* Read array size and elements
* Read element to search
* Use pointer to traverse array
* If found, print position; else print not found
* End

**Set 6:**

1. **Sum of Squares of First N Natural Numbers**

* Start
* Read N
* Initialize sum = 0
* Loop from 1 to N, add square of each number
* Print sum
* End

1. **Swap Two Integers Using Pointers**

* Start
* Read two integers
* Call swap function with addresses
* Swap values using a temporary variable
* Print swapped values
* End

Let me know if you need the algorithms for **Sets 7–14** too.

Here are the **algorithms for Set 7 to Set 11** as per your images:

**Set 7:**

1. **Check Prime Number**

* Start
* Read number
* If number ≤ 1, it's not prime
* Loop from 2 to sqrt(number):
  + If number is divisible by any, it's not prime
* Else, it's prime
* End

1. **String Operations (Length, Compare, Concatenate)**

* Start
* Read two strings
* Use string functions:
  + strlen() to find length
  + strcmp() to compare
  + strcat() to concatenate
* Print results
* End

**Set 8:**

1. **Multiplication Tables for 1 to n**

* Start
* Read n
* For i = 1 to n:
  + For j = 1 to 10: print i × j
* End

1. **Check Palindrome (Ignore Non-Alphabets & Case)**

* Start
* Read string
* Remove non-alphabets and convert to lowercase
* Reverse the cleaned string
* Compare original and reversed
* If same: print "Palindrome", else "Not Palindrome"
* End

**Set 9:**

1. **Decimal to Octal**

* Start
* Read decimal number
* Use division-by-8 method to convert
* Store remainders and print in reverse
* End

1. **Recursive Factorial**

* Start
* Read n
* If n == 0 or 1, return 1
* Else, return n × factorial(n - 1)
* End

**Set 10:**

1. **Sum of Array Elements**

* Start
* Read array size and elements
* Initialize sum = 0
* Loop through array and add elements to sum
* Print sum
* End

1. **Matrix Multiplication**

* Start
* Read matrix sizes
* Check if columns of A == rows of B
* If not, print "Not Possible"
* Else, perform multiplication
* Print result matrix
* End

**Set 11:**

1. **Average of 5 Integers**

* Start
* Read 5 integers
* Add all values
* Compute average = sum / 5
* Print average with 2 decimal places
* End

1. **Linear Search (Without Pointer Version)**

* Start
* Read array and key
* Loop through array to search for key
* If found, print position
* Else, print "Not Found"
* End

SAMPLE OUTPUTS

**Set 1**

**1. Area of Triangle**   
 *Input:* base = 10, height = 5   
 *Output:* Area = 25.00

**2. Area of Square using Macro**   
 *Input:* side = 4   
 *Output:* Area = 16

**Set 2**

**1. Temperature Conversion**   
 *Input:* 100 C   
 *Output:* Temperature in Fahrenheit: 212.00

**2. File Copy**   
 *Input:* Hello World@   
 *Output:*   
 File2 contents:   
 Hello World

**Set 3**

**1. Largest of Three**   
 *Input:* a=12, b=45, c=23   
 *Output:* Largest = 45

**2. Student Structure**   
 *Input:* Names, Rolls, Marks of 5 students   
 *Output:* List of students with details

**Set 4**

**1. Weekday Name**   
 *Input:* 3   
 *Output:* Wednesday

**2. Highest Paid Employee**   
 *Input:* Salaries: 20000, 45000, 30000   
 *Output:* Highest salary: 45000

**Set 5**

**1. Armstrong Number**   
 *Input:* 153   
 *Output:* 153 is an Armstrong number

**2. Linear Search using Pointer**   
 *Input:* Array = [2, 4, 6, 8], Key = 6   
 *Output:* Element found at position 3

**Set 6**

**1. Sum of Squares**   
 *Input:* N = 4   
 *Output:* Sum = 30

**2. Swap using Pointers**   
 *Input:* a = 5, b = 10   
 *Output:*   
 Before Swap: a=5, b=10   
 After Swap: a=10, b=5

**Set 7**

**1. Prime Check**   
 *Input:* 29   
 *Output:* 29 is a Prime Number

**2. String Operations**   
 *Input:* "hello" and "world"   
 *Output:*

* Lengths: 5, 5
* Comparison: Not equal
* Concatenation: helloworld

**Set 8**

**1. Multiplication Table**   
 *Input:* n = 3   
 *Output:*

1 x 1 = 1

...

3 x 10 = 30

**2. Palindrome Check**   
 *Input:* "A man, a plan, a canal, Panama"   
 *Output:* Palindrome

**Set 9**

**1. Decimal to Octal**   
 *Input:* 83   
 *Output:* Octal = 123

**2. Recursive Factorial**   
 *Input:* n = 5   
 *Output:* Factorial = 120

**Set 10**

**1. Sum of Array Elements**   
 *Input:* [1, 2, 3, 4, 5]   
 *Output:* Sum = 15

**2. Matrix Multiplication**   
 *Input:*   
 Matrix A (2x2): 1 2 | 3 4   
 Matrix B (2x2): 5 6 | 7 8   
 *Output:*

19 22

43 50

**Set 11**

**1. Average of 5 Integers**   
 *Input:* 2, 4, 6, 8, 10   
 *Output:* Average = 6.00

**2. Linear Search (Non-Pointer)**   
 *Input:* Array = [3, 6, 9], Key = 9   
 *Output:* Element found at position 3