**Algorithms Tasks**

**Algorithm for finding palindrome:**

**Step 1:** Scan the input from user and store the string in variable‘s’

**Step 2:** Call isPalindrome () function with a parameter s and the function should return the boolean value

**Step 3:** In the function, first check the length of the string. if the string length is less than or equal to 1 then the function should return false.

if (s.length() <= 1)

return false;

**Step 4:** Otherwise compare the first character of the string and last character of the string if both are same, return isPalindrome function with the parameter s as s.substring(1,s.length()-1) then go to step 3

if (s.charAt(0).equals(s.charAt(s.length()-1)

return isPalindrome (s.subString (1,s.length()-1))

**Step 5:** Repeat the above process until the string length reaches 0 or 1

**Step 6:** If both the above conditions fail, the function should return false

**Algorithm for anagrams:**

**Step 1:** Take an input and store it in the string str

**Step 2:** send str and a String variable ‘charac’ which stores an empty string as a parameter to anagrams function as follows:

anagrams (“ ”,str)

**Step 3:** verify if the string str length is equal to 0 then return charac

if (str.length() == 0)

print charac

**Step 4:**  Otherwise for every element in String str iterate the loop until the string length by sending the parameters to anagrams as follows:

for each index i in str.length():

anagrams (charac+str.charAt (i) , str.substring (0,i)+str.substring (i+1,str.length()))

**Step 5:** Then the control will go to step 3 and the above process is repeated for every character in a String

**Step 6:** All the anagrams of a string str will be printed after the completion of above process

**Algorithm for binary search:**

**Step 1:** Take the input of numbers and store them in an array ‘arr’ and take input for an element we need to search in an array and store it as ‘num’

**Step 2:** Check whether the array is sorted or not

**Step 3:** If the array is sorted continue from step 6.

**Step 4:** Otherwise the array should be sorted and continue from step 6 and define the variables as follows:

first = 0

last = array size - 1

**Step 5:** Compare num with the middle element of arr

Compute middle element as mid = (first + last) / 2

**Step 6:** if the mid is equal to num then print element is found otherwise continue from step 8

if(arr[mid] == num)

print (“Element found”)

**Step 7:** Divide the array into 2 parts

**Step 8:** If num is less than the middle element then store the left part as arr otherwise store the right part as arr and go to step 6

if(arr[num] < mid)

first = mid + 1

else

last = mid - 1

**Step 9:** Repeat the above process until the element is found or the size of an array becomes 0 or 1