In [ ]:
```
# While
#     while cond:
#         ..stms
while True:
    continue
    n=int(input())
    if n==4:
        print("Exit")
        break
```

In [3]:
```
number=int(input())
c=0
while number>0:
    number=number//10 #%-->remin
    c=c+1
print(c)
```

```
48484
5
```

```
## Functions in python
* Inbulit Funtions
* User defiend Function
```

In [ ]:
```
# User defiend Function
 1. A function with arg[]  and with return Value
 2.A function with arg[]  and with out return Value
 3.A function with out arg[]  and with return Value
 4.A function with out arg[]  and with out return Value
```

In [ ]:
```
# Syntax of function definaton
def function_name():
    .....stms
function_name()
```

In [9]:
```
 #1. A function with arg[]  and with return Value
def isprime(n):
    if n<2:
        return False
    for i in range(2,n//2+1):
        if n%i==0:
            return False
    return True
n=int(input("Enter the Number "))
isprime(n)
```

```
Enter the Number 23
```

Out[9]:
```
True
```

In [10]:
```
# 1,100
for i in range(1,101):
    if isprime(i):
        print(i,end=" ")
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

In [11]:
```
#2.A function with arg[]  and with out return Value
def add(a,b):
    print(a+b)
add(2,3)
```

```
5
```

In [ ]:
```
Task:
    1. Funtion to generate 2000 to 2020 range  the all yeapp years
    2.Find the given number is perfect number or not
        input -->6:1,2,3,6
            range(1,n)
            n=sum of factors
            -->28
```

In [4]:
```
#1. Funtion to generate 2000 to 2020 range  the all yeapp years
def isLeap(year):
    if (year%400==0 or (year%100!=0 and year%4==0)):
        return True
year=int(input())
```

```
2020
```

Out[4]:
```
True
```

In [7]:
```
lb,ub=2000,2020
for i in range(lb,ub+1):
    if isLeap(i):
        print(i)

```

```
2000
2004
2008
2012
2016
2020
```

In [10]:
```
#Perfect Number
# 6->1,2,3
# 6=1+2+3
# 6=6

a=int(input("enter the number"))
b=0
for c in range (1,a):
    if a%c==0:
        b=b+c
if a==b:
    print("perfect number")
else:
    print("non perfect number")
```

```
enter the number4
non perfect number
```

Strings

- it is group of chr
- defined by str or ""

In [11]:
```
s = "welcome to cbit"
```

In [12]:
```
s
```

Out[12]:
```
'welcome to cbit'
```

In [13]:
```
type(s)
```

Out[13]:
```
str
```

In [14]:
```
s[0]
```

Out[14]:
```
'w'
```

In [15]:
```
1 s[5]
```

Out[15]: 'm'

In [16]:
```
1 s[14]
```

Out[16]: 't'

In [17]:
```
1 s[-1]
```

Out[17]: 't'

In [18]:
```
1 s[-2]
```

Out[18]: 'i'

In [20]:
```
1 s
```

Out[20]: 'welcome to cbit'

In [21]:
```
1 s[8:10]
```

Out[21]: 'to'

In [22]:
```
1 m = "haiwelcometocbit"
```

In [23]:
```
1 m
```

Out[23]: 'haiwelcometocbit'

In [24]:
```
1 m[3:10]
```

Out[24]: 'welcome'

In [25]:
```
1 m[0:3:2]
```

Out[25]: 'hi'

In [26]:
```
1 m[:-3]
```

Out[26]: 'haiwelcometoc'

In [27]:
```
1 m[::-1]
```

Out[27]: 'tibcotemoclewiah'

In [30]:
```
n = "malayalam"
s = n[::-1]
if n==s:
    print("palindrome")
else:
    print("not")

```

```
palindrome
```

In [31]:
```
dir(m)
```

Out[31]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
```

In [32]:
```
s = "welcome to cbit"
s
```

Out[32]:
```
'welcome to cbit'
```

In [33]:
```
s.capitalize()
```

Out[33]:
```
'Welcome to cbit'
```

In [34]:
```
s.title()
```

Out[34]:
```
'Welcome To Cbit'
```

In [35]:
```
s.casefold()
```

Out[35]:
```
'welcome to cbit'
```

In [36]:
```
m = "HAI WelCome"
```

In [37]:
```
1 m.casefold()
```

Out[37]: 'hai welcome'

In [38]:
```
1 m
```

Out[38]: 'HAI WelCome'

In [39]:
```
1 m.upper()
```

Out[39]: 'HAI WELCOME'

In [40]:
```
1 m.lower()
```

Out[40]: 'hai welcome'

In [41]:
```
1 dir(str)
```

Out[41]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
```

```
 'istitle',
 'isupper',
 'join',
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [42]:
```
1 s
```

Out[42]: 'welcome to cbit'

In [44]:
```
1 s.count("o")
```

Out[44]: 2

In [46]:
```
1 s.count("w")
```

Out[46]: 1

In [47]:
```
1 s
```

Out[47]: 'welcome to cbit'

In [48]:
```
1 s.encode()
```

Out[48]: b'welcome to cbit'

In [49]:
```
1 s
```

Out[49]: 'welcome to cbit'

In [50]:
```
1 g = s.encode()
```

In [51]:
```
1 g
```

Out[51]: b'welcome to cbit'

In [52]:
```
1 g.decode()
```

Out[52]: 'welcome to cbit'

In [53]:
```
1 g
```

Out[53]: b'welcome to cbit'

In [54]:
```
1 s
```

Out[54]: 'welcome to cbit'

In [55]:
```
1 s.endswith("c")
```

Out[55]: False

In [56]:
```
1 s.endswith("t")
```

Out[56]: True

In [57]:
```
1 s.startswith("e")
```

Out[57]: False

In [58]:
```
1 s.startswith("w")
```

Out[58]: True

In [59]:
```
1 s.find('l')
```

Out[59]: 2

In [60]:
```
1 s
```

Out[60]: 'welcome to cbit'

In [61]:
```
1 s.isupper()
```

Out[61]: False

In [62]:
```
1 s.islower()
```

Out[62]: True

In [63]:
```
1 'HAI'.isupper()
```

Out[63]: True

In [64]:
```
1 'HaI'.isupper()
```

Out[64]: False

In [65]:
```
1 'hAi'.islower()
```

Out[65]: False

In [66]:
```
1 s1 = "     hai welcome to cbit     "
```

In [67]:
```
1 s1
```

Out[67]: '   hai welcome to cbit   '

In [69]:
```
1 s1.strip()
```

Out[69]: 'hai welcome to cbit'

In [70]:
```
1 s1.rstrip()
```

Out[70]: '   hai welcome to cbit'

In [71]:
```
1 s1.lstrip()
```

Out[71]: 'hai welcome to cbit   '

In [72]:
```
1 s
```

Out[72]: 'welcome to cbit'

In [73]:
```
1 s.split()
```

Out[73]: ['welcome', 'to', 'cbit']

In [74]:
```
1 s.split('l')
```

Out[74]: ['we', 'come to cbit']

In [75]:
```
for i in s:
    print(i)
```

```
w
e
l
c
o
m
e

t
o

c
b
i
t
```

In [76]:
```
s= "HkhjkshkKBHJKGJGkhkgshskKJGHGGKUHukJG"
```

In [77]:
```
s
```

Out[77]: 'HkhjkshkKBHJKGJGkhkgshskKJGHGGKUHukJG'

In [78]:
```
s.swapcase()
```

Out[78]: 'hKHJKSHKkbhjkgjgKHKGSHSKkjghggkuhUKjg'

In [81]:
```
s3 = "ABCvsh123"
s3.isupper()
s3.islower()
s3.isalnum()
```

Out[81]: True

In [82]:
```
s
```

Out[82]: 'HkhjkshkKBHJKGJGkhkgshskKJGHGGKUHukJG'

In [83]:
```
s.isalnum()
```

Out[83]: True

In [84]:
```
"A#b2".isalnum()
```

Out[84]: False

In [85]:
```
1 'ajhdh'.isalpha()
```

Out[85]: True

In [86]:
```
1 '124anbABC'.isalpha()
```

Out[86]: False

In [87]:
```
1 s
```

Out[87]: 'HkhjkshkKBHJKGJGkhkgshskKJGHGGKUHukJG'

In [88]:
```
1 s3
```

Out[88]: 'ABCvsh123'

In [89]:
```
1 m
```

Out[89]: 'HAI WelCome'

In [ ]:
```
1  Task1:-
2  # s1 = "AbCD12!@#$"
3  # output: Capital letters : 3
4  #           Small letters : 1
5  #        special chr : 4
6  #        Numbers : 2
7
8  Task2:-
9
10     S2 = "kalyani praveena prathuyusha charan"
11
12     output :-
13         praveena
14         prathuyusha
15
16 Task3 :-
17
18     s3 = "wer34dfg3rtdfg673"
19     output: 26
20
21 task4:-
22
23     s4 = "banana 33 apple 83 grapes 63"
24
25     output :-179
26
27
```