❑Sweet 17!

Given a maximum of four digit to the base 17(10 -> A, 11 -> B, 12 -> C, 16 -> G) as input, output its decimal value. *(TCS)*

Sample I/O

I/P: 23GF

O/P: 10980

❑ Word is the key *(TCS)*

One programming language has the following keywords that cannot be used as identifiers:

break, case, continue, default, defer, else, for, func, goto, map, if, range, return, struct, type, var

Write a program to find if the given word is a keyword or not

Sample I/O

I/P: defer

O/P: defer is a keyword

❑Write a program to receive 3 English words inputs from user *(TCS)*

1. The first word should be changed like all vowels should be replaced by *
2. The second word should be changed like all consonants should be replaced by @
3. The third word should be changed like all char should be converted to upper case
4. Then concatenate the three words and print them

❑Sample I/O

- **Input:**

how

are

you

- Expected Output : h*wa@eYOU

1, 1, 2, 3, 4, 9, 8, 27, 16, 81, 32, 243,64, 729, 128, 2187 …. *(TCS)*

- This series is a mixture of 2 series – all the odd terms in this series form a geometric series and all the even terms form yet another geometric series Write a program to find the Nth term in the series
- The value N in a positive integer that should be read from STDIN
- The Nth term that is calculated by the program should be written to STDOUT
- Other than value of nth term, no other character / string or message should be written to STDOUT
- For example , if N=16, the 16th term in the series is 2187, so only value 2187 should be printed to STDOUT

❑ Inversion count in an array *(ACCENTURE, MS, AMAZON)*

Problem statement

Let ‘j’ and ‘k’ be two indices in an array A.

If j<k and A[j] > A[k], then the pair (j,k) is known as an “Inversion pair”

You are required to implement the following function:

int InversionCount(int A, int n);

The function accepts an array ‘A’ of ‘n’ unique integers as its argument. You are required to calculate the number of ‘Inversion pair’ in an array A, and return.

Note:

If ‘n’ <2, return 0

Example:

I/P

n: 5

A: 1 20 6 4 5

O/P

5

Explanation

The inversion pair in array A are (20,6), (20,4), (20,5), (6,4) and (6,5), the count of the inversions are 5, hence 5 is returned

❑ Superior array element *(ACCENTURE)*

Problem statement

In an array, a superior element is one which is greater than all elements to its right. The rightmost element will always be considered as a superior element.

You are given an function,

int FindNumberOfSuperiorElements(int arr, int n);

The function accepts an integer array ‘arr’ and its length ‘n’. Implement the function to find and return the number of superior elements in array ‘arr’.

Assumptions:

1. n>0
2. Array index starts from 0

Example

Input	***Output***	***Explanation***
Arr: 7 9 5 2 8 7	***3***	***9 is greater than all the elements to its right, 8 is greater than elements to its right and 7 is the rightmost element. Hence total 3 superior elements***

- ❑ Given a string S(input) consisting of ‘*’ and ‘#’. The length of the string is an integer variable. The task is to find the minimum number of ‘*’ and ‘#’ required to make it a valid string. The string is considered valid if the number of ‘*’ and ‘#’ are equal. The ‘*’ and ‘#’ can be at any position in the string *(TCS)*
- ❑Note: The output will be a positive or negative integer based on number of ‘*’ and ‘#’ in the input string
- ❑ (‘*’ > ‘#’) : Positive integer
- ❑ (‘#’ > ‘*’) : Negative integer
- ❑ (‘*’ = ‘#’) : 0

❑Sample I/O

1. I/P : ###*** → value of S

O/P: 0 → Number of ‘*’ and ‘#’ are equal

2. I/P : ###***# → value of S

O/P: -1 → Number of ‘#’ is more than ‘*’

3. I/P : #*** → value of S

O/P: 2 → Number of ‘*’ is more than ‘#’