In [ ]:

```
Task-1
input:10
expression1=10+(10^10)+9 9+(9^9)+8 8+(8^8)+7 7+(7^7)+6
              ?              ?         ?        ?
expression2=1+(1^1)+2  2+(2^2)+3  3+(3^3)+4 4+(4^4)+5
                ?       ?           ?         ?
output: expression1+expression2  expre1+expre2   .......10
                  ?                    ?
```

In [1]:

```
n=int(input())
temp=1
for i in range(n,0,-1):
    expre1=i+(i**i)+i-1
    expre2=temp+(temp**temp)+temp+1
    print(expre1+expre2,end=",")
```

```
10
10000000023,387420510,16777235,823560,46671,3138,267,36,11,6,
```

In [4]:

```
## prime Series
# input:7
# output:yes it is prime
n=int(input())
count=0
for i in range(1,n+1):
    if(n%i==0):
        count=count+1
if(count==2):
    print("Yes given number ",n,"is Prime")
else:
    print("Given Number ",n, "is Not a prime")
```

```
10
Given Number  10 is Not a prime
```

In [8]:

```python
# Series of PrimeNumbers b/w two Ranges
# input:1 10
# ouput:2 3 5 7
# 1 2 3 4 5 6 7 8 9 10
# 1-->1
# 2-->1,2
# 3-->1,3
# 4-->1,2,4
# 5-->1,5
# 6-->1,2,3,6
# 7-->1,7
# 8-->1,2,4,8
# 9-->1,3,9
# 10-->1,2,5,10
input1=int(input())
input2=int(input())
p_sum=0
for i in range(input1,input2+1):
    count=0
    for j in range(1,i+1):
        if(i%j==0):
            count=count+1
    if(count==2):
        print(i,end=" ")
        p_sum=p_sum+i
print("\nsum of Given Prime Numbers is:",p_sum)

```

```
1
10
2 3 5 7 sum of Given Prime Numbers is: 17
```

In [11]:

```python
# #Perfect Number
# 6-->1,2,3 -->6=6-->perfect
# 28-->1,2,4,7,14,-->28
# input:6
# output:yes it is perfect number
n=int(input())
pe_sum=0
for i in range(1,n):
    if(n%i==0):
        pe_sum=pe_sum+i
if(n==pe_sum):
    print("Yes Given Number is perfect")
else:
    print("Not a perfect")
```

```
10
Not a perfect
```

In [14]:

```
# input:1 500
# outpu:6,28,496 -->
n1=int(input())
n2=int(input())
for i in range(n1,n2+1):
    p_sum=0
    for j in range(1,i):
        if(i%j==0):
            p_sum=p_sum+j
    if(i==p_sum):
        print(i)
```

```
1
1000
6
28
496
```

In [ ]:

```
#while
# if (n=5):
#     break

while True:
    n=int(input())
    if n==5:
        break


```

print the total letter present in the number=34216324

In [3]:

```
number=int(input("Enter the number : "))
#7653
c=0
while number>0:#7653>0(T),765>0(T),76>0(T),7>0(T),0>0(F)
    number=number//10#-->7653//10->765,765//10-->76,76//10-->7,7//10-->0
    c=c+1#1,2,3,4
print("Total letters are :",c)

#      %-->reminder
#      //-->quotint
#     2345%10-->5
#     2345//10-->234
#     6%10-->6
#     6//10-->0

```

```
Enter the number : 2342
Total letters are : 4
```

In [6]:

```
# input:45682
# ouput:4+5+6+8+2 -->25
n=int(input())
d_sum=0
while(n!=0):#or n>0
    r=n%10#-->45682-->2
    d_sum=d_sum+r
    n=n//10#45682//10-->4568
print(d_sum)
```

```
1234
10
```

In [8]:

```
n=int(input())
d_pro=1
while(n>0):
    r=n%10
    d_pro=d_pro*r
    n=n//10
print(d_pro)
```

```
1234
24
```

In [ ]:

```
Task:2
input:98456
explanation:6*5 5*5 4*5 8*5 9*5
ouput:30 25 20 40 45
```

In [13]:

```
n=int(input())
temp=n
c=0
while(n>0):
    n=n//10
    c=c+1
print(c,temp)
while(temp>0):
    l=temp%10
    print(l*c,end=" ")
    temp=temp//10
```

```
098765434567987655676
20 98765434567987655676
120 140 120 100 100 120 140 160 180 140 120 100 80 60 80 100 120 140 160 180
```

In [ ]:

```
Task:3
input:6543
Explanation:3*4+1 4*3+2 5*2+3 6*1+4
ouput: 13  14  13  10
```

In [17]:

```
n=int(input())
c=0
temp=n
while(n>0):
    n=n//10
    c+=1
v=1
while(temp>0):
    l=temp%10
    print(l*c+v,end=" ")
    temp=temp//10
    c=c-1
    v=v+1

```

```
6543
13 14 13 10
```

In [ ]:

```
Task:4
input:
    6543
    9876
explanation:3*6 4*7 5*8 6*9
output:18 28 40 54
```

In [19]:

```
n1=int(input())
n2=int(input())
while n1!=0 and n2 !=0:
    r1=n1%10
    r2=n2%10
    print(r1*r2,end=" ")
    n1=n1//10
    n2=n2//10
```

```
1234
4321
4 6 6 4
```

In [ ]:

```
Task 5:
input:1234
      8753
Explanation:4+3*4 3+5*3 2*7+2 1+8*1
output:28 24 28 9


```

In [10]:

```
n1=int(input())
n2=int(input())
c=0
t=n1
while n1>0:
    n1=n1//10
    c=c+1
while (t>0 and n2>0):
    r1=t%10
    r2=n2%10
    print((r1+r2)*c,end=" ")
    t=t//10
    n2=n2//10
    c=c-1

```

```
1234
1234
32 18 8 2
```

Strings in python

- set of char set
- The Python string data type is a sequence of one or more individual characters that could consist of letters, numbers, whitespace characters, or symbols.
- We access strings through indexing, slicing them through their character sequences, and go over some counting and character location methods.
- How Strings are Indexed?Each of a string’s characters correspond to an index number, starting with the index number 0. In this spaces also be indexed.
- We have to access both positive index and negitive index +tv-->starts with zero or 0 -tv-->starts with -1b

In [11]:

```
s="cbit"#string Declaration
s

```

Out[11]:

```
'cbit'
```

In [26]:

```
print("first index ",s[1])
print("3rd index of given string",s[3])
#print("5th char in give n str",s[5])
print("2nd negitive index value ",s[-2])
print("except first chr",s[1:4])
print("except first chr",s[1:4])
print("first letter cap",s.capitalize())

```

```
first index  b
3rd index of given string t
2nd negitive index value  i
except first chr bit
except first chr bit
first letter cap Cbit
```

In [13]:

```
#dir
print(dir(str),end=" ")
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__siz
eof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'cou
nt', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'inde
x', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'i
slower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rind
ex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startsw
ith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

In [40]:

```python
#input->cbit
# c
# b
# i
# t
st='cbitapssdc'
c=0
for letter in st:
    c=c+1
    print(letter)
print("total count is:",c)

```

```
c
b
i
t
a
p
s
s
d
c
total count is: 10
```

In [28]:

```python
s1="apssdc"
s1.count("d")
```

Out[28]:

```
1
```

In [29]:

```python
s1.islower()
```

Out[29]:

```
True
```

In [30]:

```python
s1.isupper()
```

Out[30]:

```
False
```

In [31]:

```python
s.isupper()
```

Out[31]:

```
False
```

In [32]:

```python
ns="Hello Apssdc"
ns.swapcase()
```

Out[32]:

```
'hELLO aPSSDC'
```

In [33]:

```python
print(s)
print(s1)
print(ns)
```

```
cbit
apssdc
Hello Apssdc
```

In [34]:
```
s[::-1]#string reverse
```

Out[34]:
```
'tibc'
```

In [41]:
```
st
```

Out[41]:
```
'cbitapssdc'
```

In [42]:
```
len(st)
```

Out[42]:
```
10
```

In [45]:
```
t=input("Enter the string..")
t1=t[::-1]
if t==t1:
    print("Palidrom")
else:
    print("Not a palindrom")
```

```
Enter the string..malayalam
Palidrom
```

In [47]:
```
#"python programming"
p="python programming"
p1=len(p)//2
p1
```

Out[47]:
```
9
```

In [48]:
```
p[9]
```

Out[48]:
```
'o'
```

In [49]:
```
m=" muni "
m.strip()

```

Out[49]:
```
'muni'
```

In [50]:
```
m.lstrip()
```

Out[50]:
```
'muni '
```

In [51]:
```
m.rstrip()
```

Out[51]:
```
' muni'
```

In [57]:
```
sd="yeruiyueyuiyy364736576 %#$#$%"
sd.isalpha()
```

Out[57]:
```
False
```

In [53]:

```python
sd.isnumeric()
```

Out[53]:

```
False
```

In [61]:

```python
#print(sd.isalnum())
sd.isspace()
```

Out[61]:

```
False
```

In [65]:

```python
s="hello-world"
s4=s.split("-")
s4
```

Out[65]:

```
['hello', 'world']
```

In [68]:

```python
s5="@".join(s4)
s5
```

Out[68]:

```
'hello@world'
```

In [70]:

```python
data="This is python programming workshop"
l=len(data)
middle=l//2
middle
```

Out[70]:

```
17
```

In [72]:

```python
r_Data=data[middle:]
r_Data
```

Out[72]:

```
'ogramming workshop'
```

In [75]:

```python
l_data=data[:middle]
l_data
```

Out[75]:

```
'This is python pr'
```

In [78]:

```
# Frequency of each character given by string
for ch in data:
    print(ch,"-->",data.count(ch))
```

```
T --> 1
h --> 3
i --> 3
s --> 3
  --> 4
i --> 3
s --> 3
  --> 4
p --> 3
y --> 1
t --> 1
h --> 3
o --> 4
n --> 2
  --> 4
p --> 3
r --> 3
o --> 4
g --> 2
r --> 3
a --> 1
m --> 2
m --> 2
i --> 3
n --> 2
g --> 2
  --> 4
w --> 1
o --> 4
r --> 3
k --> 1
s --> 3
h --> 3
o --> 4
p --> 3
```

In [79]:

```
data
```

Out[79]:

```
'This is python programming workshop'
```

In [83]:

```
# uniq Data collecting
u_data=""
for ch in data:
    if (ch not in u_data):
        u_data=u_data+ch
u_data
```

Out[83]:

```
'This pytonrgamwk'
```

In [84]:

```
print(data)
print(u_data)
```

```
This is python programming workshop
This pytonrgamwk
```

In [87]:

```
for ch in u_data:
        print(ch,"--->",data.count(ch))

```

```
T ---> 1
h ---> 3
i ---> 3
s ---> 3
  ---> 4
p ---> 3
y ---> 1
t ---> 1
o ---> 4
n ---> 2
r ---> 3
g ---> 2
a ---> 1
m ---> 2
w ---> 1
k ---> 1
```

In [90]:

```
data1="python"
print(data1*5)
```

```
pythonpythonpythonpythonpython
```

In [ ]:

```
# input:"AAAABBBCCCCCDDDDDD"
# ouput:A4B3C5D6
```

In [94]:

```
data=input()
u_data=""
for ch in data:
    if ch not in u_data:
        u_data=u_data+ch
o_data=""
for ch in u_data:
    c=data.count(ch)
    o_data=o_data+(ch+str(c))
o_data
```

```
AAAABBBCCCCCDDDDDD
```

Out[94]:

```
'A4B3C5D6'
```

In [105]:

```python
# input:"ECE3CSE4EEE5CIV2"
# ouput;
#     ECEECEECE
#     CSECSECSECSE
#     EEEEEEEEEEEEEEE
#     CIVCIV
# data=input()
# s_data=""
# i_data=""
# for ch in data:
#     if(ch.isalpha()):
#         s_data=s_data+ch
#     if(ch.isnumeric()):
#         i_data=i_data+ch
# print(s_data)
# print(i_data)

# for ch in i_data:
#     for i in range(3,len(s_data)+1,3):
#         print(int(ch)*s_data[:i])
data=input()
s_data=""
for i in range(len(data)):
    if(data[i].isalpha()):
        s_data=s_data+data[i]
    if(data[i].isnumeric()):
        num=int(data[i])
        print(s_data*num)
        s_data=""
# or
data=input()
s_data=""
for ch in data:
    if(ch.isalpha()):
        s_data=s_data+ch
    if(ch.isnumeric()):
        num=int(ch)
        print(s_data*num)
        s_data=""
```

```
ECE3CSE4EEE5CIV2
ECEECEECE
CSECSECSECSE
EEEEEEEEEEEEEEE
CIVCIV
```

In [97]:

```
print(dir(str))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__siz
eof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'cou
nt', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'inde
x', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'i
slower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rind
ex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startsw
ith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

In [109]:

```
# input:data="ECEITCSECIV"
# output:
#          ECETIESCVIC
# input:CBITVBITPDTRCADP
# ouput:TIBCTIBVRTDPPDAC

```

In [119]:

```
chr(65)
```

Out[119]:

```
'A'
```

In [120]:

```
for i in range(0,129):
    print(i,".-->",chr(i))
```

```
0 .-->
1 .--> □
2 .--> □
3 .--> □
4 .--> □
5 .--> □
6 .--> □
7 .--> □
8 .-->
9 .-->
10 .-->

11 .--> □
12 .-->
13 .-->
14 .--> □
15 .--> □
16 .--> □
17 .--> □
18 .--> □
```

In [125]:

```
# input:data="vijayKUMAR10022@@#$%"
# output:total charaters=20
#        alphabets=10
#        numariccharachets=5
#        special charaters=5
data=input()
t_char=0
a_char=0
n_char=0
s_char=0
# for ch in data:
#     t_char+=1
#     if(ch.isalpha()):
#         a_char+=1
#     elif(ch.isnumeric()):
#         n_char+=1
#     else:
#         s_char+=1
# print(t_char,a_char,n_char,s_char)
for i in range(len(data)):
    t_char+=1
    if((ord(data[i])>=65 and ord(data[i])<=90) or (ord(data[i])>=97 and ord(
        a_char+=1
    elif(ord(data[i])>=48 and ord(data[i])<=57):
        n_char+=1
    else:
        s_char+=1
print(t_char,a_char,n_char,s_char)
```

```
vijayKUMAR10022@@#$%
20 10 5 5
```

In [ ]: