In [1]:

```
#Python program to reverse a number
#uisng while loop
n=int(input("enter the number:"))

rev=0
while n!=0:
    rev=rev*10+n%10
    n=n//10
print(rev)
```

```
enter the number:12345
54321
```

In [6]:

```
n=(input("enter the number:"))
sli=int(str(n)[::-1])
print(sli)
```

```
enter the number:123
321
```

In [ ]:

In [3]:

```
# Check Armstrong using while loop a number that is equal to the sum of cubes of its digi
ts
n=int(input("enter a number:"))
temp=n
size=len(str(n))
sum=0
while temp>0:
    digit=temp%10
    sum=sum+digit**size
    temp=temp//10
if sum==n:
    print("yes")
else:
    print("no")
```

```
enter a number:153
yes
```

In [ ]:

In [1]:

```
5%2
```

Out[1]:

```
1
```

In [36]:

```
#check primeee
num=int(input("enter a number:"))

if num>1:
    for i in range(2,num//2):
        if num%i==0:
            print("its not a prime")
            break
    else:
        print("its a prime")


else:
    print("number is not a prime")
```

```
enter a number:11
its a prime
```

In [ ]:

```

```

In [49]:

```
#fibonnacci series
num=int(input("enter the number:"))
first=0
second=1
for i in range(0,n):
    if i<=1:
        result=i
    else:
        result=first+second
        first=second
        second=result
    print(result)
```

```
enter the number:7
0
1
1
2
3
5
8
```

In [3]:

```
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

# Example usage:
n = 5  # Change this to the desired Fibonacci number you want to calculate
result = fibonacci(n)
print(f"The {n}-th Fibonacci number is {result}")
```

```
The 5-th Fibonacci number is 5
```

In [5]:

```
s="hello"
l=[]
print(len(s))
for i in range(len(s)):
    x=s[i]
    l.append(x)
print(l)
l=list(s)
print(l)
```

```
5
['h', 'e', 'l', 'l', 'o']
```

In [11]:

```
#Python program for palindrome number using iterative method
n=int(input("enter a integer :"))
temp=n
rev=0
while temp!=0:
    rev=rev*10+temp%10
    temp=temp//10
print(rev)
if rev==n:
    print("yes")
else:
    print("No")
```

```
enter a integer :121
121
yes
```

In [83]:

```
n1=int(input("enter the num:"))
n2=int(input("enter the num:"))
n3=int(input("enter the num:"))

if n1>n2 and n1>n3:
    print("n1 is greatest")
if n1<n2 and n3<n2:
    print("n2 is greatest")
if n1<n3 and n2<n3:
    print("n3 is greatest")
```

```
enter the num:3
enter the num:2
enter the num:1
n1 is greatest
```

In [12]:

```
#to check wheather a number is binary or not
num = int(input("please give a number : "))
while(num>0):
    j=num%10
    if j!=0 and j!=1:
        print("num is not binary")
        break
    num=num//10
if num==0:
    print("num is binary")
```

```
please give a number : 110001000
num is binary
```

In [95]:

```
#swapping two number
a=int(input("enter a number"))
b=int(input("enter number"))
a=a-b
b=a+b
a=b-a
print(a,b)
```

```
enter a number6
enter number4
4 6
```

In [96]:

```
#sum of total number in the list
n=int(input("enter number:"))
su=0
while n!=0:
    su=su+n%10
    n=n//10
print(su)
```

```
enter number:123
6
```

In [97]:

```
#swapping two number using third variable
a=int(input("enter a number"))
b=int(input("enter number"))
temp=a
a=b
b=temp
print(a,b)
```

```
enter a number1
enter number2
2 1
```

In [ ]:

```

```

In [9]:

```
210/2
```

Out[9]:

```
105.0
```

In [10]:

```
105/3
```

Out[10]:

```
35.0
```

In [14]:

```
#checks prime factors
n=int(input("enter a number to check prime factors:"))
i=2
while n>1:
    rem=n%i
    if rem==0:
        n=n/i
        print(i)
    else:
        i=i+1
```

```
enter a number to check prime factors:12
2
2
3
```

In [9]:

```
n=int(input("enter:"))
sum=0
for i in range(1,n):
    if n%i==0:
        sum=sum+i
    else:
        i=i+1
if(sum==n):
    print("its perfect")
else:
    print("not a perfect")
```

```
enter:6
```

```
its perfect
```

In [24]:

```
#factorial print
fact=1
n=int(input())
for i in range (1,n+1):
    fact=fact*i
print(fact)
```

```
5
120
```

In [ ]:

```

```

In [34]:

```
# Python Program to find first n prime numbers

num = int(input("Enter a number: "))
count = 0
n = 2

print("First", num, "prime numbers are: ")
while count < num:
    prime_flag = True

    for i in range(2, n//2 + 1):
        if (n % i) == 0:
            prime_flag = False
            break

    if prime_flag:
        print(n, end =" ")
        count = count + 1
    n = n + 1
```

```
Enter a number: 9
First 9 prime numbers are:
2 3 5 7 11 13 17 19 23
```

In [3]:

```
n=str(input("enter a charecter"))
if n=="a" or "e" or "i" or "o" or "u":
    print("its a vowel")
else:
    print("consonent")
```

```
enter a charecterz
its a vowel
```

In [14]:

```
def remove_(s1,s2):
    print(s1.replace(s2,"vasu"))

remove_("i am ok","ok")
```

```
i am vasu
```

In [12]:

```
def remove_char(s1,s2):
    print(s1.replace(s2, ''))
s1 = input("please give a String : ")
s2 = input("please give a Character to remove : ")
remove_char(s1,s2)
```

```
please give a String : yahooo this is vasu
please give a Character to remove : vasu
yahooo this is
```

In [16]:

```
def count_char(c):
    count=0
    for i in range(len(c)):
        if c[i]=="a":
            count=count+1
    return count


c="amabigaoaaausaaa"
count_char(c)
```

Out[16]:

```
9
```

In [17]:

```
s1="vasu"
s1="usav"
if sorted(s1)==sorted(s2):
    print("its anagram")
else:
    print("nope bro")
```

```
its anagram
```

In [19]:

```
def ispalindrome(s):
    for i in range(0,len(s)//2):
        if s[i]!=s[len(s)-i-1]:
            return False
    return True


s="vasu"
ispalindrome(s)
```

Out[19]:

```
False
```

In [24]:

```
ch="9"
if ch>="0" and ch<="9":
    print("its true")
else:
    print("false")
```

```
its true
```

In [25]:

```
ch=input("enter")
if ch.isdigit():
    print("yupp")
else:
    print("nope")
```

```
enter123
yupp
```

In [43]:

```
def concat(s1,s2):
    res=""
    for x in s1:
        if x==" ":
            x=s2
        res=res+x
    return res

s1="hemanth loves yuk"
s2="❤"
concat(s1,s2)
```

Out[43]:

```
'hemanth❤loves❤yuk'
```

In [40]:

```
def lcase_to_ucase(s):
    res=""
    for x in s:
        if x.islower():
            x=x.upper()
        else:
            pass
        res=res+x
    return res
s="hemanth"
lcase_to_ucase(s)
```

Out[40]:

```
'HEMANTH'
```

In [41]:

```
def lcase_to_ucase(s,vowels):
    for char in s:
        if char in vowels:
            u_char=char.upper()
            s=s.replace(char,u_char)
    return s


s="hemanth"
vowels="aeiou"
result = lcase_to_ucase(s, vowels)
print(result)   # Output should be "hEmAnth"
```

```
hEmAnth
```

vowels="aeiou" st="" for x in s: if x in vowels: x="" st=st+x st.replace(" ","") small="".join(st.split()) print(small) print(st)

In [14]:

```
s="hello aeiou iam the "
vowels="aeiou"
v_count=0
c_count=0
for x in s:
    if x in vowels:
        v_count=v_count+1
    elif x==" ":
        pass
    else:
        c_count=c_count+1
print(v_count)
print(c_count)
```

```
10
6
```

In [29]:

```
s="hello aeiou iam the "
dic={}
for char in s:
    if char==" ":
        pass
    elif char in dic:
        dic[char]=dic[char]+1
    else:
        dic[char]=1
max_freq=max(dic.values())
print("max_freq is:",max_freq)

for key in dic.keys():
    print(key,":",dic[key])
for char in dic:
    if dic[char]==max_freq:
        print(char,end=" ")
```

```
max_freq is: 3
h : 2
e : 3
l : 2
o : 2
a : 2
i : 2
u : 1
m : 1
t : 1
e
```

In [28]:

```
dictionary = {"a": 1, "b": 2, "c": 3}

# Printing keys
print("Keys:")
for key in dictionary.keys():
    print(key,":",dictionary[key])
```

```
Keys:
a : 1
b : 2
c : 3
Values:
1
2
3
```

In [23]:

```
st="Hello world!"
vow="aeiou"
count=0
for x in st:
    if count>1:
        break
    if x in vow:
        st=st.replace(x,"-")
    count=count+1
print(st)
```

```
H-llo world!
```

In [34]:

```
st="Hello world1234372929!!*&!!(*!(*!))!(!)!)*(!*(!*(!())))"

alpha_count=0
num_count=0
```

```
sp_count=0
for x in st:
    if x.isalpha():
        alpha_count=alpha_count+1
    elif x.isdigit():
        num_count=num_count+1
    else:
        sp_count=sp_count+1
print(alpha_count,num_count,sp_count)
```

```
10 10 35
```

In [42]:

```
st="hello world"
for char in st:
    print(char)
```

```
h
e
l
l
o
 
w
o
r
l
d
```

In [11]:

```
st="smksdk snskssi sdnskds sidnsids snsi nc d d d fn d"
st="".join(st.split())
print(st)
```

```
smksdksnskssisdnskdssidnsidssnsincdddfnd
```

In [47]:

```
def double(x):
    return x * 2
numbers = [1, 2, 3, 4, 5]
doubled_numbers = list(map(double, numbers))
print(doubled_numbers)
```

```
[2, 4, 6, 8, 10]
```

In [13]:

```
s1="Hello"
s2="World"
print("-".join([s1,s2]))
s3=s1+s2
print(s3)
```

```
Hello-World
HelloWorld
```

In [54]:

```
st="Mississippi"
new_st=""
for x in st:
    if x in new_st:
        continue
    else:
        new_st=new_st+x
print(new_st)
```

```
Misp
```

In [4]:

```
s="d1d1d1d1d1dd1d1d1d1d1dd1d"
a=0
for char in s:
    if char.isdigit():
        a=a+int(char)
    else:
        pass
print(a)
```

```
11
```

In [19]:

```
s="hello quescol"
freq_dic={}

for char in s:
    if char in freq_dic:
        freq_dic[char]=freq_dic[char]+1
    else:
        freq_dic[char]=1
ss=""
for char in s:
    if freq_dic[char]==1:
        ss=ss+char
    else:
        continue
print(ss)
```

```
h qusc
```

In [20]:

```
s="Heloo World!"
c=""
for char in s:
    c=c+char
print(c)
```

```
Heloo World!
```

In [6]:

```
s="wefkdfdefmeefenme"
sl=list(s)
print("".join(sorted(sl)))
print("".join(reversed(sorted(sl))))
```

```
ddeeeeeeffffkmmnw
wnmmkffffeeeeeedd
```

In [12]:

```
s="1213"
print(s[::-1])
```

```
3121
```

In [15]:

```
#finding the missing value from the given array
array=[1,2,3,5]
n=len(array)+1
s=n*(n+1)//2
arry_sum=sum(array)
m_val=s-arry_sum
print(m_val)
```

```
4.0
```

In [9]:

```
s = "pwwkew"
dic={}
for char in s:
    if char in dic:
        continue
    else:
        dic[char]=1
st=""
for keys,values in dic.items():
    st=st+keys
print(st)

max_count=0
for i in range(len(st)-1):
    for j in range(i+1,len(st)):
        if st[i:j] in s:
            max_count=max_count+1
```

```
pwke
```

In [12]:

```
set={1,2,3}
set.remove(2)
print(set)
```

```
{1, 3}
```

In [13]:

```
l=[1,3,4,5,6]
n=len(l)+1
s=sum(l)
s_n=n*(n+1)//2
r=s_n-s
print(r)
```

```
2
```

In [17]:

```
arr, occur = [], [];
n = int(input("please enter the size of array: "))
for x in range(n):
    occur.append(0)
print(occur)
for x in range(n):
    element = int(input(f"please enter the element of array element between 0 to {n-1} :"
))
    arr.append(element)
    occur[arr[x]]=occur[arr[x]]+1
for x in range(n):
    if occur[x]>1:
        print(f"{x} is repeated {occur[x]} times")
```

```
please enter the size of array: 3
[0, 0, 0]
please enter the element of array element between 0 to 2 :1
please enter the element of array element between 0 to 2 :1
please enter the element of array element between 0 to 2 :2
1 is repeated 2 times
```

In [22]:

```
lis=[]
#pair of sums equal or not
def pairs(arr,s):
    for i in range(len(arr)):
        for j in range(i+1,len(arr)):
            if arr[i]+arr[j]==s:
                lis.append((arr[i],arr[j]))


arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
s=12
pairs(arr,s)
print(lis)
```

```
[(2, 10), (3, 9), (4, 8), (5, 7)]
```

In [26]:

```
arr=[0,1,23,4,577,88,7,100]
minimum=float("inf")
maximum=float("-inf")
for num in arr:
    if num<minimum:
        minimum=num
    if num>maximum:
        maximum=num
print(maximum)
print(minimum)
```

```
577
0
```

In [1]:

```
-1%10
```

Out[1]:

```
9
```

In [3]:

```
import math
```

In [5]:

```
int(math.fmod(-1,10))
```

Out[5]:

```
-1
```

In [14]:

```
x=333.8
print(int(x//10))
```

```
33.0
```

In [4]:

```
n=int(input("enter the number of rows:"))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end=" ")#adds spaces
    print()#next line printing
```

```
enter the number of rows:4
1
1 2
1 2 3
1 2 3 4
```

In [6]:

```
n=int(input())#225#228.00
bill=0
if n<=50:
    bill=n*0.5
elif n>50 and n<=150:
    x=n-50
    bill=50*0.50+x*0.75
elif n>150 and n<=250:
    x=n-150
    bill=50*0.5+100*0.75+x*1.20
else:
    x=n-250
    bill=50*0.5+100*0.75+100*1.2+x*1.50
bill=bill+(20/100)*bill
bill=format(bill,".2f")
print(bill)
```

```
225
228.00
```

In [7]:

```
s = "dog cat cat dog"
print(s.split(" "))
```

```
['dog', 'cat', 'cat', 'dog']
```

In [5]:

```
s="abc"
print(s.split())
```

```
['abc']
```

In [8]:

```
pattern="hello"
print(list(pattern))
```

```
['h', 'e', 'l', 'l', 'o']
```

In [5]:

```
l1=[1,2,3]
l2=[1,2,3]
k=(map(l1,l2))
print(k)
```

```
<map object at 0x000002E9D0D75A80>
```

In [6]:

```
l1 = [1, 2, 3]
l2 = [1, 2, 3]

mapped_result = zip(l1, l2)

for element_from_l1, element_from_l2 in mapped_result:
    print("From l1:", element_from_l1)
    print("From l2:", element_from_l2)
```

```
From l1: 1
From l2: 1
From l1: 2
From l2: 2
From l1: 3
From l2: 3
```

In [4]:

```
l=[]
z=[1,2,3]
l.append(z)
print(l)
```

```
[[1, 2, 3]]
```

In [4]:

```
class MyClass:
    def __init__(self, x):
        self.x = x

    def my_method(self):
        print(self.x)

obj = MyClass(10)
obj.my_method()
```

```
10
```

In [ ]:

```

```

In [6]:

```
class MyClass:
    def __init__(self, x):
        self.x = x

    @classmethod
    def class_method(cls, x):
        return cls(x * 2)

    def instance_method(self):
        print(self.x)

obj1 = MyClass(5)
obj2 = MyClass.class_method(5)
obj1.instance_method()
obj2.instance_method()
```

```
5
10
```

In [8]:

```
def reverse_string(s):
    for i in range(len(s)-1,-1,-1):
        yield s[i]
s="python"
print(reverse_string(s))
print("".join(reverse_string(s)))
```

```
<generator object reverse_string at 0x000001C39720BAE0>
nohtyp
```

In [9]:

```
class rectangle:
    def __init__(self,width,height):
        self.__width=width
        self.__height=height
    def area(self):
        return self.__width*self.__height
cls=rectangle(4,6)
cls.area()
```

Out[9]:

```
24
```

In [10]:

```
l1=[1,2,3]
l2=[3,4,5]
```

```
result=[]
for x in l1:
    for y in l2:
        if x+y>5:
            k=x*y
            result.append(k)
print(result)
```

```
[5, 8, 10, 9, 12, 15]
```

In [13]:

```
l=[3,2,4,1]
for i in range(len(l)-1):
    for j in range(i+1,len(l)):
        if l[i]>l[j]:
            l[i],l[j]=l[j],l[i]
print(l)
print(l[1])
```

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

In [15]:

```
l=[1,2,3,4,5]
tot=0
for i in range(len(l)):
    if i%2==0:
        l[i]=-l[i]
    tot=tot+l[i]
print(tot)
```

```
-3
```

In [25]:

```
l=[1,2,3,4,5,6,7,8,9,10]
k=2
summ=0
for num in l:
    if num%2==0 and  num%k==0:
        summ=summ+num


print(summ)
```

```
30
```

In [27]:

```
s="abbaadcs"
stack=[]
i=0
count=0
for i in range(len(s)-1):
    if s[i] ==s[i+1]:
        count=count+1
        continue
    else:
        stack.append(s[i])
print(count)
```

```
2
```

In [28]:

```
s="abcdabc"
n=len(s)
prefix=[0]*n
print(prefix)
max_len=0
```

```
[0, 0, 0, 0, 0, 0, 0]
```

In [44]:

```
def calc_prefix_function(s):
    result = [0] * len(s)
    k = 0
    for i in range(1, len(s)):
#          while k > 0 and s[i] != s[k]:
#              k = result[k - 1]
        if s[k] == s[i]:
            k += 1
        result[i] = k
    return result
s="abcdefabcdef"
print(calc_prefix_function(s))
```

```
[0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6]
```

In [46]:

```
class parent:
    def __init__(self,x):
        self.x=x
class child(parent):
    def __init__(self,x,y):
        super().__init__(x)
        self.y=y
obj=child(10,20)
print(obj.x)
```

```
10
```

In [47]:

```
class my:
    att=42
o1=my()
o2=my()
o1.att=50
print(o1.att)
print(o2.att)
print(my.att)
```

```
50
42
42
```

In [49]:

```
def outer_function(x):
    def inner_function(y):
        return x+y
    return inner_function
result=outer_function(10)(20)
print(result)
```

```
30
```

In [50]:

```
l=[1,2,3,4]
a,b,c,d=map(lambda x:x*2,l)
print(a,b,c,d)
```

```
2 4 6 8
```

In [55]:

```
d={}
d["name"]="abc"
```

```
d["age"]="20"
d["address"]="none"
print(d)
print(sorted(d.items()))
```

```
{'name': 'abc', 'age': '20', 'address': 'none'}
[('address', 'none'), ('age', '20'), ('name', 'abc')]
```

In [56]:

```
l1=[1,2,3]
l2=l1
del l1
print(l2[2])
```

```
3
```

In [57]:

```
a=set([1,2,3])
b=set([2,3,4])
c=a.intersection(b)
print(c)
print(a.intersection(b))
c.issubset(a)
```

```
{2, 3}
{2, 3}
```

Out[57]:

```
True
```

In [59]:

```
a=frozenset([1,2,3])
b=frozenset([2,3,4])
a.union(b)
```

Out[59]:

```
frozenset({1, 2, 3, 4})
```

In [68]:

```
original_list = [1, 2, 3, 4]
shallow_copy = original_list

# Modify the shallow copy
shallow_copy[0] = 100

# Changes affect the original list
print(original_list)   # Output: [1, [200, 3], 4]
print(shallow_copy)
```

```
[100, 2, 3, 4]
[100, 2, 3, 4]
```

In [69]:

```
original_list = [1, 2, 3, 4]
deep_copy = original_list.copy()

# Modify the deep copy
deep_copy[0] = 100

# Changes do not affect the original list
print(original_list)   # Output: [1, [2, 3], 4]
print(deep_copy)
```

```
[1, 2, 3, 4]
[100, 2, 3, 4]
```

In [1]:

```
def creating_gen(index):
    months = ['jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov','dec']
    yield months[index]
    yield months[index+2]
next_month = creating_gen(3)
print(next(next_month), next(next_month))
```

```
apr jun
```

In [ ]:

```
#python -m pdb Python-script.py
```

In [3]:

```
import numpy as np
a1=np.array([])
a2=np.empty(shape=(3,3))
print(a1)
print(a2)
```

```
[]
[[4.67296746e-307 1.69121096e-306 1.69118787e-306]
 [1.89146896e-307 7.56571288e-307 3.11525958e-307]
 [1.24610723e-306 1.29061142e-306 2.56761491e-312]]
```

In [ ]:

```
#python contains unique  feature called as negative indexing negative indexing means start from the last
```

In [4]:

```
sum(range(1,102))
```

Out[4]:

```
5151
```

In [5]:

```
global_var=0
def modify_global():
    global global_var
    global_var=10
    print(global_var)
modify_global()
```

```
10
```

In [6]:

```
n = int(input("Enter number: "))
rev=0
while n>0:
    digit=n%10
    rev=rev*10+digit
    n=n//10
print(rev)
```

```
Enter number: 123
321
```

In [ ]:

```

```

In [ ]:

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```