# Task:

# #Search the Contacts

# #Find the freqvency of the given string

# #Ex:number=12311114321115657111

# #ex:name="wearelaeringPythonProgrmmingggg

# output1:1:11

# g:9

# Task:

# input:{'g':4,'s':5,'a':3,'cbit':2}

# ouput:ggggsssssaaacbitcbit

In [13]:
```
s={'g':4,'s':5,'a':3,'cbit':2}
l1=list(s.keys())
```

In [14]:
```
l2=list(s.values())
```

In [15]:
```
type(l1)
```

Out[15]:
```
list
```

In [17]:
```
for i in range(len(l1)):
    print(l1[i]*l2[i],end="")

```

```
ggggsssssaaacbitcbit
```

In [18]:
```
for k,v in s.items():
    print(k*v,end="")
```

```
ggggsssssaaacbitcbit
```

In [30]:
```
#input : 10 20
#ouput:30
s="10 20 30 40 "
s=s.split()
# print(int(s[0])+int(s[1])+int(s[2])+int(s[3]))
s=list(map(int,s))
print(sum(s))
```

```
100
```

In [25]:
```
#input:1 2 3 4 5 6 7 8 9
#ouput:45
input_data=input()
input_data=input_data.split()
input_data=list(map(int,input_data))
sum(input_data)
```

```
1 2 3 4 5 6 7 8 9
```

Out[25]:
```
45
```

In [24]:
```
print(sum(list(map(int,input().split()))))
```

```
1 2 3 4 5 6 7 8 9
45
```

In [35]:
```
lenght=int(input())
test_cases=int(input())
for i in range(test_cases):
    w_h=list(map(int,input().split()))
    h=w_h[0]
    w=w_h[1]
    if(h<lenght or w<lenght):
        print("Upload Another Pic")
    else:
        if(lenght==h and lenght==w):
            print("Accepted")
        else:
            print("Crop it")

```

```
180
3
640 480
Crop it
120 300
Upload Another Pic
180 180
Accepted
```

# Python Packages And Modules

- Collection of or seqvence of functions called Modules
- set of Modules Called as Packages

- Ex:maths,re,keyword...etc
- We will use the import keyword for importing the packages

In [40]:
```
import math #all predefined packages
math.sqrt
```

Out[40]:
```
<function math.sqrt(x, /)>
```

In [36]:
```
from math import sqrt #only sqrt import
sqrt(2)
```

Out[36]:
```
1.4142135623730951
```

In [37]:
```
sqrt(3)
```

Out[37]:
```
1.7320508075688772
```

In [38]:
```
2**3
```

Out[38]:
```
8
```

In [41]:
```
print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acos
h', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cos
h', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floo
r', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfini
te', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'mod
f', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'tau', 'trunc']
```

In [42]:
```
print(math.pi)
```

```
3.141592653589793
```

In [43]:
```
math.sin(45)
```

Out[43]:
```
0.8509035245341184
```

In [52]:
```
import random as r

r.randint(1,10)
r.randrange(1,10)

for i in range(10):
    print(r.randint(1,10),end=" ")
```

```
9 10 9 1 2 5 4 7 1 10
```

In [57]:
```
str(chr(r.randrange(97,122)))
```

Out[57]:
```
'b'
```

In [62]:
```
#Randomly generating the char
w=''
for i in range(10):
    w=w+chr(r.randrange(97,122))
w
```

Out[62]: 'byogbuntod'

In [69]:
```
import mypkg

mypkg.isPrime(78)
```

Out[69]: False

In [70]:
```
lb,ub=1,1001
for prime in range(lb,ub):
    if mypkg.isPrime(prime):
        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 97 101 103
107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211
223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 313 317 331
337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449
457 461 463 467 479 487 491 499 503 509 521 523 541 547 557 563 569 571 577 587
593 599 601 607 613 617 619 631 641 643 647 653 659 661 673 677 683 691 701 709
719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 823 827 829 839 853
857 859 863 877 881 883 887 907 911 919 929 937 941 947 953 967 971 977 983 991
997
```

In [4]:
```
import myfile as m

m.isPrime(674)

m.iseven(8)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-4-8071593bf17e> in <module>
      3 m.isPrime(674)
      4
----> 5 m.iseven(8)

AttributeError: module 'myfile' has no attribute 'iseven'
```

In [5]:
```
def iseven(n):
    if(n%2==0):
        return True
def isodd(n):
    if(n%2!=0):
        return True
```

In [4]:
```

import myfile
```

In [6]:
```
myfile.isPrime(
```

Out[6]: False

In [7]:
```
import pck
```

In [8]:
```
pck.iseven(8)
```

Out[8]: True

In [9]:
```
pck.isodd(5)
```

Out[9]: True

In [10]:
```
pck.isPrime(5)
```

Out[10]: True

In [11]:
```
from pck import *
```

In [12]:
```
isPrime(8)
```

Out[12]: False

In [13]:
```
iseven(8)
```

Out[13]: True

In [14]:
```
isodd(7)
```

Out[14]: True

In [15]:
```
print(dir(pck))
```

```
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
'__package__', '__spec__', 'isPrime', 'iseven', 'isodd']
```

In [18]:
```
import apssdc.data  as cbit
```

In [17]:
```
apssdc.data.naturalnumbers(10)
```

```
0 1 2 3 4 5 6 7 8 9
```

In [19]:
```
cbit.naturalnumbers(20)
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

In [20]:
```
print(dir(cbit))
```

```
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
'__package__', '__spec__', 'isPrime', 'iseven', 'isodd', 'naturalnumbers']
```

In [21]:
```
import math
```

In [23]:
```
cbit.naturalnumbers?
```

In [24]:
```
math.pow?
```

## Regular Expressions

- It is Used For Matching the Some Patterns
- It Is is also one type Lang..
  - It is form Symbols
- Each and Every Char... have it own Indivisuval property
- Shortform is re
- We need import the re(Regular Expressions)

In [25]:
```
import re

print(dir(re))
```

```
['A', 'ASCII', 'DEBUG', 'DOTALL', 'I', 'IGNORECASE', 'L', 'LOCALE', 'M', 'MULTI
LINE', 'Match', 'Pattern', 'RegexFlag', 'S', 'Scanner', 'T', 'TEMPLATE', 'U',
'UNICODE', 'VERBOSE', 'X', '_MAXCACHE', '__all__', '__builtins__', '__cached_
_', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__',
'__version__', '_cache', '_compile', '_compile_repl', '_expand', '_locale', '_p
ickle', '_special_chars_map', '_subx', 'compile', 'copyreg', 'enum', 'error',
'escape', 'findall', 'finditer', 'fullmatch', 'functools', 'match', 'purge', 's
earch', 'split', 'sre_compile', 'sre_parse', 'sub', 'subn', 'template']
```

In [28]:
```
#Match
re.match("a","apssdc")
```

Out[28]:
```
<re.Match object; span=(0, 1), match='a'>
```

In [31]:
```
#Match :It will Match the First Char only
re.match("p","apssdc")
```

In [34]:
```
#Search:It is only search if at least one success occure
re.search('a','apssdcaaaa')
```

Out[34]:
```
<re.Match object; span=(0, 1), match='a'>
```

In [37]:
```
#Find All
re.findall('s','apssdcaaaaaaassssss')
```

Out[37]:
```
['s', 's', 's', 's', 's', 's', 's', 's']
```

## Regular Expression Symbols - Meanings

1. Char Set --->[]:lb,ub
2. Range set --->{ } :Limit ,
3. set () ------> [d][h][o][n][i] or (dhoni)
4. ^(cap)------->Exactly starting with
5. $(Dollor)---->Ending with
6. .(dot) ---->al least one char
7. *(star) ----->zero occarence
8. | xor -------> either one or two cond..

In [ ]:
```
#Match the the dhoni word

```

In [ ]:
```
#mOBILE nUMBER Validation
 # Starting with 6 or 7 or 8 or 9
        #^[6-9][0-9]{9}$
  #Starting with zero
        #^[0][6-9][0-9]{9}$
  #Staring with +91
        #^[+][9][1][6-9][0-9]{9}$
#Final Validation:
#^[+][9][1][6-9][0-9]{9}$|^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}$
```

In [42]:
```
mobilenUmbers=["132577476","9876543212","98760000001",\
               '235647254','862537553625',"+918765555552",'06999999929']
pattern="^[+][9][1][6-9][0-9]{9}$|^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}$"
def isvalidMobileNumber(number):
    if  (re.match(pattern,number)):
        return True
    return False

number=input()
isvalidMobileNumber(number)
```

```
1234590000
```

Out[42]:
```
False
```

In [43]:
```
for mobileNumber in mobilenUmbers:
    if isvalidMobileNumber(mobileNumber):
        print(mobileNumber,end=" ")
```

```
9876543212 +918765555552 06999999929
```

In [ ]:
```
#Email Validation
muni.apssdc@gmail.com
muneiahtellakula@gmail.com
muneiah.t@apssdc.in
muni_saho@yahoo.com
vijay10022@gmail.com
rgukt@rkvally.ac.in

#1.Username
    #8 to 18[]
2.domine name
    #3-7
3.Extention
    #2-3
# Final validation for email
#^[a-z][a-zA-Z._0-9]{8,18}[@][a-z]{3,8}[.][a-z]{2,3}$


```

In [46]:
```
mail=input()
emaiPattern="^[a-z][a-zA-Z._0-9]{8,18}[@][a-z]{3,8}[.][a-z]{2,3}$"
if (re.match(emaiPattern,mail)):
    print("Valid",mail)
else:
    print("Invalid :",mail)
```

```
565jkfkdfkh@gmail.com
Invalid : 565jkfkdfkh@gmail.com
```

In [ ]:
```

```