String split methods

split()	This method splits (OR) divides a string into sub string using separator. Search for separator from left to right Syntax: string-name.split(sep=' ',max_split=-1)
rsplit()	This method splitis (OR) divides a string into sub string using separator, search for separator from righ to left

Example:

```
>>> str1="a,b,c,d,e"
>>> A=str1.split(",")
>>> print(A)
['a', 'b', 'c', 'd', 'e']
>>> str2="a b c d e"
>>> B=str2.split()
>>> print(B)
['a', 'b', 'c', 'd', 'e']
>>> C=str2.split(maxsplit=2)
>>> print(C)
['a', 'b', 'c d e']
>>> str3="10 20 30 40 50"
>>> D=str3.split()
>>> print(D)
['10', '20', '30', '40', '50']
>>> E=input().split()
1 2 3 4 5
>>> print(E)
['1', '2', '3', '4', '5']
```

https://www.codechef.com/problems/GDTURN

```
t = int(input())
for i in range(0,t):
    x,y = map(int,input().split())
    # write your code here
    if x+y>6:
        print("yes")
    else:
        print("no")
```

What is packing and unpacking?

Packing is process of grouping individual values or elements inside one collection by assigning to single variable.
Unpacking is process of reading individual values or elements from collection and assigning to individual variables.

Example of packing:

```
>>> a=10,20,30,40,50
>>> print(a,type(a))
(10, 20, 30, 40, 50) <class 'tuple'>
>>> b="naresh","suresh","rajesh"
>>> print(b,type(b))
('naresh', 'suresh', 'rajesh') <class 'tuple'>
```

Example of unpacking

```
>>> x,y,z=(100,200,300)
>>> print(x,y,z)
100 200 300
>>> a,b,c,d,e=10,20,30,40,50
>>> print(a,b,c,d,e)
10 20 30 40 50
>>> t=(10,20,30,40,50)
>>> a,b,c,*d=t
>>> print(a,b,c,d)
10 20 30 [40, 50]
>>> a,*b=t
>>> print(a,b)
10 [20, 30, 40, 50]
>>> a,b,c,d,e,f="PYTHON"
>>> print(a,b,c,d,e,f)
P Y T H O N
>>> a,b,*c="PYTHON"
>>> print(a,b,c)
P Y ['T', 'H', 'O', 'N']
```

How to input multiple values in single line?

```
>>> a,b,c=input().split()
10 20 30
>>> print(a,b,c,type(a),type(b),type(c))
10 20 30 <class 'str'> <class 'str'> <class 'str'>
>>> x,y,z=map(int,input().split())
10 20 30
>>> print(x,y,z,type(x),type(y),type(z))
10 20 30 <class 'int'> <class 'int'> <class 'int'>
```

join()

This method is used to join group of strings or collection of string into one string using separator

Syntax:
Variable-name=separator.join(iterable/collection)

Note: separator is a string

```
>>> A=["10","20","30","40"]
>>> s1=",".join(A)
>>> print(A)
['10', '20', '30', '40']
>>> print(s1)
10,20,30,40
>>> B=["python","java","oracle"]
>>> s2=" ".join(B)
>>> print(B)
['python', 'java', 'oracle']
>>> print(s2)
python java oracle
>>> C=("naresh","ramesh","kishore")
>>> s3=":".join(C)
>>> print(C)
('naresh', 'ramesh', 'kishore')
>>> print(s3)
```

```
naresh:ramesh:kishore
```

Example:

```
# Reverse Words in a Given String in Python
# python java oracle
# nohtyp avaj elcaro

str1="python java oracle"
A=str1.split()
B=[]
for s in A:
    B.append(s[::-1])

str2=" ".join(B)
print(str1)
print(str2)
```

Output

```
python java oracle
nohtyp avaj elcaro
```

Example:

```
>>> str1="a,b,c,d,e"
>>> A=str1.split(",")
>>> print(A)
['a', 'b', 'c', 'd', 'e']
>>> B=str1.rsplit(",")
>>> print(B)
['a', 'b', 'c', 'd', 'e']
>>> C=str1.split(",",maxsplit=2)
>>> print(C)
['a', 'b', 'c,d,e']
>>> D=str1.rsplit(",",maxsplit=2)
>>> print(D)
['a,b,c', 'd', 'e']
```

Example:

```
# How to Remove Letters From a String in Python
# ith letter from string

str1=input("Enter any String ")
i=int(input("Enter index of letter "))
A=list(str1)
del A[i]
str2="".join(A)
print(str1)
print(str2)
```

Output

```
Enter any String python
Enter index of letter 0
python
ython
```

Alignment methods or justification method

ljust()	Align string left side within given width
rjust()	Align string right side within given width
center()	Align string center within given width

Example:

```
str1="nit"
str2=str1.ljust(10)
print(str1,len(str1))
print(str2,len(str2))
str3=str1.ljust(10,'*')
print(str3)
str4=str1.rjust(10)
print(str4)
str5=str1.rjust(10,"$")
print(str5)
str6=str1.center(10)
print(str6)
str7=str1.center(10,"*")
```

```
print(str7)
```

Output

```
nit 3
nit       10
nit*******
       nit
$$$$$$$nit
   nit
***nit****
```

Example:

```
student=[["naresh","python"],
         ["suresh","c"],
         ["ramesh","c++"],
         ["kishore","AI"]]

for stud in student:
    name,course=stud
    print(name.center(20,'*'),course.ljust(10,'*'))
```

Output

```
*******naresh******* python****
*******suresh******* c*********
*******ramesh******* c++*******
******kishore******* AI********
```

Strip methods

lstrip()	This method is used to remove leading characters from string and returns new string
rstrip()	This method is used to remove trialing characters from string and returns new string
strip()	This method is used to remove leading and trialing character from string and returns new string

Syntax of lstrip()
variable-name=string-name.lstrip(chars=' ')

default characters removed from string are spaces

```
>>> str1="     naresh"
>>> str2=str1.lstrip()
>>> print(len(str1),len(str2))
11 6
>>> print(str1,str2)
     naresh naresh
>>> str3="***nit"
>>> str4=str3.lstrip("*")
>>> print(str3,str4,sep="\n")
***nit
nit
>>> str5="n*i*t"
>>> str6=str5.lstrip('*')
>>> print(str5)
n*i*t
>>> print(str6)
n*i*t
>>> str7="**$$##**@@$$nit"
>>> str8=str7.lstrip("*$#@")
>>> print(str7,str8,sep="\n")
**$$##**@@$$nit
Nit
```

Syntax of rstrip()
variable-name=string-name.rstrip(chars=' ')

```
>>> str1="nit       "
>>> str2=str1.rstrip()
>>> print(len(str1),len(str2))
10 3
>>> str3="nit*******"
>>> str4=str3.rstrip("*")
>>> print(str3,str4,sep="\n")
nit*******
nit
>>> str5="nit***&&&^^^"
>>> str6=str5.rstrip("^*&")
>>> print(str5,str6,sep="\n")
nit***&&&^^^
nit
```

Syntax of strip method

variable-name=string-name.strip(chars=' ')

```
>>> str1="   nit    "
>>> str2=str1.strip()
>>> print(len(str1),len(str2))
10 3
>>> print(str1,str2)
   nit     nit
>>> str2="***nit****"
>>> str3=str2.strip("*")
>>> print(str2,str3,sep="\n")
***nit****
nit
str4="**$$$nit**&&&##$$$"
str5=str4.strip("*$&#")
>>> print(str4,str5,sep="\n")
**$$$nit**&&&##$$$
nit
>>> str6="www.nareshit.com"
>>> str7=str6.strip("w.com")
>>> print(str6,str7,sep="\n")
www.nareshit.com
nareshit
```