SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND SCIENCES
CHENNAI-602105

Engineer to Excel

INSTITUTE OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
DEPARTMENT OF DATA SCIENCE

Course code/Name: CSA08/ Python Programming

DAY 2 PROGRAMS

1. You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Test Case:

1.Input: n = 2
Output: 2
2.Input: n = 3
Output: 3
3.Input: n = 4
Output: 5
4.Input: n = 1
Output: 1
5.Input: n = 5
Output: 8

2. Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years.
The rules for determining whether or not a year is a leap year follow:
Any year that is divisible by 400 is a leap year.
Of the remaining years, any year that is divisible by 100 is not a leap year.
Of the remaining years, any year that is divisible by 4 is a leap year.
All other years are not leap years.
Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

Sample Input:
Enter Date : 1947
Sample Output:
Given year is Non Leap Year
Leap Year: 1944

Test cases:

1. 19.47
2. 1936
3. 0
4. 2000
5. -1428

3. Maximum Number of Words Found in Sentences

A sentence is a list of words that are separated by a single space with no leading or trailing spaces.

You are given an array of strings sentences, where each sentences[i] represents a single sentence.

Return the maximum number of words that appear in a single sentence.

Test Cases:

1.Input: sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"]
Output: 6
Explanation:
- The first sentence, "alice and bob love leetcode", has 5 words in total.
- The second sentence, "i think so too", has 4 words in total.
- The third sentence, "this is great thanks very much", has 6 words in total.

Thus, the maximum number of words in a single sentence comes from the third sentence, which has 6 words.

2. Input: sentences = ["please wait", "continue to fight", "continue to win"]
Output: 3
Explanation: It is possible that multiple sentences contain the same number of words.
In this example, the second and third sentences (underlined) have the same number of words.
3. [“the heads”, “of”, “two”, “sorted linked lists”]
4. [“python” , “is”, “an object-oriented programming language”]
5. [“python”, “is”, “an interactive language”]

4. Merge Two Sorted Lists

You are given the heads of two sorted linked lists list1 and list2.

Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists.

Return the head of the merged linked list.

Constraints:

- The number of nodes in both lists is in the range [0, 50].
- Both list1 and list2 are sorted in non-decreasing order.

Test cases:

1.Input: list1 = [1,2,4], list2 = [1,3,4]
Output: [1,1,2,3,4,4]
2.Input: list1 = [], list2 = []
Output: []
3.Input: list1 = [], list2 = [0]
Output: [0]
4.list1=[],list2=[1,2,3,4,5]
5.list1=[0,1,9], list2=[3,4,5]

5. Basic Calculator II

Given a string s which represents an expression, evaluate this expression and return its value.

The integer division should truncate toward zero.

You may assume that the given expression is always valid. All intermediate results will be in the range of $[-2^{31}, 2^{31} - 1]$.

- s consists of integers and operators ('+', '-', '*', '/') separated by some number of spaces.
- s represents a valid expression.

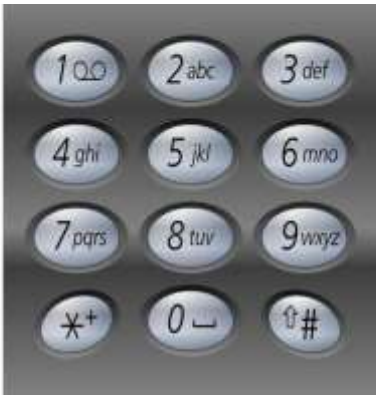
- All the integers in the expression are non-negative integers in the range [0, 2^{31} - 1].
- The answer is guaranteed to fit in a 32-bit integer.

Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval().

Test cases:

1.Input: s = "3+2*2"
Output: 7
2.Input: s = " 3/2 "
Output: 1
3.Input: s = " 3+5 / 2 "
Output: 5
4.s= “-1+5”
5.s= “2+3+5”

6.



Test Cases:

1.Input: digits = "23"
Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]
2.Input: digits = ""
Output: []
3.Input: digits = "2"
Output: ["a","b","c"]
4.Input: digits = "9"
Output: ["w","x","y",”z”]
5.Input: digits = "87"
Output: [“vs”,“vr”,”vq”, “vp”, “us”, “ur”, “uq”, “up”, “ts”, “tr”, “tq”, “tp”]

7. **Generate Parentheses**

Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

Test cases:

1.Input: n = 3
Output: ["((()))","(()())","(())()","()(())","()()()"]
2. Input: n = 1
Output: ["()"]
3.n=5
4.n=-1

5.n=0

8. Given an input string s and a pattern p, implement regular expression matching with support for '.' and '*' where:

- '.' Matches any single character.
- '*' Matches zero or more of the preceding element.

The matching should cover the **entire** input string (not partial).

Test case:

1.Input: s = "aa", p = "a"

Output: false

2.Input: s = "aa", p = "a*"

Output: true

3.Input: s = "ab", p = ".*"

Output: true

4.Input: s = " aaa", p = "aa"

Output: true

5.Input: s = "aab", p = "c*a*b"

Output: true

9. The year is divided into four seasons: spring, summer, fall and winter. While the exact dates that the seasons change vary a little bit from year to year because of the way that the calendar is constructed, we will use the following dates for this exercise:

Season First day

Summer March 20

Spring June 21

Fall September 22

Winter December 21

Create a program that reads a month and day from the user. The user will enter the name of the month as a string, followed by the day within the month as an integer. Then your program should display the season associated with the date that was entered. Note: Enter First three letter for month example: Jan for January, Feb for February and so on....and first letter of the month should be capital

Input:

Enter the month: march

Enter the date: 21

Output:

The season is currently summer

Test Cases:

1. July, 29
2. September, 5
3. December, 30
4. March, 12
5. June, 27

10. Python program to remove words that are common in two Strings
Given two strings S1 and S2, representing sentences, the task is to print both sentences after removing all words which are present in both sentences.

Test cases:

1.Input: S1 = “sky is blue in color”, S2 =”Raj likes sky blue color “
Output: is in
Raj likes
Explanation: The common words are [sky, blue, color]. Removing these words from the two sentences modifies the sentences to the specified output.
2.Input: S1 = “learn python“, S2 = “python is easy to learn“
3.S1= “raju likes apple”, S2=”apple is red in color”

4.S1= “ sita likes orange” S2=”orange is rich in anti-oxidents”

5. S1=”raj is travelling to Chennai in train” S2=”the rain will reach Chennai at 8 pm”