

Unit 1

Write a python program to check for validity of the password using if
statement.

```
import re
password = input('password: ') # eg: wewewetRRRRTY45@

def e() :
    print("Invalid password, please choose a different password")
    quit()

if (len(password)<8):
    e()
elif not re.search("[a-z]", password):
    e()
elif not re.search("[A-Z]", password):
    e()
elif not re.search("[0-9]", password):
    e()
else:
    print("Good password")
#-----
```

```
#-----
```

Write a Python program in which a student enters the number of college credits
earned. If the number of credits is greater than 90, 'Senior Status' is displayed;
if greater than 60, 'Junior Status' is displayed; if greater than 30, 'Sophomore
Status' is displayed; else, 'Freshman Status' is displayed.

```
credits = int(input('Enter your credits: '))

if credits > 90 :
    print('Senior status')
elif credits > 60 :
    print('Junior status')
elif credits > 30 :
    print('Sophomore status')
elif credits > 0 :
    print('Freshman Status')
else :
    print('Enter valid credits')
```

```
#-----
```

```
#-----
```

```
# Design and implement a program to count the number of individual
# characters in a string.
# Sample string:"yahoo.com"
# Result:{'o':3,'y':1,'.':1,'a':1,'h':1,'m':1,'c':1}
```

```
def ch_count(ip_string) :
    count = {}
    for c in ip_string :
        try :
            count[c] += 1
        except :
            count[c] = 1
    return count
```

```
print(ch_count('abcbabcabc'))
```

```
# o/p : {'a': 3, 'b': 3, 'c': 3}
```

```
#-----
```

```
#-----
```

```
# Write a program that sums a series of integers entered by the user, excluding
# all numbers that are greater than 100.
```

```
num_arr = map(int,input('Enter number array seperated by spaces: ').split())
```

```
def sum_all_99(num_list) :
    return sum(filter(lambda n: n < 101, num_list))
```

```
print(sum_all_99(num_arr))
```

```
#-----
```

```
#-----
```

```
# Write a Python program which takes two digits m (row) and n (column) as
# input and generates a two-dimensional array. The element value in the i-th
# row and j-th column of the array should be i*j.
# Test Data : Rows = 3, Columns = 4
# Expected Result : [[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]
```

```
m, n = map(int, input('Enter m and n values: ').split())
```

```
l = []
for i in range(m) :
    ll = []
    for j in range(n) :
        ll.append(i * j)
    l.append(ll)
```

```
print(l)
```

```
#-----
```

```
#-----
```

Write a Python program to find numbers between 100 and 400 (both included) where each digit of a number is an odd number. The numbers obtained should be printed in a comma-separated sequence

```
def all_odd() :  
    for i in range(99, 401, 2) :  
        if i//100 % 2 :  
            if i//10 % 2 :  
                print(i, end=', ')
```

```
all_odd()  
#-----
```

```
#-----
```

Write a Python program that accepts a word from the user and reverse it.

```
ip_str = input('Enter string to reverse: ')  
rev_str = ip_str[::-1]
```

```
print(rev_str)  
#-----
```

```
#-----
```

Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included).

```
for i in range(1500, 2701) :  
    if i % 7 == 0:  
        if i % 5 == 0:  
            print(i, end=', ')
```

```
#-----
```

```
#-----
```

Write a Python program to construct the following pattern, using a nested for loop.

```
# *  
# * *  
# * * *  
# * * * *  
# * * * * *  
# * * * *  
# * * *  
# * *  
# *
```

```
for i in range(1, 10):  
    print('*' * (i * (i<6)) + '*' * ((10-i)*(i>5)))
```

```
#-----
```

```
#-----
```

```
# Write a Python program that allows the user to enter a four-digit binary number
# and displays its value in base 10. Each binary digit should be entered one per
# line, starting with the leftmost digit , as shown below.
# Enter leftmost digit: 1 Enter the next digit: 0
# Enter the next digit: 0 Enter the next digit: 1
# The value is 9
```

```
one = int(input( 'Enter leftmost val  : '))
two = int(input( 'Enter next val      : '))
four = int(input( 'Enter next val      : '))
eight = int(input('Enter rightmost val : '))
```

```
val = eight * 8 + four * 4 + two * 2 + one * 1
```

```
print(val)
```

```
#-----
```

```
#-----
```

```
# Design and implement a program
# 1) To prompt the user for hours ad rate per hour to compute gross pay
# 2) To print the python version that is used.
```

```
# didn't understand the question
```

```
#-----
```

```
#-----
```

```
# Write a Python program to check the validity of password input by users. Go
# to the editor
# Validation :
# At least 1 letter between [a-z] and 1 letter between [A-Z].
# At least 1 number between [0-9].
# At least 1 character from [$#@].
# Minimum length 6 characters.
# Maximum length 16 characters
```

```
import re
password = input('password: ') # eg: wewewetRRRRTY45@
```

```
def e() :
    print("Invalid password, please choose a different password")
    quit()
```

```
if (len(password)<6):
    e()
elif (len(password)>16):
    e()
elif not re.search("[a-z]", password):
    e()
elif not re.search("[A-Z]", password):
    e()
elif not re.search("[0-9]", password):
    e()
elif not re.search("[$#@]", password):
    e()
else:
    print("Good password")
```

```
#-----
```

```
#-----
```

```
# Write a Python program to find numbers between 100 and 400 (both
# included) where each digit of a number is an even number. The numbers
# obtained should be printed in a comma-separated sequence
```

```
def all_even() :
    for i in range(100, 401, 2) :
        if not i//100 % 2 :
            if not i//10 % 2 :
                print(i, end=', ')
```

```
all_even()
#-----
```

```
#-----
```

```
# Write a Python program that accepts a string and calculate the number of
# digits and letters.
# Sample Data : Python 3.2
# Expected Output :
# Letters 6
# Digits 2
```

```
ip_str = input("Enter a string: ")
digits = letters = 0
for c in ip_str:
    if c.isdigit():
        digits += 1
    elif c.isalpha():
        letters += 1
    else:
        continue
print("digits: ", letters)
print("letters: ", digits)
#-----
```

```
#-----
```

```
# Write a Python program to convert temperatures to and from celsius,
# fahrenheit. [ Formula : c/5 = f-32/9 [ where c = temperature in celsius and f =
# temperature in fahrenheit ]
# Expected Output : 60°C is 140 in Fahrenheit.45°F is 7 in Celsius
```

```
unit = input('Enter C2F (celsius to fahrenheit) or F2C (fahrenheit to celsius) : ')
val = int(input('enter temperature value: '))
```

```
def convert(val, unit) :
    if unit == 'C2F' :
        valn = (val * 1.8) + 32
        print(f'{val}°C is {valn} in Fahrenheit.')
    elif unit == 'F2C' :
        valn = (val - 32) * (5 / 9)
        print(f'{val}°F is {valn} in Celsius.')
```

```
    else:
        print('error : try again')
```

```
convert(val, unit)
```

```
#-----
```

```
#-----
```

```
# Write a Python program to count the number of even and odd numbers from  
# a series of numbers
```

```
num_list = [1, 2, 3, 4, 5, 6]  
l = len(list(filter(lambda x: x % 2, num_list)))  
print(f'Number of even numbers: {len(num_list) - l}')  
print(f'Number of odd numbers: {l}')  
#-----
```

```
#-----
```

```
# Write a Python program to check a string represent an integer or not.  
# Expected Output:  
# Input a string: Python  
# The string is not an integer.
```

```
string = input('Input a string: ')
```

```
def int_or_str(str) :  
    try:  
        int(str)  
        print('The string is an integer.')  
    except:  
        print('The string is not an integer.')
```

```
int_or_str(string)  
#-----
```

```
#-----
```

```
# Write a Python program to calculate the difference between the squared sum  
# of first n natural numbers and the sum of squared first n natural numbers
```

```
n = int(input('Enter value of n: '))
```

```
sq_sum = sum([i for i in range(0, n)] ** 2)  
sum_sq = sum([i ** 2 for i in range(0, n)])
```

```
print(sq_sum - sum_sq)  
#-----
```

```
#-----
```

```
# Write a Python program to convert a decimal number to binary number.
```

```
n = int(input('Enter a decimal number: '))
```

```
def dec_to_bin(n):
    if n > 1:
        dec_to_bin(n//2)
    print(n % 2,end = '')
```

```
dec_to_bin(n)
```

```
#-----
```

```
#-----
```

```
# Write a Python program to create the multiplication table (from 1 to 10) of a
# number.
```

```
n = int(input('Enter a number: '))
```

```
[print(f'{n} x {i} = {n * i}') for i in range(1, 11)]
```

```
mul_tab(n)
```

```
#-----
```

```
#-----
```

```
# Write a Python program to construct the following pattern, using a nested
# loop number
# Expected Output:
```

```
# 1
# 22
# 333
# 4444
# 55555
# 666666
# 7777777
```

```
[print(str(i) * i) for i in range(1, 8)]
```

```
#-----
```

```
#-----
```

```
# q1 : Design and implement a python program to get a new string from a give
# string where 'is' has been added to the front. If the string already begins with
# "is" the return the string unchanged.
```

```
# answer
```

```
def add_is(ip_string) :
    return ip_string if ip_string[0:2] == 'is' else 'is' + ip_string
```

```
print(add_is(''))
```

```
# o/p : 'is'
```

```
#-----
```

```
#-----
```

```
# Write a Python program to get a single string from two given strings,
# separated by a space and swap the first two characters of each string.
# Sample String : '&#39;abc&#39;, '&#39;xyz&#39;
```

Expected Result : 'xyz'

```
str1 = 'abcd'
```

```
str2 = 'xyzd'
```

```
string = str2[:2] + str1[1:-1][::-1] + ' ' + str1[:2] + str2[1:-1][::-1]
```

```
print(string)
```

```
#-----
```

```
#-----
```

Write a Python program to count the number of even and odd numbers from

a series of numbers

```
num_list = [1, 2, 3, 4, 5, 6]
```

```
l = len(list(filter(lambda x: x % 2, num_list)))
```

```
print(f'Number of even numbers: {len(num_list) - l}') 
```

```
print(f'Number of odd numbers: {l}') 
```

```
#-----
```

```
#-----
```

Write a Python program to get a string from a given string where all

occurrences of its first char have been changed to '\$', except the first char

itself.

Sample String : 'restart'

Expected Result : 'resta\$t'

```
string = input('Enter a string: ')
```

```
print(string[0], end='') 
```

```
[print('$',end='') if c == string[0] else print(c,end='') for c in string[1:]] 
```

```
#-----
```

```
#-----
```

Design and implement a program that evaluates the value of the quadratic

equation $ax^2 + bx + c$ by prompting the user to enter the values of a,b,c and x.

answer

```
def quad_eval(a, b, c, x):
```

```
    return a * x ** 2 + b * x + c
```

```
a, b, c, x = map(int,input('Enter values of a, b, c, x respectively, seperated by a space: ').split())
```

```
print('ax^2 + bx + c = ',quad_eval(a, b, c, x))
```

o/p : Enter values of a, b, c, x respectively, seperated by a space: 1 1 1 1

$ax^2 + bx + c = 3$

```
#-----
```

```
#-----
```

Write a Python program to check a triangle is equilateral, isosceles or scalene.


```
# answer
```

```
def abt_triangle(x, y, z) : # input length of sides
    return 'equilateral triangle' if x == y == z else 'isocoles triangle' if x == y or y == z or z == x
else 'scalene triangle'
```

```
print(abt_triangle(1,2,3))
```

```
# o/p: scalene triangle
```

```
#-----
```

```
#-----
```

```
# design and implement a python program that accepts an integer n and
# computes the value of n+nn+nnn
```

```
# answer
```

```
def cal_n_eq(n) :
    return sum(map(int, [str(n) * i for i in range(1,3)]))
```

```
print(cal_n_eq(1))
```

```
# o/p : 123
```

```
#-----
```

```
#-----
```

```
# Design and implement a python program to test whether a number is within
# 100 of 1000 Or 2000.
```

```
# didn't understand the question
```

```
#-----
```

```
#-----
```

```
# Write a program that sums a series of (positive) integers entered by the user,
# excluding all numbers that are greater than 100.
```

```
# answer
```

```
num_arr = map(int,input('Enter number array seperated by spaces: ').split())
def sum_99(num_list) :
    return sum(filter(lambda n: n < 101, num_list))
```

```
print(sum_99(num_arr))
```

```
# eg : [1, 99, 2, 100, 101]
```

```
# o/p : 202
```

```
#-----
```

```
#-----
```

```
# Write a python program that asks the user how many coins of various types
# they have, and then print the total amount of money in rupees.
```

```
# didn't understand the question
```

```
#-----
```

```
#-----
```

```
# Write a Python program to get the Fibonacci series between 0 to 50.
```

```
def fib(n):
```

```
    a, b = 0, 1
```

```
    for _ in range(n):
```

```
        yield a
```

```
        a, b = b, a + b
```

```
print(list(fib(50)))
```

```
# o/p :
```

```
# [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597,
```

```
# 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811,
```

```
# 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352,
```

```
# 24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437,
```

```
# 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049]
```

```
#-----
```

```
#-----
```

Unit 2

```
# Write a Python program to remove duplicates from a list.
```

```
l = input('Enter list elements separated by space: ').split()
```

```
nl = list(set(l))
```

```
print(nl)
```

```
#-----
```

```
#-----
```

```
# Write a Python function that accepts a string and calculate the number of
```

```
# upper case letters and lower case letters.
```

```
# Sample String : &#39;The quick Brown Fox&#39;
```

```
# Expected Output :
```

```
# No. of Upper case characters : 3
```

```
# No. of Lower case Characters : 13
```

```
def upper_lower(s):
```

```
    d={"U":0, "L":0}
```

```
    for c in s:
```

```
        if c.isupper():
```

```
            d["U"]+=1
```

```
        elif c.islower():
```

```
            d["L"]+=1
```

```
        else:
```

```
            pass
```

```
print ("No. of Upper case characters : ", d["U"])
print ("No. of Lower case Characters : ", d["L"])
```

```
upper_lower('The quick Brown Fox')
```

```
#-----
```

```
#-----
```

```
# Write a Python function that takes a list and returns a new list with unique
# elements of the first list
# Sample List : [1,2,3,3,3,3,4,5]
# Unique List : [1, 2, 3, 4, 5]
```

```
l = input('Enter list elements seperated by space: ').split()
```

```
nl = list(set(l))
print('Original list: ', l)
print('New list: ', nl)
```

```
#-----
```

```
#-----
```

```
# Write a Python function that takes a number as a parameter and check the
# number is prime or not
```

```
def prime_test(n):
    if (n==1):
        return 'Number is not a prime'
    elif (n==2):
        return 'Number is prime'
    else:
        for x in range(2,n):
            if(n % x==0):
                return 'Number is not a prime'
        return 'Number is prime'
```

```
print(prime_test(9))
```

```
#-----
```

```
#-----
```

```
# Write a Python program to check whether a list contains a sublist.
```

```
l1 = input('Enter list1 elements seperated by space: ').split()
l2 = input('Enter list2 elements seperated by space: ').split()
```

```
def is_sublist(lst1, lst2):
    ls1 = [element for element in lst1 if element in lst2]
    ls2 = [element for element in lst2 if element in lst1]
    return ls1 == ls2
```

```
print(is_sublist(l1, l2))
```

```
#-----
```

```
#-----
```

Define a function which can generate a dictionary where the keys are
numbers between 1 and 20 (both included) and the values are square of keys.
The function should just print the keys only.

```
def sqr_dict() :  
    d = {}  
    for x in range(1, 21):  
        d[x] = x**2  
  
    print(d.keys())
```

```
sqr_dict()
```

```
#-----
```

```
#-----
```

Write a program which count and print the numbers of each character in a
string input by console.
Example:
If the following string is given as input to the program:
abcdeabc
Then, the output of the program should be:
a,2 c,2 b,2 e,1 d,1

```
string = input("Enter a string: ")
```

```
str_list = list(string)
```

```
count_list=[]
```

```
for s in str_list:  
    if s not in count_list:  
        count_list.append(s)  
        count = 0  
    for i in range(len(str_list)):  
        if s == str_list[i]:  
            count+=1
```

```
    print(f"{s},{count}", end=' ')
```

```
#-----
```

```
#-----
```

Write a Python program to check whether a list contains a sublist.

```
l1 = input('Enter list1 elements seperated by space: ').split()  
l2 = input('Enter list2 elements seperated by space: ').split()
```

```
def is_sublist(lst1, lst2):  
    ls1 = [element for element in lst1 if element in lst2]  
    ls2 = [element for element in lst2 if element in lst1]  
    return ls1 == ls2
```

```
print(is_sublist(l1, l2))
```

```
#-----
```

```
#-----

# Write a program to generate and print another tuple whose values are even
# numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

t = (1,2,3,4,5,6,7,8,9,10)

et = tuple(filter(lambda x: not x % 2, t))

print(et)

#-----

#-----

# Write a function called count_values that takes a single dictionary as an
# argument and returns the number of distinct values it contains. Given the
# input {'red':1,'green':1,'blue':2}, for example it should return 2.

d = {'red':1,'green':1,'blue':2, 'yellow':3}

def count_values(dic) :
    print('No. of distinct values: ',len(set(dic.values())))

count_values(d)

#-----

#-----

# Design and implement a python program to get a string made of the first 2
# and the last 2 chars from a given string. If the string length is less than 2
# return instead the empty string.

string = input('Enter a string: ')

def strman(string) :
    if len(string) < 2 :
        print('')
    else :
        print(string[:2] + string[:: -1][2:][::-1])

strman(string)

#-----

#-----

# Write a Python program to generate and print a list except for the first 5
# elements, where the values are square of numbers between 1 and 30 (both
# included)

#-----

#-----
```

```
# Write a Python Program using function to Sort Words in Alphabetic
# Order
```

```
def alpha_sort(string) :
    lstr = string.split()
    lstr.sort()
    print("The sorted words are:")
    for w in lstr:
        print(w)
```

```
string = 'Write a Python Program using function to Sort Words in Alphabetic'
```

```
alpha_sort(string)
```

```
#-----
```

```
#-----
```

```
# Write a program to read details of the student(name as string ,usn as int
# ,semester as int) in comma separated values from the keyboard and split &
# typecast the input based on comma delimiter and store it in list before
# displaying it.
```

```
student_details = input('Enter student details seperated by comma: ')
```

```
name, usn, semester = student_details.split(',')
```

```
student_details_list = [name, int(usn), int(semester)]
```

```
print(student_details_list)
```

```
#-----
```

```
#-----
```

```
# Design and implement a python program to count the number 4 in a given list.
```

```
num_list = [1, 2, 3, 4 , 4, 5, 4]
```

```
print(len(list(filter(lambda x: x == 4, num_list))))
```

```
#-----
```

```
#-----
```

```
# Design and implement a python program to generate and print list except the
# first 5 elements, where the values are square of numbers between 1 and
# 30(both included).
```

```
#-----
```

```
#-----
```

```
# Design and implement a python function that takes two lists and returns true
# if they have at least one common.
```

```
l1 = input('Enter list1 elements seperated by space: ').split()
```

```
l2 = input('Enter list2 elements seperated by space: ').split()
```

```
def common_ele(l1, l2) :  
    s = set(l1).intersection(set(l2))  
    return True if len(s) > 0 else False
```

```
print(common_ele(l1, l2))
```

```
#-----
```

```
#-----
```

```
# Design and implement a python program to find the list of words that are no  
# longer than n from a given list of words.
```

```
def words(n, str):  
    nword = []  
    txt = str.split(" ")  
    for x in txt:  
        if len(x) > n:  
            nword.append(x)  
    return nword
```

```
print(words(3, "wubba lubba dub dub"))
```

```
#-----
```

```
#-----
```

```
# Design and implement a program to count the number of vowels present in a  
# given list of strings using 'in' or 'not in' operator.
```

```
def count_vowels(str):  
    count = 0  
    vowel = set("aeiouAEIOU")  
    for c in str:  
        if c in vowel:  
            count = count + 1
```

```
    print("No. of vowels :", count)
```

```
string = input("Enter string: ")
```

```
count_vowels(string)
```

```
#-----
```

```
#-----
```

```
# Write a function char_freq() that takes a string and builds a frequency listing  
# of the characters contained in it. Represent the frequency listing as a python  
# dictionary.
```

```
def char_freq(ip_string) :  
    count = {}  
    for c in ip_string :  
        try :  
            count[c] += 1  
        except :  
            count[c] = 1  
    return count
```

```
string = input("Enter string: ")
```

```
print(char_freq(string))
```

```
#-----
```

```
#-----
```

```
# Design and implement a program to count the number of vowels present in a  
# given list of strings using 'in' or 'not in' operator.
```

```
def count_vowels(str):
```

```
    count = 0
```

```
    vowel = set("aeiouAEIOU")
```

```
    for c in str:
```

```
        if c in vowel:
```

```
            count = count + 1
```

```
    print("No. of vowels :", count)
```

```
string = input("Enter string: ")
```

```
count_vowels(string)
```

```
#-----
```

```
#-----
```

```
# Write a function called dict_intersect that takes two dictionaries as arguments  
# and returns a dictionary that contains only the key/value pairs found in both  
# of the original dictionaries.
```

```
a = {'s': 'ssss', 'n': 'nnnnn'}
```

```
b = {'s': 'ssss', 'k': 'kkk'}
```

```
def dict_intersect(a, b) :
```

```
    ak = set(a.keys())
```

```
    bk= set(b.keys())
```

```
    common = ak & bk
```

```
    return common
```

```
print(dict_intersect(a, b))
```

```
#-----
```

```
#-----
```