*#INTERVIEW QUESTIONS*

*=====================*

*#1.WAP to find the length of the string without using inbuilt funct.  
# s = 'hello python'  
# #print(len(s))  
# length = 0  
# for \_ in s: #when we are not changing ref var--> \_(throw away var)  
# length += 1  
# print(f'The length of the string {s}---> {length}')  
# print()  
#  
# #2.WAP to reverse a string without using inbuilt function  
# s = 'hello pyhton'  
# #print(s[::-1])  
# res = ''  
# for i in s:  
# res = i + res  
# print(res)  
# print()  
#  
# # s1 = reversed(s)  
# # print(list(s1))  
#  
# #3. WAP to replace one string with another.  
# #eg: hello world---> hello Universe.  
# s = 'Hello World'  
# u = 'Universe'  
# r = ''  
# for i in s.split():  
# if i == 'World':  
# r += u  
#  
# else:  
# r = i+ ' '  
# print(r)  
#  
# #another way.  
# if 'World' in s:  
# s1 = s.replace('World', 'Universe')  
# print(s1)  
# else:  
# print('Check question once')  
# print()  
#  
# #4.WAP to convert string into list and vice versa.  
# s = 'hello world'  
# # print(s.split())  
# # print(''.join(s.split()))  
# l = []  
# st = ''  
# for i in s:  
# if i != ' ':  
# st += i  
# else:  
# l += [st]  
# st = ''  
#  
# l += [st]  
# print(l)  
#  
# #Another way string to list  
# for i in s:  
# l += [i]  
# print(l)  
#  
# #converting from list to string  
# for i in l:  
# st += i  
#  
# print(st)  
#  
# #5.WAP to convert 'hello welcome to python' to comma separated string.  
# #o/p--->hello,welcome,to,strin  
# s = 'hello welcome to python'  
# for i in s.split():  
# print(i, end=',')  
# #  
# # print()  
# # print(','.join(s.split()))  
#  
# # s = '@#$%^&hello world@#$%^'  
# # print(s.strip('@#%$^&'))  
#  
# #6.WAP to print alternate characters from a given string  
# s = 'hello python'  
# #o/p-->'hlopto'  
# print(s[::2])  
#  
# for i in range(0, len(s), 2):  
# print(s[i], end= '')  
# print()  
#  
# #7.WAP to print ascii values of string  
# s = 'hello python'  
# d = {}  
# for i in s:  
# d[i] = ord(i)  
# print(d)  
# print()  
#  
# #8.WAF to convert upper case into lower case and vice versa.  
# def swap\_case(string, s1 =''):  
# for i in string:  
# if 'a' <= i <= 'z':  
# s1 += chr(ord(i)-32)  
# else:  
# s1 += chr(ord(i)+32)  
# return s1  
#  
# print(swap\_case('helloworld'))  
# print(swap\_case('HELLOWORLD'))  
# print()  
#  
# #9.WAP to swap 2 numbers without using third variable  
# a = 45  
# b = 76  
# #Swapping variables  
# # a = a+b  
# # b = a-b  
# # a = a-b  
# # a,b = b,a  
#  
# #with using third Variable  
# # a = 67  
# # b = 87  
# # c = 0  
# # a = b+c  
# # b = a+c  
# # c = a  
#  
# l = [2,4]  
# for i in range(len(l)-1):  
# l[i],l[i+1] = l[i+1],l[i]  
#  
# print(l)  
# print()  
# l = [2,4]  
# for i in range(len(l)-1):  
# l[i],l[i+1] = l[i+1],l[i]  
#  
# print(l)  
# print()  
#  
# #10.WAP to merge two list.  
# l1 = [1,3,5,7]  
# l2 = [2,4,6,8]  
# l3 = [3,6,9,12]  
# l4 = []  
#  
# for i in zip(l1,l2):  
# l4.append(i)  
# print(l4) #[(1, 2), (3, 4), (5, 6), (7, 8)]  
#  
# print([\*l1,\*l2]) #[1, 3, 5, 7, 2, 4, 6, 8]  
# print(sum([l1,l2], [])) #[1, 3, 5, 7, 2, 4, 6, 8]  
# print(sum([l1,l2,l3],[])) #[1, 3, 5, 7, 2, 4, 6, 8,3, 6, 9, 12]  
#  
# l2.extend(l1)  
# print(l2) #[2, 4, 6, 8, 1, 3, 5, 7]  
#  
# #14.WAP to check given string is Palindrome.  
# s = 'malayalam'  
# if s == s[::-1]:  
# print(f'the string {s} is Palindrome.')  
#  
# else:  
# print(f'the string {s} is not a Palindrome.')  
# print()  
#  
# #15.WAP to search for the character in a string and return the  
# #corresponding index.  
# s = 'hello world'  
# ch = 'w'  
# for index,element in enumerate(s):  
# if element == ch:  
# print(f'The char {ch} is present in index number {index}.')  
# print()  
#  
# #16. WAP to get below o/p  
# sentence = 'hello world welcome to python programming hi there'  
# #{'h':['hello', 'hai'], 'w':['world', 'welcome']......}  
# d = {}  
# for word in sentence.split():  
# if word[0] not in d:  
# d[word[0]] = [word]  
#  
# else:  
# d[word[0]] += [word]  
# print(d)  
#  
# #default dict  
# from collections import defaultdict  
# dd = defaultdict(list)  
#  
# for ch in sentence.split():  
# dd[ch[0]] += [ch]  
# print(dd)  
#  
# #17 WAP to replace all the characters with '-' if the characters occurs more than  
# #once in a string.  
# s = 'hellohai'  
# #o/p---> -e--o-ai  
# for i in s:  
# if s.count(i) > 1:  
# s = s.replace(i, '-')  
# print(s)  
# print()  
  
#18. WADF that returns only +ve values of subtraction  
# def outer(func):  
# def inner(\*args, \*\*kwargs):  
# res = func(\*args, \*\*kwargs)  
# #return abs(func(\*args, \*\*kwargs)  
# return abs(res)  
#  
# return inner  
#  
# @outer  
# def sub\_(a,b):  
# return a-b  
#  
# print(sub\_(6,12)) #o/p --> 6  
  
#20. WAF which takes list of strings and int , float, if it is of string print it  
#as it is else reverse it.  
# l = [34, 'hello', 'apple', 56.7, 4546, 67.8, 'google', 45]  
# def rev\_int\_float(lst, res = []):  
# for ch in lst:  
# if isinstance(ch, str):  
# res.append(ch)  
# elif isinstance(ch, int):  
# res += [int(str(ch)[::-1])]  
# elif isinstance(ch, float):  
# res += [float(str(ch)[::-1])]  
# return res  
#  
# print(rev\_int\_float(l))  
# print()  
  
#21. WA class called simple and it should have iteration capabilty  
# class Simple:  
# def \_\_init\_\_(self, a, b):  
# self.a = a  
# self.b = b  
# def add\_(self, dx, dy):  
# return self.a + dx, self.b+dy  
#  
# def sub\_(self, dx, dy):  
# return self.a-dx, self.b-dy  
#  
# s = Simple(4, 7)  
# print(s.add\_(3, 6))  
# print(s.sub\_(5, 4))*print()  
  
*# #22. Write a custom class which can access values of dict using d['a'] and d.a  
# class Access\_dict:  
#  
# def \_\_init\_\_(self, name, age):  
# self.name = name  
# self.age = age  
#  
# def \_\_getitem\_\_(self, key):  
# return self.\_\_dict\_\_[key]  
#  
# d = Access\_dict('Roshan', 25)  
# print(d['name'])  
# print(d.name)  
  
#23. WAP to get below o/p  
# s = 'Hi How are you'  
#o/p--> 'iH woH rea uoy  
# res = ''  
# for ch in s.split():  
# res += ch[::-1] + ' '  
# print(res)  
  
#24.WAP to get below o/p  
# s = 'Hi How are you'  
# #o/p --> 'uoy era woH iH'  
# res = ''  
# for ch in s:  
# res = ch+res  
# print(res)  
# print()  
#another way  
# res = ''  
# for ch in s.split():  
# res = ch[::-1] + ' ' + res  
# print(res)  
  
#25. WALE to add 2 numbers.(a,b)  
# add = lambda a,b : a + b  
# print(add(5,7))  
  
#26.What is o/p of the following  
# l = [1,2,3,4]  
# l1 = [2,4,6,8]  
# print([l,l1]) #---> list of list  
# print((l, l1)) #---> tuple of list  
#  
# #27.WAP to remove duplicates from a list without using inbuilt function.  
# l = [1,3,5,7,2,4,6,7,3,1]  
# dup = []  
# non\_dup = []  
# for i in l:  
# if i not in non\_dup:  
# non\_dup.append(i)  
#  
# else:  
# dup.append(i)  
# print(non\_dup)  
# print(dup)  
# print()  
  
#28. WAP to find longest word in sentence.  
# s = 'Life is full of surprises and miracles'  
# longest\_word = ''  
# max\_len = 0  
# for i in s.split():  
# if len(i) > max\_len:  
# max\_len = len(i)  
# longest\_word = i  
# print(longest\_word)  
# print()  
#another way  
# for ch in s.split():  
# if len(longest\_word) < len(ch):  
# longest\_word = ch  
# print(longest\_word)  
  
#29.WAP to reverse the values in the dictionary if value is of string type.  
# d = {'a': 'apple', 'one': 1, 'b': 'ball', 'three': 3, 'four':4, 'n': 45.7}  
# d1 = {}  
# for key,value in d.items():  
# if isinstance(value, str):  
# d1[key] = value[::-1]  
#  
# else:  
# d1[key] = value  
#  
# print(d1)  
# print()  
#  
# #30.WAP to get 1234  
# t = ('1', '2', '3', '4')  
# res = ''  
# for i in t:  
# res += i  
# print(res)  
# print()  
  
#31.How to get elements that are present in list b but not in list a.  
# a = ['hello', 'hai', 'world']  
# b = ['hello', 'hai', 'world', 'python']  
# # c = set(a)  
# # d = set(b)  
# # print(d.difference(c))  
# # for i in b:  
# # if i not in a:  
# # print(i)  
  
#32.A function takes variable number of positional arguments as input.  
#how to check if the arguments are more the 5.  
# def check\_(\*args, \*\*kwargs):  
# if len(args) > 5:  
# print(f'The arguments are {len(args)} which is more than 5')  
#  
# check\_(1,3,5,7,8,9)  
# print()  
  
#34.WAF to reverse any iterable without using reverse function.  
# # def reverse\_(iterable):  
# # s = '' #s = [], s= ()  
# # for i in iterable:  
# # s = i+s  
# # return s  
# #  
# # print(reverse\_('hello'))  
# print()  
#another way  
# def rev(\*args):  
# for i in args:  
# if isinstance(i, (str, list, tuple)):  
# return i[::-1]  
#  
# return args  
#  
# print(rev('hello'))  
# print(rev([1,3,5,7]))  
# print(rev((2,4,6,8)))  
# print(rev({1,2,3,4}))  
# print()  
#  
# #35.WAF to get the below o/p  
# #func('TRACXN', 0) ---> RCN  
# #func('TRACXN', 1) ---> TAX  
#  
# def func(string, i):  
# if i == 0:  
# print(string[1::2])  
#  
# else: #elif i == 1:  
# print(string[0::2])  
#  
# func('TRACXN', 0)  
# func('TRACXN', 1)  
# print()  
  
#36. WAP to sum all the numbers in below string.  
# s = 'Sony12India567pvt21ltd'  
# #1+2+5+6+7+2+1 = 24  
# res = 0  
# for i in s:  
# if i.isdigit(): #or if i.isdigit() == True  
# res += int(i)  
# print(res)  
  
#regular exp  
# from re import findall  
#  
# r = findall('[0-9]',s)  
# total = [int(i) for i in r]  
# print(sum(total))  
  
#37. Sum of numbers  
# s = 'Sony12India567pvt21ltd'  
# #12+21+567 = 600***from** re **import** findall  
*# res = findall('[0-9]+', s) -> 1256721  
# sum\_ = 0  
# for i in res:  
# sum\_ += int(i) #  
# print(sum\_)  
#print()  
  
#38.WAP to print all the numbers in below list.  
# l = ['hello', '123', 'hai', 'python', '345']  
# di = []  
# for i in l:  
# if i.isdigit():  
# di.append(i) #or di.append(int(i))  
# print(di)  
  
#regular exp:  
  
# jo = ''.join(l)  
# res = findall('[0-9]+', jo)  
# print(res)  
# print()  
  
#39.WAP to print number of occurance of a char in a given string  
# without using inbuilt func  
# s = 'hiihellowordhellowar'  
# d = {}  
# for i in s:  
# if i not in d:  
# d[i] = 1  
# else:  
# d[i] += 1  
# print(d)  
# #default dict  
# from collections import defaultdict  
# dd = defaultdict(int)  
# for i in s:  
# dd[i] +=1  
# print(dd)  
# print()  
  
#40.WAP to print repeated char and count the same  
# s = 'helloworld'  
# d = {}  
# for i in s:  
# if s.count(i) > 1:  
# d[i] = s.count(i)  
# print(d)  
# print()  
  
#41.WAP to get alternate char of a string in list.  
# s = 'helloworld'  
# l = []  
# for i in s[::2]:  
# l += [i]  
# print(l)  
# print(list(s[::2]))  
# print()  
  
#42.WAP to get squares of number using lambda  
# l = [1,3,5,7]  
# #o/p-->[1,9,25,49]  
# squ = lambda x : x \*\* 2  
# print(list(map(squ, l)))  
# print()  
  
#43.WAF that accepts two strings and returns True if strings are anagrams of each other.  
# def is\_anagram(string1, string2):  
# s1 = sorted(string1) #tea --> aet, eat--> aet  
# s2 = sorted(string2)  
# #or return s1 == s2  
#  
# if s1 == s2:  
# return True  
# else:  
# return False  
#  
#print(is\_anagram('tea', 'ate'))  
#print(is\_anagram('tiger', 'liger'))  
#print(is\_anagram('fare', 'fear'))  
  
#44.WAP to iterate through list and build a new list that contains  
# only even length elements  
# names = ['apple', 'google', 'yahoo', 'gmail', 'flipkart', 'amazon']  
# new\_list = []  
# for name in names:  
# if len(name) % 2 == 0:  
# new\_list.append(name)  
#  
# print(new\_list)  
#print()  
  
#45.WAP to create a dictionry of even length words.  
# names = ['apple', 'google', 'yahoo', 'gmail', 'flipkart', 'amazon']  
# d = {}  
# for name in names:  
# if len(name) % 2 == 0:  
# d[name] = len(name)  
# print(d)  
# print()  
  
#46.  
#l = [1,3,5,7]  
# #o/p-->[1,9,25,49]  
# squ = lambda x : x \*\* 2  
# print(list(map(squ, l)))  
  
# #49. WAP to print sum of internal and extrtenal list  
# l = [[1,2,3], [4,5,6], [7,8,9]]  
#internal = 6, 15, 24 #external --> 45  
#sum\_internal*res = []  
*# for i in l:  
# sum\_internal = 0  
# for j in i:  
# sum\_internal += j  
# res.append(sum\_internal)  
# print(res)  
  
# external = 0  
# for i in l:  
# for j in i:  
# external += j  
# print(external)  
# for i,j,k in l:  
# internal = 0  
# external = []  
# internal = i+j+k  
# external += [i+j+k]  
# print(internal)  
# print(external)  
  
#or  
# intrnl = [sum(i) for i in l]  
# print(intrnl)  
# extrnl = sum(intrnl)  
# print(extrnl)  
  
#50.WAP to reverse list as below  
# s = ['hello', 'hai', 'python']  
# l = []  
# for i in s:  
# l = [i]+l  
# print(l)  
  
# print(s[::-1])  
#print(list(reversed(s)))  
  
#51.WAP to update the update the tuple  
# t1 = (1,3,5,7)  
# t2 = (2,4,6,8)  
# print(t1+t2)  
# print((\*t1,\*t2))  
# print()  
#  
# #52.WAP to replace the value present in nested dict. i,e--> nose with net  
# d = {'a': 100, 'b':{'m':'man', 'n':'nose', 'o':'ox'}}  
# # d['b']['n'] = 'net'  
# #print(d)  
# # def replace\_(dict\_, old\_, new\_):  
# # for key,value in dict\_.items():  
# # if isinstance(value, dict):  
# # for k,v in value.items():  
# # if v == old\_:  
# # value[k] = new\_  
# # return dict\_  
# #  
# # print(replace\_(d,'nose','net'))  
# # print()  
#  
# #54.Grouping anagrams  
# names = ['listen', 'hello', 'eat', 'desserts', 'silent', 'peek', 'ate',  
# 'keep', 'tea', 'stressed']  
# d = {}  
# for name in names:  
# nme = ''.join(sorted(name))  
# if nme not in d:  
# d[nme] = [name]  
# else:  
# d[nme] += [name]  
#  
# print(d)  
# print()  
#  
# #55-58----> Theory Questions.  
#  
# #59. WALC to get a list of even numbers from 1,50  
#  
# print([i for i in range(2,51,2)])  
#  
# lst = [i for i in range(1,51) if i % 2 == 0]  
# print(lst)  
# print()  
  
#60.Find the longest non-repeated substring in the given.  
# s = 'This is a programming language and programming is fun'  
# s1 = ''  
# for i in s.split():  
# if len(s1) < len(i) and s.count(i)==1:  
# s1 = i  
# print(s1)  
  
#61.WAP to find the duplicate elements in the list without using  
#inbuilt func.  
# names = ['apple', 'google', 'gmail', 'apple', 'yahoo', 'google']  
# # l =[]  
# # for name in names:  
# # if names.count(name) > 1:  
# # if name not in l:  
# # l.append(name)  
# print(l)  
# di = [name for name in names if names.count(name)>1]  
# print(set(di))  
  
#62.WAP to count the number of occurances of each item in the list  
#without using inbuilt function.  
  
# names = ['apple', 'google', 'yahoo', 'google', 'apple', 'yahoo',  
# 'apple', 'yahoo', 'gamil']  
# word\_count = {}  
# for name in names:  
# if name not in word\_count:  
# word\_count[name] = 1  
#  
# else:  
# word\_count[name] += 1  
  
# print(word\_count)  
# #dict comprehension  
# print({name: names.count(name) for name in names})  
# print()  
  
#63. WAF to check the given number is prime or not.  
# def is\_prime(num):  
# if num > 1:  
# for i in range(2, num):  
# if num % i == 0:  
# print(f'the given number {num} is not a prime.')  
# break  
#  
# else:  
# print(f'The given number {num} is prime.')  
#  
# is\_prime(6)  
# is\_prime(7)  
#print()  
  
#64.HOw to create a tuple of numbers from 0-10 using range func  
# l = []  
# for num in range(10):  
# l.append(num)  
#  
# print(tuple(l))  
#print()  
  
#65.WAP to print largest number in the list without using inbuilt fun  
# numbers = [10,30, 50, 40, 60, 20]  
# s = sorted(numbers)  
# print(s[-1])  
  
# n = 0  
# for num in numbers:  
# if num > n:  
# n = num  
# print(n)  
  
# for i in range(len(numbers)-1):  
# if numbers[i] > numbers[i+1]:  
# numbers[i], numbers[i+1] = numbers[i+1], numbers[i]  
# print(numbers[-1])  
  
# for i in range(len(numbers)):  
# for j in range(len(numbers)-1):  
# if numbers[j] > numbers[j+1]:  
# numbers[j],numbers[j + 1] = numbers[j+1], numbers[j]  
# print(numbers[-1])  
#print()  
  
#66.Write a method that returns last digit of an integer.  
# def get\_lastdigit(num):  
# res = str(num)  
# return int(res[-1])  
#  
# print(get\_lastdigit(5467))  
  
#67.WAP to find the most common words in the list.  
# words=['look','into','my','eyes','look','into','my','eyes','the','eyes','the','eyes','the','eyes','not','around',  
# 'the','eyes','dont','look','around','the','eyes','look','into','my','eyes',"youre",'under']  
#  
# d = {word:words.count(word) for word in words}  
# #print(d)  
# sort = sorted(d.items(), key = lambda item: item[1])  
# print(sort[-1])  
  
#68.make a func named tail that takes a seq(string, list, tuple)  
#and a number n and returns last n elements from the given seq as a list.  
  
# def tail(args, n):  
# return list(args[-n:])  
#  
# print(tail('helloworld',2))  
# print()  
  
#69. WAF named is\_perfect that accepts number and returns True  
#if its a perfect square else False.  
  
# import math  
# def is\_perfectsqu(num):  
# res = num//2  
# for i in range(res):  
# if i \* i == num:  
# return True  
# #return f'{num}--> is a perfect square'  
# return False  
# # return f'{num}--> is not a perfect square'  
#  
# print(is\_perfectsqu(11))  
# print(is\_perfectsqu(169))  
# print(is\_perfectsqu(256))  
  
#OR  
# import math  
# def is\_perfectsq(num):  
# res = math.sqrt(num)  
# if res == int(res):  
# return True  
# else:  
# return False  
#  
# print(is\_perfectsq(25))  
  
# perfect num  
# def is\_perfectnum(num):  
# res = 0  
# for i in range(1,num):  
# if num % i == 0:  
# res += i  
# print(num==res)  
  
#70. WAP to get all the duplicates items and numbers of times  
#it is repeated in list.  
# names = ['apple', 'google', 'yahoo', 'google', 'apple', 'yahoo',  
# 'apple', 'yahoo', 'gamil']  
#  
# count\_pair = {name:names.count(name) for name in names if names.count(name) > 1}  
# print(count\_pair)  
  
#or  
# res = {}  
# for name,count\_ in count\_pair.items():  
# if count\_ > 1:  
# res[name] = count\_  
# print(res)  
#print()  
  
#73. WAP to all numeric values in a list  
# l = ['apple', 123,45.6, 'google', [1,2,3], '4+6', 3+3j]  
# res = []  
# for i in l:  
# if isinstance(i, (int, float, complex)):  
# res.append(i)  
# print(res)  
#  
# print([i for i in l if isinstance(i, (int, float, complex))])  
  
#74. Trainale pattern.  
# \*  
# \* \*  
# \* \* \*  
# \* \* \* \*  
# # \* \* \* \* \*  
# n = int(input('enter a number:'))  
# for i in range(n):  
# for j in range(i+1):  
# print('\*', end = ' ')  
# print()  
#reversed triangle  
# n = int(input('enter a number:'))  
# for i in range(n):  
# for j in range(n-i):  
# print(' ', end = ' ')  
# for j in range(i+1):  
# print('\*',end = ' ')  
# print()  
  
#76. WAP to to map a product to a company and build a dictionary with company  
#and list of products pair.***from** collections **import** defaultdict  
all\_products = [**'iphone'**, **'mac'**, **'gmail'**, **'google maps'**, **'iwatch'**, **'windows'**,  
 **'ios'**,**'google drive'**, **'one drive'**]  
apple\_products = []  
google\_products = []  
windows\_products = []  
*# apple\_products = ['iphone', 'mac', 'iwatch', 'ios']  
# google\_products = ['gmail', 'google maps', 'google drive']  
# windows\_products = ['windows', 'one drive']  
  
# exp o/p= {'apple\_products':['iphone', 'mac', 'iwatch', 'ios'],  
# 'google\_products':['gmail', 'google maps', 'google drive'],  
# 'windows\_products':['windows', 'one drive']}  
  
# products = defaultdict(list)  
#  
# for product in all\_products:  
# if product in apple\_products:  
# products['apple\_products'] += [product]  
#  
# elif product in google\_products:  
# products['google\_products'] += [product]  
#  
# elif product in windows\_products:  
# products['windows\_products'] += [product]  
#  
# print(products)  
  
# apple = []  
# google = []  
# windows = []  
# d = defaultdict(list)  
# for item in all\_products:  
# if item.startswith('i') or item.startswith('m'):  
# d['apple'] += [item]  
#  
# elif item.startswith('g'):  
# d['google'] += [item]  
#  
# else:  
# d['windows'] += [item]  
# print(d)  
  
#hard-coding  
# for product in all\_products:  
# if product == 'iphone' and product == 'mac' and product == 'iwatch' and product == 'ios':  
# apple\_products.append(product)  
#  
# elif product == 'gmail' and product == 'google maps' and product == 'google drive':  
# google\_products.append(product)  
#  
# elif product == 'windows' and product == 'one drive':  
# windows\_products.append(product)  
#  
# products = defaultdict(list)  
# for product in all\_products:  
# if product in apple\_products:  
# products['apple\_products'] += [product]  
#  
# elif product in google\_products:  
# products['google\_products'] += [product]  
#  
# elif product in windows\_products:  
# products['windows\_products'] += [product]  
# print(products)  
  
#77. WAP to rotate items of the list  
# names = ['apple', 'google', 'yahoo', 'gamil', 'facebook', 'flipkart', 'amazon']  
  
# def rotate(l, n):  
# return l[n:] + l[:n]  
#  
# print(rotate(names, -3))  
# print()  
# l = [1,2,3,4,5]  
# shift = 2  
# for i in range(0,shift):  
# temp = l[0]  
# for j in range(0,len(l)-1):  
# l[j] = l[j+1]  
# l[len(l)-1] = temp  
#  
# for i in range(0,len(l)):  
# print(l[i])  
# print()  
#  
# #78. WAP to rotate characters in a string.  
# s = 'darshan'  
# def rotate\_str(string, n):  
# return string[n:] + string[:n]  
#  
# print(rotate(s, 2))  
# print()  
  
#79. WAP to to count the numbers of white spaces in a given string  
# from re import findall  
# s = 'hai hello how are you'  
# space = findall('\s', s)  
# print(len(space))  
  
#or  
# count = 0  
# for i in s:  
# if i == ' ':  
# count += 1  
# print(count)  
  
#80. WAP to print only non-repeated characters in a string.  
# s = 'hai hello how are you'  
# res = ''  
# for i in s:  
# if s.count(i) == 1: #s.count(i) < 2  
# res += i  
# print(res)  
  
#81. theory  
  
#82. WAP to print all the consonants in the string.  
# s = 'hello world'  
# consonants = ''  
# for i in s:  
# if i not in 'aeiouAEIOU':  
# consonants += i  
# print(consonants)  
  
#84. WAP to check if the year is leap year or not.  
# year = eval(input('enter the year:'))  
# if year % 4 == 0:  
# print('its a leap year')  
#  
# else:  
# print('its not a leap year')  
  
# if year % 4 == 0 and year % 100 == 0:  
# print('It is a leap year')  
#  
# elif year % 4 == 0 and year % 100 != 0:  
# print('It is also leap year')  
#  
# else:  
# print('its not a leap year')  
  
#85.linear search : search one by one in a sequence  
  
#86. Differnece b/w x-range and range  
#both are same x-range is used in python 2 and range is used in python 3  
  
#87. WAP to count number of capital letters in a string.  
# s = 'Hi How are You Welcome to Python And its Fun'  
# c = 0  
# for i in s:  
# if i.isupper():  
# c += 1  
# print(c)  
#  
# #regular exp  
# from re import findall  
# upper\_case = findall('[A-Z]', s)  
# print(len(upper\_case))  
  
#88. WAPt to get below o/p  
# \*  
# \* \*  
# \* \* \*  
# \* \* \* \*  
# n = 4  
# for i in range(n):  
# for j in range(i+1):  
# print('\*', end = ' ')  
# print()  
  
#89. WAP to get below o/p*l = [1,2,3,4,5,6,7,8,9]  
*#exp o/p is below.  
# [1,2]  
# [3,4]  
# [5,6]  
# [7,8]  
# [9]  
# res = []  
# for i,j in enumerate(l):  
# if i % 2 == 0:  
# res.append(j)  
#  
# else:  
# res.append(j)  
# print(res)  
# res = []  
# if len(l) % 2 == 1:  
# print(res)  
  
#90. WAP to check if the elements in the second list is series  
#of continuation of the items in the first list.  
#  
  
#91. Difference between append(), extend() methods in list.  
#in append() we can pass both individual and collection datatypes  
#it will add the element at the last  
#extend() : We can pass only iterables, it will extend the existing list.  
  
#92. WAP to find the first repeating character in strings.  
# s = 'hi there how are you'  
# res = []  
# for i in s:  
# if i not in res:  
# res.append(i)  
#  
# else:  
# print(i)  
# break  
# print(res)  
# print()  
#  
# #93.WAP to find the the index of the nth occurance of a substring in a string  
# s = 'hi hello world how are you hello how are you'  
#  
# from re import finditer  
# res = finditer('you', s)  
# out\_put = list(res)  
# print(out\_put[-1])  
  
#94.WAP to print prime numbers from 1-50  
# l = []  
# for num in range(1,50):  
# for i in range(2,num):  
# if num % i == 0:  
# break  
#  
# else:  
# l.append(num)  
# print(l)  
# print()  
  
#95. WAP to sort the list which is mix of both odd and even numbers, the sorted  
# list should have odd numbers first and then even numbers in sorted order.  
  
# l = [3,4,1,7,2,12,8,6,9,11]  
# #odd = [3,1,7,9,11]--> [1,3,7,9,11]  
# #even = [4,2,12,8,6]--> [2,4,6,8,12]  
#  
# odd = []  
# even = []  
# for i in l:  
# if i % 2 != 0:  
# odd.append(i)  
#  
# else:  
# even.append(i)  
#  
# res = sorted(odd) + sorted(even)  
# print(res)  
  
#96. WAP to sort the list which is mix of both odd and even numbers, the sorted  
# list should have odd numbers be in ascending order and even numbers in  
# descending order.  
  
#l = [3,4,1,7,2,12,8,6,9,11]  
# #odd = [3,1,7,9,11]--> [1,3,7,9,11]  
# #even = [4,2,12,8,6]--> [12,8,6,4,2]  
# odd = []  
# even = []  
# for i in l:  
# if i % 2 != 0:  
# odd.append(i)  
#  
# else:  
# even.append(i)  
#  
# res = sorted(odd) + sorted(even,reverse= True)  
# print(res)  
  
#97. WAP to count the numbers of occurances of non-special characters in a given string  
# s = 'hello@world!welcome!!!python hi how are you & where are you'  
# c = 0  
# for i in s:  
# if i.isalpha():  
# c += 1  
# print(c)  
#  
# from re import findall  
# res = findall('[A-Za-z0-9]',s)  
# print(len(res))  
  
#98.Grouping flowers and animals separately  
# items = ['lotus-flower', 'lilly-flower', 'cat-animal', 'dog-animal',  
# 'sunflower-flower']  
# d = {}  
# for i in items:  
# temp = i.split('-') #-->['lotus', 'flower']  
# if temp[-1] not in d:  
# d[temp[-1]] = [temp[0]]  
#  
# else:  
# d[temp[-1]] += [temp[0]]  
# print(d)  
  
# #99. Grouping files with same extension  
# files = ['apple.txt', 'yahoo.pdf', 'google.pdf', 'gmail.txt', 'amazon.pdf',  
# 'flipkart.txt']  
#  
# d\_files = {}  
# for i in files:  
# file = i.split('.')  
# if file[-1] not in d\_files:  
# d\_files[file[-1]] = [file[0]]  
#  
# else:  
# d\_files[file[-1]] += [file[0]]  
# print(d\_files)  
  
#100.Filter only characters except digits.  
# s = 'ghello12world34welcome! 123'  
# res = ''  
# for i in s:  
# if i.isdigit() != True:  
# res += i  
# print(res)  
  
#101.Count the number of words in a sentence ignore special character.  
# sentence = 'Hi there! how are you:) How are you doing toady!'  
# from re import findall  
# res = findall('[A-Za-z0-9]+', sentence)  
# print(len(res))  
  
#102. Grouping even and odd numbers.  
# numbers = [1,2,3,4,5,6,7,8,9,10]  
# odd\_even = {}  
# for i in numbers:  
# if i % 2 == 0:  
# if 'even' not in odd\_even:  
# odd\_even['even'] = [i]  
# else:  
# odd\_even['even'] += [i]  
#  
#  
# else:  
# if 'odd' not in odd\_even:  
# odd\_even['odd'] = [i]  
#  
# else:  
# odd\_even['odd'] += [i]  
# print(odd\_even)  
  
#103.find all the max numbers from below list  
# numbers = [1,2,3,0,4,3,2,4,2,2,0,4]  
# sort = sorted(numbers)  
# max\_ = [num for num in sort if num >= sort[-1]]  
# print(max\_)  
#or  
# max\_num = []  
# for num in sort:  
# if num >= sort[-1]:  
# max\_num.append(num)  
# print(max\_num)  
  
#104.Find all the max length words from below sentence  
# s = 'hello world hi apple you yahoo to you'  
# s1 = s.split()  
  
# d = {i:len(i) for i in s1}  
# sort = sorted(d.items(), key = lambda item: item[-1])  
#  
# max\_words = []  
# for i in sort:  
# if i[-1] >= sort[-1][-1]:  
# max\_words.append(i)  
# print(max\_words)  
  
#105. find the range from the following string.  
# s = '0-0,4-8,20-20,43-45'  
# s1 = s.split(',')  
# res = []  
# for i in s1:  
# var = i.split('-')  
# for j in range(int(var[0]), int(var[1])+1):  
# res.append(j)  
# print(res)  
  
#106. Can we overide static method in python.  
#solution : Yes.  
  
#107. WAF to which returns the sum of length of the iterables.  
#total\_length = ([1, 2, 3], (4,5), ['apple', 'google', 'yahoo', 'gmail'],  
# (1,2,3), {'a':1, 'b': 2})  
#sample\_sum --> (3+2+4+3+2)= 14  
# def sum\_length(\*args):  
# sum\_ = 0  
# for i in args:  
# for j in i:  
# sum\_ += len(j)  
# return sum\_  
#  
# print(sum\_length(([1, 2, 3], (4,5), ['apple', 'google', 'yahoo', 'gmail'],  
# (1,2,3), {'a':1, 'b': 2})))  
# print()  
  
#or  
# def total\_len(args):  
# length = 0  
# for i in args:  
# length += len(i)  
#  
# return length  
#  
# print(total\_len(([1, 2, 3], (4,5), ['apple', 'google', 'yahoo', 'gmail'],  
# (1,2,3), {'a':1, 'b': 2})))  
  
  
#108. Replaces whitespaces with newline char in the below string.  
#s = 'hello world welcome to python'  
#hello  
#world  
#welcome  
#to  
#python  
  
# for i in s:  
# if i == ' ':  
# res = s.replace(i, '\n')  
# print(res)  
  
# result = '\n'.join(s.split())  
# print(result)  
  
# res1 = s.replace(' ', '\n')  
# print(res1)  
# print()  
  
#109. Replace all vowels with '\*'  
# s = 'hello world welcome to python'  
# #h\*ll\* w\*rld w\*lc\*m\* t\* pyth\*n  
# for i in s:  
# if i in 'AEIOUaeiou':  
# res = s.replace(i, '\*')  
# print(res)  
#print()  
  
#or  
# res = ''  
# for i in s:  
# if i in 'AEIOUaeiou':  
# res += '\*'  
#  
# else:  
# res += i  
# print(res)  
  
#or  
# from re import sub  
# res = sub('["AEIOUaeiou"]', '\*', s)  
# print(res)  
  
#110.Replace all ocuurance of 'java' with 'Python' in a file.  
#Assume file is sample\_file  
# import os  
# with open(r'C:\Users\Admin\_name\Desktop\foldername\sample\_file.txt', 'r') as file:  
# for i in file:  
# if 'Java' in file:  
# file.write('Python')  
  
#111.Maximum sum of 3 numbers and Minimum sum of 3 numbers.  
#numbers = [18, 15, 20, 25, 30, 35, 40, 15, 5]  
#max\_sum = 30+35+40 = 105  
#min\_sum = 5+15+15 = 35  
  
# sort = sorted(numbers)  
# add\_min = sum(sort[:3])  
# add\_max = sum(sort[-3:])  
# print(sort)  
# print(add\_min, add\_max)  
  
# numbers = [10, 15, 20, 25, 30, 35, 40, 15, 15]  
# sort = sorted(numbers)  
# add\_max = sum(sort[0:3:1])  
# add\_min= sum(sort[-3:len(numbers):1])  
# print(sort)  
# print(add\_max,add\_min)  
# print(numbers[0:3:1])  
# print()  
  
#112. WAP to get below o/p.  
# s = 'python@#$%pool'  
#o/p-->['python', 'pool']  
  
# import re  
# print(re.findall(r'p\w+',s))  
#or  
# from re import findall  
# res = findall('[a-z]+', s)  
# print(res)  
#print()  
  
#113.WAP to print all numbers which are ending with 5  
# num = ['1', '12', '13', '12345', '125', '905', '55', '5', '95655', '55555']  
# #o/p : ['12345', '125', '905', '55', '5', '95655', '55555']  
# import re  
# print(list(filter(lambda s: re.findall(r'.\*5$', s),num)))  
  
#or  
# l = []  
# for i in num:  
# if i.endswith('5'):  
# l.append(int(i))  
# print(l)  
  
# #114.WAP to to get the indicies of each item in the list  
# names = ['apple', 'google', 'yahoo', 'apple', 'yahoo', 'google', 'gmail',  
# 'apple', 'gmail', 'yahoo']  
# #apple --> [0, 3, 7]  
# #google --> [1, 5]  
# #yahoo --> [2, 4, 9]  
# #gmail --> [6, 8]  
# d = {}  
# for index, element in enumerate(names):  
# if element not in d:  
# d[element] = [index]  
#  
# else:  
# d[element] += [index]  
# print(d)  
#print()  
  
#115.WAP to print 'Bangalore' for 10 times without using 'for' loop  
# print('Banglore\n' \* 10)  
  
#or  
# s = 'Banglore'  
# i = 1  
# while i <= 10:  
# print(s)  
# i += 1  
#print()  
  
#116.WAP to print all the words which starts with letter 'h' in the given string.*s = **'hello world hi hello universe how are you happy birthday'***#o/p--> hello, hi, hello, how, happy  
# res = []  
# for i in s.split():  
# if i.startswith('h'):  
# res.append(i)  
# print(' '.join(res))  
  
#print(res) #o/p---> list of strings  
  
#or  
# from re import findall  
# result = findall(r'\bh[a-z]+\b', s)  
# print(' '.join(result))  
  
#117. WAP to sum of even numbers in the given string.  
# s = 'hello 123 world 567 wlcome to 9724 python'  
# #2+6+2+4--> 14  
# sum\_even = 0  
# for i in s:  
# if i.isdigit() and int(i) % 2 == 0:  
# sum\_even += int(i)  
# print(sum\_even)  
#  
# #or  
# from re import findall  
# res = findall('[\d]', s)  
# ev\_num = 0  
# for i in res:  
# if int(i) % 2 == 0:  
# ev\_num += int(i)  
# print(ev\_num)  
  
#118.WAP to add each number in word1 to number in word2  
# word1 = 'hello 1 2 3 4 5'  
# word2 = 'world 5 6 7 8 9'  
# a = word1.split()  
# b = word2.split()  
# l = []  
# for i, j in zip(a,b):  
# if i.isdigit() and j.isdigit():  
# l.append(int(i)+int(j))  
# print(l)  
#print()  
  
# #119.WAP to filter out even and odd numbers in the given string.  
# s = 'hello 123 world 456 welcome to python498675634'  
# even = ''  
# odd = ''  
# for i in s:  
# if i.isdigit() and int(i) % 2 == 0:  
# even += i  
#  
# else:  
# if i.isdigit():  
# odd += i  
#  
# print(even)  
# print(odd)  
#print()  
  
#120.WAP to print all the numbers starting with 8  
# numbers = ['857', '987', '8', '128', '88888', '547', '7674', '89', '589',  
# '38888', '2889']  
#  
# import re  
# print(list(filter(lambda s : re.findall(r'^8.\*',s),numbers)))  
  
#question? one more Regular expression..  
  
# #121. WAP to remove duplicates from the list without using set or empty list  
# l = [1, 2, 3, 4, 1, 2, 3, 4, 3, 4, 4]  
# #1, 2, 3, 4  
# res = []  
# for i in l:  
# if i not in res:  
# res += [i]  
# print(res)  
  
#122.Print all the missing numbers from 1-10 in the below list  
# l = [1, 2, 3, 4, 6, 7, 10]  
# res = []  
# for i in range(1, 11):  
# if i not in l:  
# res += [i]  
# print(res)  
  
#123. WAP to get below o/p  
# l1 = [1, 2, 3]  
# l2 = ['a', 'b', 'c']  
# print([(str(i)+j) for i in l1 for j in l2])  
  
#124. Write a python program to get the below output  
# a = "10.20.30.40"  
# res = a.split(".")[::-1]  
# print(".".join(res))  
  
#  
# a = [3, 5, -4, 8, 11, 1, -1, 6]  
# for i in a:  
# for j in a:  
# if i - j == 10 or i +j == 10 and i !=j :  
# print(i,j)  
  
#125.What is the o/p of the below fubction call  
# class Demo:  
# def greet(self):  
# print('hello world')  
#  
# def greet(self):  
# print('hello universe')  
#  
# d = Demo()  
# d.greet() #o/p ----> hello universe  
  
#126.In the below, find all the number pairs which results in 10 either when  
#we added or subtracted.*l = [3, 5, 4, 8, 11, 1, -1, 6]  
*# for i in l:  
# for j in l:  
# if i-j == 10 or i+j == 10 and i != j:  
# print(i,j)  
  
#or  
# res = []  
# for i in l:  
# for j in l:  
# if i-j == 10 or j-i == 10 or i+j == 10:  
# res.append((i,j))  
# print(res)  
#print()  
  
#127. WADF to prefix +91 to original phone number  
# def prefix(func):  
# def wrapper(\*args, \*\*kwargs):  
# res = func(\*args, \*\*kwargs)  
# return f'+91{res}'  
#  
# return wrapper  
#  
# @prefix  
# def mob\_num(n):  
# return n  
#  
# print(mob\_num(9087654321))  
  
#or---> for list of numbers  
# def addcode(func):  
# def inner(args):  
# for i in args:  
# print(f"+91{i}")  
# func(args)  
# return inner  
# @addcode  
# def phoneno(no):  
# return no  
# phoneno([9563478902,9876502345,7890567845])  
#print()  
  
#128. WAP to get below o/p.  
# d = {'a':1, 'b': 2, 'c':3, 'd': 4, 'e': 5}  
# #o/p--> ['b', 'd']  
# res = list(d.keys())  
# print(res[1::2])  
#  
# #or  
# l = []  
# for i in d:  
# if i == 'b' or i == 'd':  
# l.append(i)  
# print(l)  
#  
# #or  
# print([i for i in d if i == 'b' or i == 'd'])  
  
#129. Can we hae multiple \_\_init\_\_methods in a class.  
  
#solu : Yes we can have but it will override latest one will be priority  
#we should have \_\_init\_\_ methods calling multiple super classes.  
  
#130. Why python is object oriented?  
#solu : It is one of its feature and Any objects which surrounds by its functions  
#is called as Object oriented, Since python supports all OOPS concepts hence  
#it is called object oriented.  
  
  
#131.What are .pyc files.  
#solu : It is python compiled and it will in byte format(machine code)  
  
#132. Reverse a list without using any built-in fucntions and slicing.  
# l = [1, 2, 3, 4]  
# res = []  
# for i in l:  
# res = [i] + res  
# print(res)  
#print()  
  
#133. Repeated with Q.no- 124  
  
#134. What is the differenec b/w while loop and for loop  
#solu : When we know the range we go for for loop  
#when we dont know the range we go for while loop  
  
#135.What are magic methods.  
#solu : protocols which followed during constrcution any conceots such as oops  
#function object is called magic methods.  
#these are also called as special methods, dunder metnods, double underscore methods  
  
#136.What is pylint?  
#solu : It is a static code analysis tool to identify errors in Python code  
# and helps programmers enforce good coding style.  
# This tool enables them debugging complex code with less manual work.  
# It is one of the tools which gets used for test-driven development (TDD)  
#print()  
  
#137.What is the o/p of the below program  
# print([1, 2, 3, 4] \* 2)  
# #obtained o/p --> [1, 2, 3, 4, 1, 2, 3, 4]  
  
#138.What is the differnece b/w is and == operators.  
#is opeator : It returns True if objects are pointed to the same memory allocation.  
#It belongs to Identity operator  
# == operator : It returns True if operand1 exactly equals to operand2.  
#It belongs to comparision operator.  
  
#139.What is 'self' in class?  
#solu : self holds the address of instance which invokes the methods.  
  
#140.What is assert statement? What is the diff b/w assert & if/else statement?  
#solu : If the condition is True it will print TSB(true statement block) if the condition is False it returns user message.  
  
# def Divexp(a,b):  
# assert a > 0, 'Error'  
# if b == 0:  
# raise ZeroDivisionError  
#  
# else:  
# c = a/b  
# return c  
#  
# a = eval(input('enter a:'))  
# b = eval(input('enter b:'))  
# # print(Divexp(a,b))  
  
#OR.  
# batch = [ 40, 26, 39, 30, 25, 21]  
# cut = int(input('enter c:'))  
# for i in batch:  
# assert i > cut, "Batch is Rejected"  
# print (str(i) + " is O.K" )  
#print()  
  
#141. Diff b/w module, package, library  
#module --> python file with .py extensions  
#package --> python file folder conatin : \_\_init\_\_.py is called package  
#library --> one or more package and python python file.  
  
#142.WAP to get below o/p using while loop***'''  
1  
12  
123  
1234  
'''***# n = int(input('enter a number:'))  
# for i in range(1,n+1):  
# for j in range(1, i+1):  
# print(j, end = ' ')  
# print()  
#  
# 1  
# 1 2  
# 1 2 3  
# 1 2 3 4  
# 1 2 3 4 5  
  
#while loop  
# i = 1  
# while i <= 5:  
# j = 1  
# while j <= i:  
# print(j, end = ' ')  
# j += 1  
# print()  
# i += 1  
# # print()  
#print()  
  
#143. WAP to get below o/p.  
# items = ['$123.45', '$434.23', '$567.89']  
# #o/p-->[123.45, 434/23, 567.89]  
# res = []  
# for i in items:  
# res.append(float(i.strip('$')))  
# print(res)  
  
#or  
#from re import findall  
# result = ''.join(items)  
# l = []  
# r = findall('[\d\.\d]+', result)  
# for i in r:  
# l.append(float(i))  
# print(l)  
  
#144. Geneartor function for fibonicci series.  
# def fib(n):  
# a,b = 0,1  
# for i in range(n):  
# c = a+b  
# yield a  
# a = b  
# b = c  
#  
# res = list(fib(10))  
# print(res)  
  
#145.WAP to print common characters present in all the items of the below list  
# items = ['glory', 'glass', 'signt', 'tight']  
# res = set(items[0])  
# for word in items[1:]:  
# res = res.intersection(set(word))  
# for char in res:  
# print(char)  
  
#146.***def** modify(list):  
 res = []  
 **for** i **in** list:  
 **if** i %3 == 0:  
 i = 33  
 res += [i]  
 **else**:  
 res+= [i]  
  
 **return** res  
print(modify([2,3,7,8,12,8,50,63,100]))  
  
*#147.  
#1 2 3 \*  
#1 2 \* 4  
#1 \* 3 4  
#\* 2 3 4*n = int(input(**"Enter a number: "**))  
**for** i **in** range(1, n+1):  
 **for** j **in** range(1, n+1):  
 **if** i+j == n+1:  
 print(**"\*"**, end = **" "**)  
 **else**:  
 print(j, end = **" "**)  
 print()