**ALGORITHMS**

1. **Write an algorithm for finding a palindrome. (Use recursive method).**

Step 1: Start

Step 2: Consider a word or string “word”

Step 3: Initially keep “word” equal to zero

Step 4: Read rev = reverse function

In function:

* Repeat until “word” is not equal to zero. Compute:
* word is equal to word \* 10 + word % 10
* word is equal to word / 10

Return word

Call the function

Step 5: Check rev is equal to “word”

If equals: Print “word” is a palindrome

Not equals: Print “word” is not a palindrome

Step 6: Stop

1. **Write an algorithm for generating anagrams for user entered input.**

Step 1: Start

Step 2: Take a function print Anagrams with parameters prefix and word

Step 3: Take if condition to check whether word’s length is equal to 1 or not

Step 4: If yes, check condition words that contains prefix + word

Step 5: Print the prefix and word

Step 6: If no, take a loop and check the condition should be less than the word length

Step 7: Then, take a variable ‘current’ of type string and add word with its substring (I, i+1)

Step 8: Take another variable ‘before’ of type string and add word with its substring (0, i)

Step 9: Take other variable ‘after’ of type string and add word.substring (i+1)

Step 10: Call the function print Anagrams with parameters prefix + current, before + after

Step 11: Close the loop and exit.

1. **Write an algorithm for binary search.**

Step 1: Take an array of n elements with values

Step 2: Sort the array and add that array to a variable X

Step 3: Set left index to zero and right index to n-1

Step 4: If left index is greater than right index, the search terminates unsuccessful

Step 5: Set middle element position to the largest integer of (left index + right index) /2

Step 6: If array of element is less than X, set left index to middle element + 1 and repeat step 4

Step 7: If array of element is greater than X, set right index to middle element – 1 and repeat step 4

Step 8: Represent array of element is equal to X, the search is completed, return to the middle element.