

while loop solved programs

1. print numbers from 1 to 10

```
print("numbers from one to ten")
num = 1
while num <=10:
    print(num, end=", ")
    num += 1
```

2. print numbers from 10 to 1

```
a=10
while a>=1:
    print(a)
    a-=1
```

3. wap to print even numbers from zero to 50

```
a=0
while a<=50:
    if a %2==0:
        print(a)
    a+=1
```

4. wap to print odd numbers from 0 to 50

```
a=0
while a<=50:
    if a %2!=0:
        print(a)
    a+=1
```

5. wap to iterate over a string and print each character separating by ",".

```
str1 = input()
c = 0
print(f"the string is {str1} and length of the string is {len(str1)}")
while c < len(str1):
    print(str1[c], end=", ")
    c += 1
```

6. write a program to print alternative character of a string.

```
str1 = input()
c = 1
print(f"the string is {str1} and length of the string is {len(str1)}")
while c < len(str1):
    print(str1[c], end=", ")
    c += 2
```

7. wap to read a string from user and print each character in both forward and reverse direction.

```
str1 = input()
c = 0
while c < len(str1):
    print(str1[c], end=" ")
    c += 1
print()
r = len(str1)-1
while r >= 0:
    print(str1[r],end=" ")
    r -= 1
```

8. write a program to print all the vowels present inside a given string.

```
print("enter any string")
str1=input()
c=0
while c<len(str1):
    if str1[c] in "aeiouAEIOU":
        print(str1[c])
    c+=1
```

9. wap to display all positions of substrings in a given main string.

```
s = input()
sub_s = input()
p = -1
n = len(s)
while len(s) > p:
    p = s.find(sub_s,p+1,n)
    if p == -1:
        break
    print(p)
```

- 10.wap to reverse the order of words from the sentence.

```
str2=input()
l=str2.split()
str3=""
c=len(l)-1
while 0<=c:
    str3+=l[c]+" "
    c-=1
print(str3)
```

11.wap to print all the consonents from the given string.

```
print("enter the string")
a=input()
c=0
while c<len(a):
    if a[c] not in "aeoiuAEIOU1234567890":
        print(a[c],end = " ")
    c+=1
```

12.wap to print sum of digits from the given string.

```
str1 = input()
c = 0
sum = 0
while c <= len(str1)-1:
    if str1[c] in "0123456789":
        sum = sum + int(str1[c])
    c += 1
print(sum)
```

13.wap to iterate over a string and print each character twice

```
str1 = input()
c = 0
while c < len(str1):
    res = str1[c], str1[c]
    print(res)
    c += 1
```

14.wap to reverse internal content of each word.

```
print("enter the sentences")
str2=input()
l=str2.split()
str3=""
c=0
while c<len(l):
    str3+=l[c][::-1]+" "
    c+=1
print(str3)
```

15.wap to print the characters in odd position and even position separately

```
print("enter the string")
str3=input()
c=0
str4=""
```

```

str5=""
while c< len (str3):
    if c %2==0 :
        str4+=str3[c]
    else:
        str5+=str3[c]
    c+=1
print(f"even position charecters {str4}")
print(f"odd position char {str5}")

```

16. wap to merge 2 characters of 2 strings into a single string by taking characters alternatively. (s1 = good, s2 = well o/p: gwoeoldl)

```

s1 = input()
s2 = input()
s3 = ""
i = 0
j = 0
while i < len(s1) or j < len(s2):
    if i < len(s1):
        s3 += s1[i]
        i += 1
    if j < len(s2):
        s3 += s2[j]
        j += 1
print(s3)

```

- 17.wap to display the sum of first n numbers.

```

print("enter value of n")
n = int(input())
sum = 0
c = 0
while c <= n:
    sum = sum + c
    c += 1
print(sum)

```

18. wap to achieve given output.

```

i/p : "one two three four five six seven"
o/p : "one owt three ruof five xis seven"
print("enter any sentences")
str1=input()
c=0
str2=""
l=str1.split()
while c<len(l):
    if c %2==0:
        str2 +=l[c]+" "

```

```

else:
    str2+=l[c][::-1]+ " "
    c+=1
print(str2)

```

- 19.wap to enter name, percentage and marks in a dictionary and print information on the screen.

```

print("enter name :")
name=input()
print("enter the percentage:")
percentage = float(input())
print("enter the marks:")
marks=int(input())
d={ }
d.setdefault("name",name)
d.setdefault("percentage",percentage)
d.setdefault("marks",marks)
print(d)

```

- 20.wap to prompt user to enter some name until they enter "Python"

```

name = ""
while name != "python":
    name = input()
print("prompted python")

```

- 21.wap to iterate in a list check for even numbers inside list and add them

```

l = [10,20,25,30,45,60]
c = 0
sum = 0
while c < len(l):
    if isinstance(l[c],int) and l[c] % 2 == 0:
        sum = sum + l[c]
    c += 1
print(sum)

```

- 22.wap to iterate inside dictionary and fetch each key separately.

```

d2={ 1:10,2:20, 'b':"bat",3:30, "a":"apple",4:60}
l1=list(d2.keys())
c=0
while c<len(l1):
    print(l1[c],end=" ")
    c+=1

```

- 23.wap to iterate inside dictionary fetch all the keys. if keys are integers multiply them

```

d = { 'a': 100, 455:200, 'b':300, 60:400}
dkey = list(d.keys())          #dkey=[a,455,b,60]
print(dkey)

```

```

c = 0
mul = 1
while c <= len(dkey):

    if isinstance(dkey[c],int):
        mul = mul * dkey[c]
    c += 1
print(mul)

```

24. wap to iterate in a list if list elements are string type then store them in separate list

```

list1 = ["hello", 10, 20.55, True, "hai", "bye"]
list2 = []
c = 0
while c < len(list1):
    if isinstance(list1[c],str):
        list2.append(list1[c])
    c+=1
print(list2)

```

25. wap to iterate inside a list check if it has nested list if yes merge them.

```

list1 = ["hello", 10, 20.55, True, "hai", "bye",[25,90,False,"good"]]
list2 = []
c = 0
while c < len(list1):
    if isinstance(list1[c],list):
        list2.extend(list1[c])
        list1.remove(list1[c])
    c += 1
print(list1+list2)

```

26. create a list using while loop, take list elements from user.

```

l = []
c = 1
while c <= 5:
    l.append(input())
    c+= 1
print(l)

```

27. wp to print 2 tables.

```

n = 1
while n<=10:
    res = 2 * n
    print(f"2 * {n} = {res}")
    n += 1

```

28. wp to print sum of 'n' natural number.

```

num = eval(input("enter value of 'N':"))

```

```

n = 1
res = 0
while n<=num:
    res = res + n
    n += 1

print(res)

```

29. wp ro print 2 table in reverse order. $2 * 10 = 20$, $2 * 9 = 18$, ... $2 * 1 = 2$

```

n = 10
while n>=1:
    res = 2*n
    print(f"2 * {n} = {res}")
    n -= 1

```

30. wp to print name which are starts with vowels in given list

```

l = ["Asha", "Sushma", "Nishma", "Oshma", "Eshma", "Kashma"]
n = 0
while n<=len(l)-1:
    if l[n][0].lower() in "aeiou":
        print(f"{l[n]} starts with vowels")
    n += 1

```

31. wp to print name only if the length of the name is even.

```

l = ["Rama", "Bhema", "Shamamama", "Komaa", "Soma", "Obbama"]
n = 0
while n<=len(l)-1:
    if len(l[n]) % 2 == 0:
        print(f"{l[n]} is a even length name")
    n += 1

```

32. wp to print the elemnts in tuple only if it is collection data type.

```

l = (10, 2.5, [10, 20], True, (33, 22), {30, 10}, 2.2, 3+5j, {"a":10}, "last")
n = 0
while n<=len(l)-1:
    if isinstance(l[n], (set, str, list, tuple, dict)):
        print(f"{l[n]} is a collection data type")
    n += 1

```

For loop solved programs

1. wap to traverse through a string and print each character.

```
str1 = input()
for a in str1:
    print(f'characters present in the string :{a} ")
(or)
str1 = input()
for i in range(len(str1)):
    print(str1[i],end=" ")
```

2. wap to traverse through a dictionary

```
d = {'a':"hello", 'b':100, 'c':45.55, 'd':True}
for rv in d:
    print(d[rv])
```

3. wap to traverse through an iterable and fetch each element.(all list,set,tuple)

```
iter1 = [10,20,30,40,50]
for rv in iter1:
    print(rv)
```

```
s = {10,20,"hello",True,75.55}
for rv in s:
    print(rv)
```

4. wap to iterate over a string in reverse direction.

```
str1 = input()
for ch in range(-1,-1-len(str1),-1):
    print(str1[ch])
```

```
for ch in str1[::-1]:
    print(ch)
```

5. wap to print index as well as character of the string

```
str1 = input()
for index in range(0,len(str1)):
    print(index, str1[index])
or
str1 = input()
for i in enumerate(str1):
    print(i)
```

6. wap to access dictionary keys and values using enumerate and other methods.


```

d = {1:100,2:200,3:300,4:400}
for rv,key in enumerate(d):
    print(key,end = " ")
    #accessing dictionary keys
    for key in d:
        print(key, end = " ")
    print()
    for key in d.keys():
        print(key, end="*")
    print()
    for key in d.items():
        print(key[0])
    print()
    for key,value in d.items():
        print(key)
        print(value)

```

```

# accessing values
for key in d:
    print(d[key], end=",")
for value in d.values():
    print(value, end=" ")
for value in d.items():
    print(value[1])
for key, value in d.items():
    print(value)
    print(key, value)

```

8. wap to create a dictionary of characters and its ASCII value pair by taking string as input

```

d.update({ch:ord(ch)})
str1 = input()
d = { }
for ch in str1:
    #d[ch] = ord(ch)
    d[ord(ch)] = ch
print(d)

```

9. wap to create a dictionary of word and its length pair from a string(sentence).

```

s = "python is an easiest programming language"
d = { }
word = s.split()
for rv in word:
    d[rv] = len(rv)
print(d)

```

10. wap to create a dictionary of index and character pair of a string.

```
str1 = "good morning"
d = {}
for key in str1:
    d[key] = str1.index(key)
print(d)
or
str1 = "good morning"
d = {}
for index, value in enumerate(str1):
    d[index] = value
print(d)
```

11. wap to create a dictionary with word and its count pair from the given sentence.

```
s = "my day is good day or bad day or just a day"
str1 = "my day is good day or bad day or just a day"
w_list = str1.split()
print(w_list)
d = {}
for rv in w_list:
    d[rv] = w_list.count(rv)
print(d)
```

12. wap to print fibonacci series.

```
n1 = 0
n2 = 1
f_s = 10
for r_v in range(f_s):
    print(n1, end = " ")
    next_ = n1 + n2
    n1 = n2
    n2 = next_
print(f"the 10th fibonacci value is: {n1}")
```

13.wap to print nth fibonacci value

```
f_s = int(input())
n1 = 0
n2 = 1
for r_v in range(f_s):
    print(n1, end=" ")
    next_ = n1 + n2
    n1 = n2
    n2 = next_
print()
print(f"the 10th fibonacci value is: {n1}")
```

14.wap to print prime numbers from 1 to 10

```
num = int(input())
for i in range(2,num-1):
    if num % i == 0:
        break
else:
    print("its a prime number")
```

15. wap to print nth prime numbers.(while loop, one extra variable)

```
num = int(input())
for i in range(2,num-1):
    if num % i == 0:
        break
else:
    print("its a prime number")
```

16. wap to use enumerate,reversed,zip and zip_longest with any iterable.

```
str1 = "hello"
str2 = "good"
print(list(enumerate(str1)))
print(tuple(reversed(str1)))
print(list(zip(str1,str2)))
from itertools import zip_longest
print(list(zip_longest(str1,str2)))
print(list(zip_longest(str1,str2, fillvalue=5)))
print(list(zip_longest(str1,str2, fillvalue='d')))
```

17.wap to check the very first integer which is greater than 400 and print it.

```
l = [10,50,80,600,400,300,100]
for rv in l:
    if rv > 400:
        print(f"greater value: {rv}")
        break
print(rv)
```

18.wap to print odd numbers 0 to 9 using continue keyword

```
for rv in range(10):
    if rv % 2 == 0:
        continue
    print(rv)
```

19. demonstrate using of pass keyword

```
for rv in range(10):
    pass
```

```
if 10 > 100:
    pass
print("passed the for loop")
or
num = 0
while num < 10:
    num += 1
    pass
print("passed the while loop")
```

20. wap to print 1 to 20 numbers using break statement in while loop:

```
num = 1
while num <= 100:
    print(num)
    if num == 20:
        break
    num += 1
```

21.wap to print 1 to 50 odd numbers using continue with while loop

```
num = 0
while num < 50:
    num += 1
    if num % 2 == 0:
        continue
    print(num, end=" ")
```

22.wap to check whether the elements of list are lesser than 100 using nested loops and loops with else block

```
l1 = [10,20,30,40,50,60]
for rv in l1:
    if rv >= 100:
        print("element is greater than 100")
        break
    print(rv)
else:
    print("every value is valid")
```

23.wap to print 0 1 0 0 1 1 using nested while loop

```
a = 0
while a < 2:
    b = 0
    while b < 2:
        print(b,end=" ")
        b += 1
```

```
print(a, end=" ")
a += 1
```

24. wp to print index and character of string in tuples

```
string = "hello world"
for index in range(len(string)):
    print((index, string[index]), end=" ")
```

Or

```
for item in enumerate(string):
    print(item, end=" ")
```

25. demonstrate packing and unpacking of different iterables using zip

```
l = [10, 20, 30]
s = "hai"
for item in zip(l, s):    # [(10, 'h'), (20, 'a'), (30, 'i')] --> item - (10, h)
    print(item)
```

```
for ele1, ele2 in zip(l, s):
    print(ele1, ele2)
```

26. wp to print first and last char of each name in the list

```
l = ["sunil", "anil", "suresh", "mahesh", "dinesh"]
for name in l:    # name="sunil"
    print(name[0], name[-1])
```

27. wp to create new list as squares of each number of below list

```
l = [2, 4, 5, 1, 9, 7, 3]
res = []
for i in l:    # i = 2 i = 4, .....
    res.append(i**2)
```

```
print(res)
```

28. wp if number is even the print its square else print its cube

```
l = [2, 4, 5, 1, 9, 7, 3]

for i in l:    # i = 2
    if i % 2 == 0:
        print(f"the square of {i} is {i**2}")
    else:
        print(f"the cube of {i} is {i**3}")
```

29. wp to create a new list of separate even number and odd number

```
l = [2, 4, 5, 1, 9, 7, 3]
```

```

evn = []
odd = []

for i in l:
    if i % 2 == 0:
        evn.append(i)
    else:
        odd.append(i)

```

```

print(evn)
print(odd)

```

30. wp to create a list with square and cube of each number

```

l = [2, 4, 5, 1, 6, 9, 7, 3]
res = []
for i in l:
    res.append((i**2, i**3))

print(res)
[(4, 8), (16, 64), (25, 125), (1, 1), (36, 216), (81, 729), (49, 343), (9, 27)]

```

31. wp to create a new list of reversing each name from the list.

```

names = ["sunil", "dinga", "penga", "harsha", "manga"]
res = []
for name in names:
    res.append(name[::-1])
print(res)

```

32. wp to create a new list of company name in the given emails list.

```

emails = ["abc@gmail.com", "xyz@yahoo.com", "ding@hotmail.com",
"mang@redifmail.com", "peng@orkut.com"]
ids = []
for mail in emails:
    words = mail.split("@")    #["abc", "gmail.com"]
    ids.append(words[1])

print(ids)

```

33. wp to create a new list of individual and collection data type from list

```

datas = [20.12, True, [10, 20], {1, 2}, 2+6j, {"a":10}, 100]
ind = []
col = []
for data in datas:
    if isinstance(data, (int, float, complex, bool)):
        ind.append(data)

```

```
else:  
    col.append(data)
```

```
print(ind)  
print(col)
```

34.wp to create a new list, if the name is palindrome then convert to upper case and add to list, read names from tuple.

```
names = ("mom", "Komal", "pop", "nayan", "malayalam", "eye")
```

```
n = []
```

```
for name in names:          #name=mom  
    res = "  
    for i in range(len(name)-1, -1, -1):    #i=2  
        res += name[i]  
    if res == name:  
        n += [res.upper()]  
print(n)
```

35.wp to create a list with fruit name and along with index

```
fruits = ["apple", "orange", "banana", "mango", "grapes"]
```

```
logic1
```

```
n = 0
```

```
f = []
```

```
for fruit in fruits:  
    f.append((fruit,n))  
    n += 1
```

```
print(f)
```

```
or
```

```
logic2
```

```
f = []
```

```
for i in range(0, len(fruits)):
```

```
    f.append((fruits[i],i))
```

```
print(f)
```

```
[('apple', 0), ('orange', 1), ('banana', 2), ('mango', 3), ('grapes', 4)]
```

36.wp to find factorial of given number

```
n = eval(input("enter number:"))
```

```
i = 1
```

```
res = 1
```

```
while i<=n:
```

```
    res = res * i
```

```
    i += 1
```

```
print(res)
```

```
or
```

```
for loop
```

```
n = eval(input("enter number:"))
```

```

res = 1
for i in range(1, n+1):
    res = res * i
print(res)

```

37. wp to create a list with name and no. of occurrence.

```

i/p:- names = ["gmail", "yahoo", "fb", "google", "gmail", "yahoo", "fb", "google", "yahoo",
"fb"]
o/p: [(gmail, 2), (yahoo,3),...]
names = ["gmail", "yahoo", "fb", "google", "gmail", "yahoo", "fb", "google", "yahoo", "fb"]
res = []
for name in names:
    res.append((name,names.count(name)))

print(set(res))
{'fb', 3), ('gmail', 2), ('google', 2), ('yahoo', 3)}

```

38. wp to create a list of festival which are marked with holiday

```

i/p: holidays = {"Pongal":"Holiday","Ugadi":"No
holiday","Navami":"Doubt","RepublicDay":"Holiday","SRatri":"Fasting"}
o/p: ["Pongal", "RepublicDay"]
holidays = {"Pongal":"Holiday","Ugadi":"No
holiday","Navami":"Doubt","RepublicDay":"Holiday","SRatri":"Fasting"}
hol=[]
for item in holidays.items():
    if item[1].lower() == "holiday" :
        hol.append(item[0])

print(hol)

```

39. wp to create a list if the price if the product is more than 1000 then add the product name to a list

```

products = {"Asian-paint":4000, "Mobile":90000, "Hen":500, "Pen":5, "Watch":2500}
pro = []
for item in products.items():
    if item[1]>1000:
        pro.append(item[0])

print(pro)

```

40. wp create a list, if a key is a individual DT then add key and data type as its value

```

i/p:- elemnts = {10:"Ind", "hai":"Colle", [10]:"Colction", True:"Inddual", (11,22):"Colltion"}
o/p [(10, <class 'int'>), (True, <class 'bool'>)]
res = []

for key in elemnts:

```



```
if isinstance(key, (int, float, bool, complex)):
    res.append((key, type(key)))
```

```
print(res)
```

41. wp to print author name in books dictionary

```
books = {"loverstory":["Harish",30], "biography":["Padam", 150], "animals":["vimala",75],
"songs":["venkat",90]}
for author,price in books.values():
    print(author)
```

42.wp to print book name along with author

```
for key,val in books.items():
    print(key,val[0])
```

43. wp to create a dictionary name and its count pair

```
names = ["gmail", "fb", "insta", "google", "gmail", "yahoo", "fb", "fb"]
d = { }
for name in names:
    d[name] = names.count(name)
print(d)
{'gmail': 2, 'fb': 3, 'insta': 1, 'google': 1, 'yahoo': 1}
```

44.wp to create a dictionary characers and its count pair

```
chars = ['a', 'M', 'i', 'A', 'M', 'h', 'i', 'H', 'a', 'B']
d = { }
for chr in chars:
    d[chr] = chars.count(chr)

print(d)
{'a': 2, 'M': 2, 'i': 2, 'A': 1, 'h': 1, 'H': 1, 'B': 1}
```

45.wp to create a dictionary with mobile and its count pair with and without inbuilt function

```
mobiles = ["mi", "iphone", "vivo", "poco", "samsung", "iphone", "mi", "vivo", "mi",
"relame"]
d = { }
for mobile in mobiles:
    d[mobile] = mobiles.count(mobile)
print(d)
or
d = { }
for mobile in mobiles:
    if mobile not in d:
        d[mobile] = 1
    else:
        d[mobile] += 1
```

```
print(d)
{'mi': 3, 'iphone': 2, 'vivo': 2, 'poco': 1, 'samsung': 1, 'relame': 1}
```

46.wp to create a dictionary character and its count pair

```
chars = ['A', "s", "o", "P", "O", "S", "H", "J", "A", "V", "H", "o"]
d = {}
for ch in chars:
    if ch not in d:
        d[ch] = 1
    else:
        d[ch] += 1
```

```
print(d)
{'A': 2, 's': 1, 'o': 2, 'P': 1, 'O': 2, 'S': 1, 'H': 2, 'J': 1, 'V': 1}
```

47.wp to create a dictionary domain and its count pair

```
urls = ["www.gmail.com", "www.upsc.gov.in", "www.irctc.in", "www.qsp.edu.in",
"www.twitter.com", "www.python.org"]
d = {}
```

```
for url in urls:
    words = url.split(".")
    domain = "." + words[-1]
    if domain not in d:
        d[domain] = 1
    else:
        d[domain] += 1
```

```
print(d)
{'com': 2, 'in': 3, 'org': 1}
```

48.modify the "bal" from 100 to 1000, using for loop

```
info = {"name": "sunil", "bank": {"branch": "banglore", "bal": 100, "ifsc": "sbi420"}}
# without using loop
info["bank"]["bal"] = 1000
print(info)
#using for loop
info = {"name": "sunil", "bank": {"branch": "banglore", "bal": 100, "ifsc": "sbi420"}}
for key, value in info.items():
    if key == "bank":
        for key1, value1 in value.items():
            if key1 == "bal":
                value[key1] = 1000
print(info)
```

49. wp to create a dictionary 1st character as "key" and names starts with same 1st character should be the value

```
names = ["anu", "dinesh", "abhi", "divya", "suresh", "dileep", "mahesh"]
```

```
{"a":["anu", "abhi"]}
```

```
d = { }
```

```
for name in names:
```

```
    if name[0] not in d:
```

```
        d[name[0]] = [name]
```

```
    else:
```

```
        d[name[0]].append(name)
```

```
print(d)
```

50. wp to create a dictionary length as key and same length as list of value

```
fruits = ["apple", "grapes", "kiwi", "mango", "papaya", "bannana", "cherry", "berry"]
```

```
d = { }
```

```
for fruit in fruits:
```

```
    if len(fruit) not in d:
```

```
        d[len(fruit)] = [fruit]
```

```
    else:
```

```
        d[len(fruit)].append(fruit)
```

```
print(d)
```

51. wp to create a dictionary, with list of even and odd number separately (initially we should only add key and empty list)

```
numbers = [22, 34, 89, 10, 2, 4, 9, 7, 1, 56]
```

```
d = {"even": [], "odd": []}
```

```
for num in numbers:
```

```
    if num%2 == 0:
```

```
        d["even"].append(num)
```

```
    else:
```

```
        d["odd"].append(num)
```

```
print(d)
```

```
{'even': [22, 34, 10, 2, 4, 56], 'odd': [89, 9, 7, 1]}
```

52. word and length pair

```
sentence = "python is a programming language"
```

```
words = sentence.split()
```

```
word_len = { }
```

```
for word in words:
    word_len[word] = len(word)
# print(word_len)
```

53. index and element pair

```
sequence = "hello"
index_ele = { }

for index, char in enumerate(sequence):
    index_ele[index] = char

print(index_ele)
```

54. word and its count pair

```
sentence = "hai hello hai world hai good morning world hello hello"
words = sentence.split()
word_count = { }
```

```
# using inbuilt method
for word in set(words):
    word_count[word] = words.count(word)    # hello

print(word_count)
```

```
# without using inbuilt method
sentence = "hai hello hai world hai good morning world hello hello"
words = sentence.split()
word_count = { }
```

```
for word in words:
    if word not in word_count:
        word_count[word] = 1
    else:
        word_count[word] = word_count[word] + 1
```

55. character and count pair

```
s = "helloall"
d = { }
for rv in s:
    d[rv] = d.get(rv,0)+1
print(d)
o/p:{'h': 1, 'e': 1, 'l': 4, 'o': 1, 'a': 1}
```

56.create a dictionary with character and ascii value pair

```
s = "hello"
d = {}
for rv in s:
    d[rv] = d.get(rv,ord(rv))
print(d)
o/p:{'h': 104, 'e': 101, 'l': 108, 'o': 111}
```

(or)

```
s = "hello"
d = {}
for rv in s:
    if rv not in d:
        d[rv] = ord(rv)
print(d)
o/p:{'h': 104, 'e': 101, 'l': 108, 'o': 111}
```

57. create a dictionary with character and count pair, with only repeated character

```
s = "hello all... good morning... welcome all..."
d = {}
for rv in s:
    if s.count(rv) > 1:
        d[rv]= s.count(rv)
print(d)
o/p:{'e': 3, 'l': 7, 'o': 5, ' ': 5, 'a': 2, '!': 9, 'g': 2, 'm': 2, 'n': 2}
```

58. create a dictionary with word and its count pair

```
s = "hello all how are you hope you all doing well well well"
l = s.split()
print(l)
d = {}
for rv in l:
    if rv in d:
        d[rv]+= 1
    else:
        d[rv] = 1
    #d[rv] = l.count(rv)

print(d)
o/p:
['hello', 'all', 'how', 'are', 'you', 'hope', 'you', 'all', 'doing', 'well', 'well', 'well']
{'hello': 1, 'all': 2, 'how': 1, 'are': 1, 'you': 2, 'hope': 1, 'doing': 1, 'well': 3}
```

59.**break:** if list elements are greater than 100 then i will break

```
l = [10,20,30,300, 100, 40,50]
```

```
for rv in l:
```

```
    if rv > 100: #true means break false means continue to looping
        break
```

```
    else:
```

```
        print(rv)
```

60. **continue:**

```
s = "hello"
```

```
for rv in range(0,len(s)):
```

```
    if rv % 2 == 1: #true, skip and continue to next loop
        continue
```

```
    else:
```

```
        print(rv)
```

61. wap to print sum of numbers present inside a list

```
"""
```

```
#wap to print sum of first n numbers
```

```
c = 0
```

```
n = eval(input())
```

```
for r in range(0,n+1):
```

```
    c += r
```

```
print(c)
```

62. wap to display * in right angled triangle

```
n = int(input())
```

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

```
    for j in range(i+1):
```

```
        print("*", end = " ")
```

```
    print()
```

```
#or
```

```
n = 5
```

```
for i in range(1,n+1):
```

```
    print("*" * i)
```

63. Wap to check if the number is greater than 100 break the loop.

```
l = [10,20,30,300,10,50]
```

```
for r in l:
```

```
    if r > 100:
```

```
        break
```

```
    else:
```

```
        print(r)
```

```
else:
```

```
    print("hello")
```

64. wap to check the given number is prime or not:

```
n = eval(input())
for i in range(2,n):
    if n % i == 0:
        print("it is not a prime")
        break
else:
    print("it is a prime")
```

65.wap to print the nth prime number

```
n = eval(input())
num = 0
p_c = 0
while p_c < n:
    if num > 1:
        for i in range(2,num):          #(2,2)
            if num % i ==0:
                break
        else:
            p_c +=1
            if p_c == n:
                print(num)
    num +=1
```

66.

```
s1 = "hello"
s2 = "hai"
s3 = "brave"
s4 = "mind"
from itertools import zip_longest
zv = list(zip_longest(s1,s2,s3,s4))
```

```
for r1 in zv:
    if None not in r1:
        new = r1
        s5 = ""
        for r2 in new:
            s5 += r2
        print(s5)
```

67.

```
sum1 = 0
t = (1,2,3,4)
```

```
for rv in t:
    sum1 += rv
print(sum1)
```

68. wap to iterate through a string and access index value and corresponding character from a string using enumerate

```
str1= input()
for i in enumerate(str1):
    print(i,end=",")
```

69. wap to iterate through a string and access only index value using enumerate

```
str1= input()
for i in enumerate(str1):
    print(i[0],end=",")
```

70. wap to iterate through a string and access only values(characters) using enumerate

```
str1= input()
for i in enumerate(str1):
    print(i[1],end=",")
```

71.wap to iter through a string unpack the string in for loop.

```
str1 = input()
for i in enumerate(str1):
    print(i[0], i[1], end=",")
```

72. wap to unpack a string in loop definition

```
for i,e in enumerate(str1):
    print(i,e)
```

73.wap unpack the string in for loop definition itself and pring all characters

```
for i,e in enumerate(str1):
    print(e)
```

74.wap to check each character from the string is vowel or not by using enumerate:

```
str1 = input()
for i,e in enumerate(str1):
    if e in "aeiouAEIOU":
        print(e)
```

75.wap to check each word of the sentence are even length or not,if even prepare a list of them.

```
str1 = input()
l = str1.split()
l1= []
for i,e in enumerate(l):
    if len(e) % 2 == 0:
        l1.append(e)
```



```
print(l1)
```

76.

```
s = "hello"
print(list(reversed(s)))
l = [10,20,30,40,50]
print(list(reversed(l)))
t = (10,20,30,40,50)
print(tuple(reversed(t)))
```

77.

```
s1 = {10,20,30,40,50}
#print(set(reversed(s1)))
#as set type is unordered we cannot reverse it.
```

78.

```
d = {'a':100,'b':200,'c':300}
print(list(reversed(d)))
#only key layer is visible to interpreter so that only reversed.
```

79.

```
s = "hello"
s1 = "apple"
s2 = "orange"
print(list(zip(s,s1,s2)))
```

80. wap to create a new list of power numbers(one list elements should be base number and second list elements should be considered as their power).

```
l=[5,6,7,8,9]
l1=[4,3,2,1,0]
l_p = []
for e1,e2 in zip(l,l1):
    e3 = e1 ** e2
    l_p.append(e3)
print(l_p)
```

81. wap to find sum of cosecutive numbers of two different lists

```
l=[5,6,7,8,9]
l1=[4,3,2,1,0]
for e1,e2 in zip(l,l1):
    e3 = e1 + e2
    print(e3)
```

82. wap to create a list of numbers from 0 to 10 digit with words
["0:zero", "1:one", "2:two" "10:ten"]

```

l=["0","1","2","3","4","5","6","7","8","9","10"]
l1=["zero","one","two","three","four","five","six","seven","eight","nine","ten"]
l2=[]
for e1,e2 in zip(l,l1):
    e3 = e1+"."+e2
    l2.append(e3)
print(l2)

```

```

83.
l = [10,20,30,40,50]
l1 = ["pinki","rinki","chinki","dimpi"]

```

```

from itertools import zip_longest
print(list(zip_longest(l,l1)))

```

84. wap to sagregate 4 different strings and concatenate same indexed characters together using ziplongest if no character is none

```

s1 = input()
s2 = input()
s3 = input()
s4 = input()
from itertools import zip_longest
print(str(zip_longest(s1,s2,s3,s4)))

```

```

85.
s = "abracadadraca"
d = {}
for ch in s:
    d[ch] = s.count(ch)
print(d)

```

```

86.
s1 = "abracadadraca"
d1 = defaultdict(int)
for ch in s1:
    d1[ch] += 1 #it will take 0 and
print(d1)

```

```

87.
s2 = "abracadadraca"
d1 = {}
for ch in s2:
    if ch not in d1:
        d1[ch] = 1
    else:
        d1[ch] += 1

```

```
print(d1)
```

88.

```
s = "hello word uou"
d = {}
for ch in s:
    if ch in "aeiouAEIOU":
        d[ch] = s.count(ch)
print(d)
```

89.

```
s1 = "hello word uou"
d1 = defaultdict(int)
for ch in s:
    if ch in "aeiouAEIOU":
        d1[ch] += 1
print(d)
```

90.

```
s = "hello word uou"
d = {}
for ch in s:
    if ch in "aeiouAEIOU":
        if ch not in d:
            d[ch] = 1
        else:
            d[ch] += 1
print(d)
```

91.

```
a = [1, 2, 3, 4]
b = [5, 6, 7, 8]
s1 = []
for e1, e2 in zip(a, b):
    s1 += [e1+e2]
    #s1.append(e1 + e2)
print(s1)
```

92.

```
names = ["apple", "google", "apple", "yahoo", "facebook", "google"]
d = {}
for na in names:
    if names.count(na) > 1:
        d[na] = names.count(na)
print(d)
(or)
```

93.

```
names = ["apple", "google", "apple", "yahoo", "facebook", "google"]
#names = {"apple" : [0,2] , "google" : [1,5]}
d = { }
```

```
for index, ele in enumerate(names):
```

```
    if ele not in d:
```

```
        d[ele] = [index]
```

```
    else:
```

```
        d[ele].append(index)
```

```
print(d)
```

94.

```
names = ['steve', 'john', 'adam']
```

```
l = []
```

```
for cnt in names:
```

```
    l.append(len(cnt))
```

```
print(l)
```

```
l1 = [(con , len(con)) for con in names]
```

```
print(l1)
```

95.

```
s1 = "abcd"
```

```
l = []
```

```
for ind, ch in enumerate(s1):
```

```
    l.append((ind, ch))
```

```
print(l)
```

```
l1 = [(ind, ch) for ind, ch in enumerate(s1)]
```

```
print(l1)
```

96.

```
l = []
```

```
for no in range(1, 51):
```

```
    if no % 2 == 0:
```

```
        l.append(no)
```

```
print(l)
```

```
l1 = [ ]
```

97.

```
s = ["hello", "bye", "appl"]
```

```
s1 = []
```

```
for ch in s:
```

```
if ch[0] in "aeiouAEIOU":
    s1.append(ch)
print(s1)
```

(or)

```
s1 = [ ch for ch in s if ch[0] in "aeiouAEIOU"]
print(s1)
"""
```

98. wp to create a list of lang's starting with p

```
lang = ['python', 'sql', 'manual', 'perl']
l1 = []
for ch in lang:
    if ch[0].lower() == 'p':
        l1.append(ch)
print(l1)
```

(or)

```
l2 = [ ch for ch in lang if ch[0].lower() == 'p']
print(l2)
```

99. wp to create a list which starts with consonants

```
s = ["hello", "bye", "appl"]
s1 = []
for ch in s:
    if ch[0] not in "aeiouAEIOU":
        s1.append(ch)
print(s1)
(or)
s1 = [ ch for ch in s if ch[0] not in "aeiouAEIOU"]
print(s1)
```

100. wp build a list with only even length string

```
name = ['apple', 'mango']
```

```
rev = ["helllo", "world", "bye"]
l1 = []
for ch in rev:
    if len(ch) % 2 != 0:
        l1.append(ch[::-1])
```

```
    else:
        l1.append(ch)
print(l1)
```

101.

```
elnt = ["alex", 12, "13.1", 19, "90"]
e1 = []
for ch in elnt:
    if isinstance(ch, int):
        val = str(ch)[-1]
        e1.append(int(val))
    else:
        e1.append(ch)
print(e1)
```

102. word and its count pair

```
s = "mary had a little lamb lamb lamb"
words = s.split()
d = { }
for rv in words:
    if rv not in d:
        d[rv] = 1
    else:
        d[rv] += 1

print(d)
```

104. print only first 5 characters of the string

```
str1 = input()
if len(str1) >= 5:
    for rv in range(0, len(str1)):
        print(str1[rv])
        if rv == 4:
            break
else:
    print("insufficient characters")
```

105 .wap to traverse inside a string and print alternative characters.

```
str1 = input()
for rv in range(0, len(str1), 2):
    print(str1[rv], end = " ")
```

#i/p: hello all

#o/p: h l o a l

(OR)

```
str1 = input()
for rv in range(0,len(str1)):
    if rv % 2 == 0:
        print(str1[rv])
```

106.wap to fetch each character from a string and check whether they are vowels or not. if vowels print them.

```
str1 = input()
for rv in str1:
    if rv in "aeiouAEIOU":
        print(rv)
```

107 .wap to fetch each character from a string and check whether they are vowels or not. if vowels print them.

if not append them in a list

```
str1 = input()
s = ""
l=[]
for rv in str1:
    if rv in "aeiouAEIOU":
        s += rv          #s = s+rv,
        print()
    else:
        l.append(rv)
print(s)
print(l)
```

108. wap to fetch all the characters and remove the repated characters from the string.

```
str1 = input()
s = ""
for rv in str1:
    if rv not in s:
        s += rv
print(s)
```

109.

```
l1 = [1, 2, 3, 2, 1]
l2 = []
for no in l1:
    if no not in l2:
        l2.append(no)
print(l2)
```

110.

```
l1 = []
```

```
[l1.append(no) for no in l1 if no not in l1] #here we can't write and inbuilt method directly so  
we need a empty list
```

```
print(l1)
```

111.

```
l3 = list(dict.fromkeys(l1))
```

```
print(l3)
```

```
from itertools import zip_longest
```

112.

```
a = [1,2,3,4]
```

```
b = [1,2,3,4,5,8]
```

```
print(set(b) - set(a))
```

```
for ch in b:
```

```
    if ch not in a:
```

```
        print(ch, end=" ")
```

```
print()
```

```
for a1, b1 in zip_longest(a, b, fillvalue=0):
```

```
    if a1 == 0:
```

```
        print(b1, end=" ")
```

113.

```
a = ["abc", "123", "hello", "23"]
```

```
for val in a:
```

```
    if val.isdigit():
```

```
        print(val)
```

114.

```
a = "hello"
```

```
print(*a[::-2])
```

```
l = []
```

```
l.extend((a[::-2]))
```

```
print(l)
```

115.

```
names = ['apple', 'amazon', 'google', 'yah']
```

```
nn = []
```

```
for ch in names:
```

```
    if len(ch) % 2 == 0:
```

```
        nn.append(ch)
```



```
print(nn)
```

116.

```
l = [[1,2,3], [4,5,6], [7,8,9]]
res = 0
for le in range(len(l)):
    print(sum(l[le]))
    res = res + sum(l[le])
print(res)
```

117.

```
words = ["hi", "hello", "python"]
rev_wd = []
for ch in reversed(words):
    rev_wd.append(ch[::-1])
print(rev_wd)
```

```
rev_wd1 = []
for ch in range(-1,-len(words)-1,-1):
    rev_wd1.append(words[ch][::-1])
print(rev_wd1)
```

118.

1 - 50 even comprehension

```
l1 = [ n for n in range(51) if n % 2 == 0]
print(l1)
```

```
l1 = [ n for n in range(0,51,2) ]
print(l1)
```

119.

```
names = ["apple", "yahoo", "apple", "apple"]
for no in range(len(names)):
    if names.count(no) > 1:
        # for num in range(no+1,len(names)):
        #     if names[no] == names[no+1]:
        #         break
        print(names[no])
```

120.

```
st = "hello today is bad"
d = {}
for ch in st.split():
    d[ch] = len(ch)
print(d)
```

121.

```
d1 = { ch: len(ch) for ch in st.split() }  
print(d1)
```

```
from collections import defaultdict
```

122.

```
d = {'a':1, 'b':2}  
d1 = {}  
for ke in d:  
    d1[d[ke]] = ke  
print(d1)
```

123.

```
s = "helloe"  
d = {}  
for ch in s:  
    if s.count(ch) > 1:  
        d[ch] = s.count(ch)  
print(d)
```

124.

```
st = "today is tues day"  
s1 = {}  
for ind, word in enumerate(st.split()):  
    if ind % 2 == 0:  
        s1[ind] = word[::-1]  
    else:  
        s1[ind] = word  
print(s1)
```

125.

```
stmt = "hello world welcome to python hi"  
d = {}  
word = stmt.split()  
for ch in word:  
    if ch[0] not in d:  
        d[ch[0]] = [ch]  
    else:  
        d[ch[0]].append(ch)  
  
print(d)
```

126.

```
d1 = { ch[0] : [ch] if ch[0] not in {} else ch for ch in word }  
print(d1)
```

127.

```
d = {'a':'hello', 'b':100, 'c':10.1, 'd':'world'}
d1 = {}
for key, value in d.items():
    if isinstance(value, str):
        d1[key] = value[::-1]
    else:
        d1[key] = value
print(d1)
```

```
d2 = { key : value[::-1] if isinstance(value, str) else value for key, value in d.items() }
print(d2)
```

128.

```
s = "helloworld"
d1 = {}
for ch in s:
    if ch not in d1:
        d1[ch] = 1
    else:
        d1[ch] += 1
print(d1)
```

```
d3 = { ch: s.count(ch) for ch in s } #doubt
print(d3)
```

129.

```
s = "helloworld"
d1 = {}
for ch in s:
    if s.count(ch) > 1:
        d1[ch] = s.count(ch)
print(d1)
```

```
d2 = { ch : s.count(ch) for ch in s if s.count(ch) > 1 }
print(d2)
```

130.

```
d = {'a':100, 'b':{'m':'man', 'n':'nose', 'o':'ox', 'c':'cat'}}
for ch in d:
    if isinstance(d[ch], dict):
        for val in d[ch].values():
            if val == 'nose':
                d[ch] = 'net'

print(d)
```

131.

```
names = ['apple', 'mango', 'apple', 'juice']
d1 = {}
for ch in names:
    if ch not in d1:
        d1[ch] = 1
    else:
        d1[ch] += 1
print(d1)
```

132.

```
names = ['apple', 'mango', 'apple', 'juice']
d1 = {}
for ch in names:
    if names.count(ch) > 1:
        if ch not in d1:
            d1[ch] = 1
        else:
            d1[ch] += 1
print(d1)
```

133.

```
items = ['lotus-flower', 'lily-flower', 'cat-animal', 'sunflow-flower', 'dog-animal']
d1 = {}
for ch in items:
    for name, catr in [ch.split('-')]:
        if catr not in d1:
            d1[catr] = [name]
        else:
            d1[catr].append(name)
print(d1)
```

134.

```
files = ['apple.txt', 'yahoo.pdf', 'gmail.pdf', 'google.txt', 'amazon.pdf']
d = {}
for ch in files:
    for name, ext in [ch.split('.')]:
        if ext not in d:
            d[ext] = [name]
        else:
            d[ext].append(name)
print(d)
```

135.

```
numbers = [1,2,3,4,5,6,7,8,9,10]
```

```
evn = []
odd = []
d1 = {}
for no in numbers:
    if no % 2 == 0:
        evn.append(no)
    else:
        odd.append(no)
```

```
d1[0] = evn
d1[1] = odd
print(d1)
```

136.

```
name = ['apple', 'google', 'apple', 'yahoo', 'yahoo', 'google'] # {'apple': [0,2]}
d1 = {}
for ind, name in enumerate(name):
    if name not in d1:
        d1[name] = [ind]
    else:
        d1[name].append(ind)
print(d1)
```

FUNCTIONS :

1. fetch even numbers from the list.

```
l = [1,2,3,4,5,6,7,8,9,10]
def even(num):
    if num % 2 == 0:
        return num
print(list(map(even,l)))
```

(or)

```
l = [1,2,3,4,5,6,7,8,9,10]
even = map(lambda num : num%2 == 0, l)
print(list(even))
```

(or)

```
l = [1,2,3,4,5,6,7,8,9,10]
def even(num):
    if num % 2 == 0:
        return num
print(list(filter(even, l)))
```

(or)

```
l = [1,2,3,4,5,6,7,8,9,10]
even = lambda num: num%2==0
print(list(filter(even, l)))
```

2. wap to build a list of strings with even length from a list using filter

```
l = ["good", "boom", "spark","light","color","colour"]
even = lambda var : len(var)%2 == 0
print(list(filter(even, l)))
```

3. wap to build a list of strings which are starting from vowels

```
l = ["apple","banana","orange","pine","strawberry"]
vowel = lambda var : var[0] in "AEIOUaeiou"
print(list(filter(vowel,l)))
```

4. build a list of positive integers from the given list

```
l = [10,-67,100,55.5,-98.8,55,34,32.22]
p_i = lambda var : (isinstance(var, int) and var > 0)
print(list(filter(p_i, l)))
```

5.wap to build a list of prime numbers from 1 to 20 using filter

```
def prime_no(num):
    if num > 1:
        for i in range(2,num):
            if num % i == 0:
                break
        else:
            return True
print(list(filter(prime_no,range(2,20))))
```

6.

```
n = 20
num = 0
prime = 0
while prime < n:
    if num > 1:
        for i in range(2,num):
            if num % i == 0:
                break
        else:
            prime += 1
            if prime == n:
                print(num)
    num += 1
```

```

7.
num = int(input())
for i in range(2,num-1):
    if num % i == 0:
        break
else:
    return True

```

ASSIGNMENT :

ON FOR LOOP:

1. wap to create a dictionary with character and index value.
2. wap to sort the characters of the string (alphabets followed by digits)
3. wap to print a character repeatedly as many times as mentioned in consecutive digit (i/p: a4b3c2 o/p: aaaabbbcc)
4. wap to perform the following (i/p: a4k3b2 o/p: aeknbd)
5. wap to remove duplicate characters from the given string.
6. wap to find number of occurrences of each character present in the given string.
7. wap to display unique vowels present inside the given string.
8. wap to read a tuple from the user and print its sum and average.
9. wap to eliminate duplicates present in the list by reading list from user.
10. wap to read dictionary from user and find the sum of values if they are integer.
11. write a program to create a dictionary to get only the duplicate items and the number of times the item is repeated in the list. Names=["apple", "yahoo", "google", "apple", "yahoo", "facebook", "apple", "gmail", "gmail", "gmail"]
12. write a program to get the indices of each item in the below list.
Names=["apple", "google", "apple", "yahoo", "yahoo", "facebook", "apple", "gmail", "gmail", "gmail"]
o/p should be in dictionary with name and value list pair.
13. sort the list based on last character of each element.
names = ["steve", "eve", "adani", "bob", "birla", "ambani"]
14. sort the dictionary based on keys.
15. sort the dictionary based on length of the keys.
16. sort the dictionary based on first character of the keys
17. sort the dictionary based on last character of keys.
d = {"walmart": 7, "gmail": 5, "google": 6, "flipkart": 8}
18. sort the dictionary based on its values.
19. sort the elements from the given list. check if they are anagrams and group them into a dictionary
items = ["eat", "ate", "listen", "silent", "stressed", "tea", "desserts", "hai", "hello"]
20. wap to get indexes of each item from the list given.
names = ["apple", "google", "apple", "yahoo", "google", "gmail", "gmail"]
21. reverse the value of the dictionary if the value is of type string
d = {'a': "hello", 'b': 100, 'c': 10.1, 'd': "world"}

22. wap to get all the duplicates and the number of times the element is repeated in the list

```
names = ["apple", "google", "apple", "yahoo", "google", "gmail", "gmail", "gmail", "apple", "google", "apple", "yahoo", "google", "gmail"]
```

23. wap to create a dictionary taking keys and values using items in the list. take key as before special character and value as after special character in each element.

```
l = ["food@table", "lily@flower", "human@walk", "being@work"]  
o/p: d = {food:table, lily: flower}
```

24. wap to create a dictionary of index value and element pair of any iterable.

25.wap to count the number of characters in a string and create a dictionary with character and count pair

```
s = "good morning"  
d = {}  
for ch in s:  
    d[ch] = s.count(ch)  
print(d)
```

26.wap to create a dictionary with character and count pair if the character is vowel

27.wap to create a dictionary with character and count pair if the character is consonent

28.wap to create a dictionary with character and count pair if the character is digit

29.wap to create a dictionary with character and count pair if the character is spl characters

30. wp to print laptop name only if the price of the laptop is more than 30000 and it should be "i5" generation.

```
laptop = {"hp":["i5", 20000], "dell":["amd", 18000], "sony":["i5", 45000]}
```

31. wp print only boys name whose age is more than 18.

```
shadi = {"Sunil":["Male","Bang",19], "Prabhu":["Male","UK",28],  
"Dingi":["Female","Kashmir", 55], "Atul":["Male","Pune",15]}
```

32. wap to display * in pyramid(assignment)

33.wap to print odd numbers from 1 to 10 using continue keyword

34.wap to print the characters of string in both forward and reverse direction.

35.wap to display the positions of substring from the main string(pos of words in sentence)

36. wp list of names which are less than 6 character.

37. wap to count list elements and print only even numbers from the list

```
i/p: l = [5,10,15,17,88,45,66,73,98]
```

38. wap to print the elements if the length of list is even, else add one element and make it even.

39.wap to fetch last character of each word from the given sentence

```
sen = "india is my country all indians are my brothers and sisters"
```

40.wap to fetch all even character words from a sentence.

```
sen = "jony jony yes papa eating sugar no papa"
```

41.wap to check if the word is of odd length from given string and find mid character of each word.

```
sen= "twinkle twinkle little star how i wonder what you are ?"
```


