

In [2]:

```
1  # 2.check given number is prime number or not
2  #      ex:2,3,5,11,13,7,17...
3  n=int(input())
4  fact=0
5  for i in range(1,n+1):
6      if n%i==0:
7          fact=fact+1
8  if fact==2:
9      print("given number is prime ")
10 else:
11     print("not a prime ")
12
13
```

6  
not a prime

In [6]:

```
1  # 3.check the given number is perfect number or not
2  #      ex:input n:6
3  #      1,2,3,6
4  #      1+2+3=6
5  #      sum of the factors = n
6  n=int(input())
7  f=0
8  for i in range(1,n):
9      if n%i==0:
10         f=f+i
11 if f==n:
12     print("perfect")
13 else:
14     print("not a perfect")
```

28  
perfect

In [ ]:

```
1  #1.Check the give number is factor of 1000 between 1 to 10
```

In [12]:

```
1  #Sum of first 10 natural numbers
2  sum=0
3  c=0
4  for number in range(1,11):
5      sum=sum+number
6      c+=1
7  print("Sum is :",sum)
8  print("avg is :",sum//c)
```

Sum is : 55  
avg is : 5

```
In [2]: 1 #while loop
2 # while cond:
3 #     ..stms
4 n=int(input())
5 c=0
6 while n>0:
7     n=n//10
8     c=c+1
9 print(c)
10
11
```

```
987
3
```

```
In [15]: 1 12345678//10
```

```
Out[15]: 1234567
```

```
In [2]: 1 n1=8
2 while True:
3     n2=int(input())
4     if n2==n1:
5         print("hello bye..")
6         break
```

```
098
08
hello bye..
```

```
In [3]: 1 4/2
```

```
Out[3]: 2.0
```

```
In [ ]: 1 #Functions in python
2 * A function is a collection of Statements
3 * Code Re-Usability
4 * Type of the funtions
5     - User Defiend fun()
6         * four types:
7             1.A function with arg[] and with return value
8             2.A function with out arg[] and with return value
9             3.A function with arg[] and with out return value
10            4.A function with out arg[] and with out return value
11
12     - predefiened fun()
13
```

```
In [5]: 1 #1.A function with arg[] and with return value
2 # def funtion_name(arg1[],arg2[]):
3 #     return ..stms
4 # funtion_name(arg1[],arg2[])
5 def addition(a,b):
6     return a+b
7 a=int(input())
8 b=int(input())
9 addition(a,b)
```

78

5

Out[5]: 83

```
In [8]: 1 #check give number is prime or not
2 def isPrime(n):
3     if n<2:
4         return False
5     for i in range(2,n//2+1):
6         if n%i==0:
7             return False
8     return True
9 n=int(input())
10 isPrime(n)
```

4

Out[8]: False

```
In [9]: 1 for prime in range(1,101):
2     if isPrime(prime):
3         print(prime,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 [ ]: 1 #task :
2 #generate the all leap years from 2000 to 2020
3
4
5
```

## Strings:

- Collection of data set
- Ex:chars,digits,spcl char,spce....etc
- Short form is str
- dir(str)

In [10]: 1 `print(dir(str))`

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattr__', '__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 [12]: 1 `s="hello cbit"`  
2 `s`

Out[12]: 'hello cbit'

In [13]: 1 `len(s)` *#length of the string*

Out[13]: 10

In [15]: 1 *#string indexing*  
2 `s[4]`

Out[15]: 'o'

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

Out[16]: 't'

In [18]: 1 `s[-4]`

Out[18]: 'c'

In [20]: 1 *#strings slicing*  
2 `s[0:5]`

Out[20]: 'hello'

In [22]: 1 `s[6:10]`

Out[22]: 'cbit'

In [ ]: 1 `b="we are in cbit "`

In [27]: 1 *# o/p:ello cbit*  
2 `s[1:]`

Out[27]: 'ello cbit'

```
In [28]: 1 s
```

```
Out[28]: 'hello cbit'
```

```
In [31]: 1 s[::-2]
```

```
Out[31]: 'hloci'
```

```
In [32]: 1 s[::-1]
```

```
Out[32]: 'tibc olleh'
```

```
In [34]: 1 s
```

```
Out[34]: 'hello cbit'
```

```
In [35]: 1 s.capitalize()
```

```
Out[35]: 'Hello cbit'
```

```
In [36]: 1 ns="Apssdc Python"
```

```
In [37]: 1 ns.swapcase()
```

```
Out[37]: 'aPSSDC pYTHON'
```

```
In [39]: 1 ns.count('s')
```

```
Out[39]: 2
```

```
In [ ]: 1 #Task :  
2 #1.print the all unique chars in given string  
3 #input:hello cbit and vbit  
4 #output:helocbandv  
5 #2.Print the last two chars in revers order  
6 #input:hello apssdc  
7 #output:cd  
8 #3.s="yourbadstudentsright100times"  
9 #out put :100  
10 #4.st="cse 60 ece 70 eee 50 "  
11 #output:sum the all digits :180  
12 #5.Check the given string is palidrome or not  
13 #6.Find the avarage char of the given string  
14 #input:s=pyhton  
15 output:h  
16 #7.Find the all prime numbers sum and avarage 1-100  
17  
18  
19  
20
```

