

In [8]:

```

1  # For examples:
2
3  n = int(input())
4  for i in range(1,n+1):
5      print("11AK1A05%02d\t12AK1A05%02d\t13AK1A05%02d"%(i,i,i))
6
7

```

```

10
11AK1A0501      12AK1A0501      13AK1A0501
11AK1A0502      12AK1A0502      13AK1A0502
11AK1A0503      12AK1A0503      13AK1A0503
11AK1A0504      12AK1A0504      13AK1A0504
11AK1A0505      12AK1A0505      13AK1A0505
11AK1A0506      12AK1A0506      13AK1A0506
11AK1A0507      12AK1A0507      13AK1A0507
11AK1A0508      12AK1A0508      13AK1A0508
11AK1A0509      12AK1A0509      13AK1A0509
11AK1A0510      12AK1A0510      13AK1A0510

```

In [11]:

```

1  # 4.5001
2  # 4.5
3  # print("%.")
4  # pyformat.info -> Format Specifiers
5
6
7  n = int(input())
8  for i in range(1,n+1):
9      print("11AK1A05{:02}\t12AK1A05{:02}".format(i,i))
10

```

```

10
11AK1A0501      12AK1A0501
11AK1A0502      12AK1A0502
11AK1A0503      12AK1A0503
11AK1A0504      12AK1A0504
11AK1A0505      12AK1A0505
11AK1A0506      12AK1A0506
11AK1A0507      12AK1A0507
11AK1A0508      12AK1A0508
11AK1A0509      12AK1A0509
11AK1A0510      12AK1A0510

```

In []:

```

1  Sample Test Case:
2  Input: Enter starting range from user: 250
3          Enter ending range from user: 260
4  Output:
5      11AK1A0250
6          .....
7          .....
8      11AK1A0260

```

In [14]:

```
1 n = 10
2 s = 0
3 for i in range(10):
4     print(i,end=" ")
5     s = s+i
6 print(s)
```

0 1 2 3 4 5 6 7 8 9 45

In [15]:

```
1 #print the factors of given number
2
3 #print the sum of n natural numbers
4
5 #print the even numbers in bettween given range
6 # both(lowerbound and upperbound) are inclusive
7
8 #check the given number is a prime or not
9
10 #print the prime numbers inbetween given range
```

In [17]:

```
1 # range(Lb,ub,step)
2 for i in range(1,10):
3     print(i,end=" ")
```

1 2 3 4 5 6 7 8 9

In [19]:

```
1 # range(Lb,ub,step)
2 for i in range(1,10,2):
3     print(i,end=" ")
4 #1 2 3 4 5 6 7 8 9
```

1 3 5 7 9

In [23]:

```
1 # 11,100
2 for i in range(5,100,5):
3     print(i,end=" ")
```

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95

In [25]:

```

1  # While:
2
3  #     Syntax:
4
5  #         while Condition:
6  #             stmnts
7  #             updation of itervariable
8
9  n = int(input())
10 c = 0
11 while n!=0:
12     c+=1
13     n=n//10
14 print(c)

```

5344678902615565168453515468135513511355163023156123021354468613515313235315
319876543
85

In []:

```

1  # Input: 132465
2  # Output: Given number is : 132465
3  #     Digit Count is: 6
4
5  Task - 1
6  # Input: 134234
7  # Output:
8  #     Given number is: 134234
9  #     Digit count is: 6
10 #     Reverse number is: 432431
11 #     Even Digit Count is: 3
12 #     Odd Digit Count is: 3
13 #     Even digits sum : 10
14 #     Odd digits sum: 7
15 -----
16 Task - 2 | Task -3
17 -----
18 Input: 5 | Input: 3
19 Output: | Ouput:
20     1 1 1 1 1 | * * *
21     2 2 2 2 2 | * *
22     3 3 3 3 3 | * * *
23     4 4 4 4 4 |
24     5 5 5 5 5 |
25

```

In []:

1

Strings

In [26]:

```
1 s = 'ramu'  
2 s1 = " mandava"  
3 s3 = '''fgfdhgpu'''
```

In [27]:

```
1 s[0]
```

Out[27]:

'r'

In [28]:

```
1 s[1]
```

Out[28]:

'a'

In [29]:

```
1 len(s)
```

Out[29]:

4

In [30]:

```
1 dir(str)
```

Out[30]:

```
['__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 [31]:

```
1 s
```

Out[31]:

```
'ramu'
```

In [32]:

```
1 s[0:2]
```

Out[32]:

```
'ra'
```

In [35]:

```
1 s[2:5]
```

Out[35]:

```
'mu'
```

In [36]:

```
1 s = 'fohohfghtph12'
```

In [42]:

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

Out[42]:

```
'12'
```

In [43]:

```
1 s[len(s)//2]
```

Out[43]:

```
'g'
```

In [44]:

```
1 s[::-1]
```

Out[44]:

```
'21hpthgfhohof'
```

In [45]:

```
1 s[0:len(s):2]
```

Out[45]:

```
'fhhgth2'
```

In [46]:

```
1 s
```

Out[46]:

```
'fohohfghtph12'
```

In [47]:

```
1 s[4:10]
```

Out[47]:

```
'hfghtp'
```

In [48]:

```
1 s.capitalize()
```

Out[48]:

```
'Fohohfghtph12'
```

In [49]:

```
1 s.casefold()
```

Out[49]:

```
'fohohfghtph12'
```

In [55]:

```
1 s.center(5)
```

Out[55]:

```
'fohohfghtph12'
```

In [56]:

```
1 s = ""
```

In [57]:

```
1 s=s+'r'
```

In [58]:

```
1 s
```

Out[58]:

```
'r'
```

In [62]:

```
1 s='ramu'
```

In [64]:

```
1 s.replace('a','j')
```

Out[64]:

```
'rjmu'
```

In [65]:

```
1 s.count('j')
```

Out[65]:

```
0
```


In [66]:

```
1 s
```

Out[66]:

```
'ramu'
```

In [67]:

```
1 s.count('a')
```

Out[67]:

```
1
```

In [68]:

```
1 s = '437657834832'
```

In [69]:

```
1 s.count('7')
```

Out[69]:

```
2
```

In [70]:

```
1 s[0]="6"
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-70-f97275ea236d> in <module>  
----> 1 s[0]="6"
```

TypeError: 'str' object does not support item assignment

In [71]:

```
1 s.endswith('2')
```

Out[71]:

```
True
```

In [72]:

```
1 s.startswith('91')
```

Out[72]:

```
False
```

In [73]:

```
1 s = 'hriuhdTREtr'  
2 s.swapcase()
```

Out[73]:

'HRIUHDtreTR'

In [75]:

```
1 s.islower()
```

Out[75]:

False

In [76]:

```
1 s
```

Out[76]:

'hriuhdTREtr'

In [77]:

```
1 s.islower()
```

Out[77]:

False

In [78]:

```
1 s.casefold()
```

Out[78]:

'hriuhdtretr'

In [79]:

```
1 s.islower()
```

Out[79]:

False

In [80]:

```
1 s
```

Out[80]:

'hriuhdTREtr'

In [85]:

```
1 s =s.casefold()
2 print(s)
```

hriuhdtretr

In [86]:

```
1 s.islower()
```

Out[86]:

True

In [90]:

```
1 s.index('')
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-90-664193e3e34f> in <module>
----> 1 s.index('2')
```

ValueError: substring not found

In [102]:

```
1 s = ' lk fg '
2 s = s.strip()
3 print(s)
4 s="".join(" ")
```

lk fg

In [95]:

```
1 s =s.join(' ')
2 print(s)
```

In [105]:

```
1 d = "sdasd a asd asd"
2 d = "".join(d)
3 d
```

Out[105]:

'sdasd a asd asd'

In [107]:

```
1 s = 'jhgfdsd jdfhl'
```

In [110]:

```
1  #' ' or '"
2  s = "python"
3  # if you want acces the charcters from
4  # the string here we are having a three ways to access
5  #1.By using positive indexing
6  #2.By using negative indexing
7  #3.By using for loop also we can also
```

In [114]:

```
1  #1.By using positive indexing
2  print(s)
3  s[0]#It gives first character of string
4  s[1]#It gives second character of the string
5  s[2]
```

python

Out[114]:

't'

In [118]:

```
1  #2.By using negative indexing
2  s[-1]#It gives last character of string
3  s[-2]
```

Out[118]:

'o'

In [119]:

```
1  #If you want find the string length there is predefined
2  # method called len()
3  print(len(s))
4
```

6

In [123]:

```
1  #3.By using for loop also we can also
2  for i in range(len(s)):
3      print(s[i],end = "")
```

python

In [124]:

```
1 s
```

Out[124]:

```
'python'
```

In [126]:

```
1 s[0:3]
```

Out[126]:

```
'pyt'
```

In [127]:

```
1 s1 = "Python Programming"  
2 len(s1)
```

Out[127]:

```
18
```

In [129]:

```
1 s1[0:6]
```

Out[129]:

```
'Python'
```

In [131]:

```
1 s1[6:]
```

Out[131]:

```
' Programming'
```

In [132]:

```
1 s1[:6]
```

Out[132]:

```
'Python'
```

In [133]:

```
1 s1[0:6:1]
```

Out[133]:

```
'Python'
```

In [134]:

```
1 s1[::-2]
```

Out[134]:

'Pto rgamn'

In [137]:

```
1 s1[18:0:-1]
```

Out[137]:

'gnimmargorP nohty'

In [138]:

```
1 s1[::-1]
```

Out[138]:

'gnimmargorP nohtyP'

In [141]:

```
1 #check the given string is palindrome or not
2 s = input()
3 rs = s[::-1]
4 if s == rs:
5     print("Palindrome")
6 else:
7     print("not Palindrome")
8 i
```

liril
Palindrome

Out[141]:

5

In [142]:

```
1 dir(str)
```

Out[142]:

```
['__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 [143]:

```
1 s
```

Out[143]:

```
'liril'
```

In [144]:

```
1 s = "python"
```

In [145]:

```
1 s
```

Out[145]:

```
'python'
```

In [146]:

```
1 s.capitalize()
```

Out[146]:

```
'Python'
```


In [148]:

```
1 s1 = "python programming"
2 s1
```

Out[148]:

'python programming'

In [149]:

```
1 s1.capitalize()
```

Out[149]:

'Python programming'

In [150]:

```
1 s1.title()
```

Out[150]:

'Python Programming'

In [151]:

```
1 s = "Hello Good Afternoon"
2 print(s)
```

Hello Good Afternoon

In [152]:

```
1 for i in s:
2     print(i,end = " ")
```

H e l l o G o o d A f t e r n o o n

In [153]:

```
1 for i in range(len(s)):
2     print(s[i],end=" ")
```

H e l l o G o o d A f t e r n o o n

In [156]:

```
1 s = "hello"
2 s*3
```

Out[156]:

'hello hello hello '

In [157]:

```
1 s.count('l')
```

Out[157]:

2

Collections in Python

- List
- Tuple
- Dictionary
- Set

List

- List is collection of heterogeneous data and it is mutable (changeable)
- List will support indexing
- If you want to create the list here we are having two ways i.e. 1. By using square braces [] and 2. by using list() predefined method

In [165]:

```
1 #Declaration of List
2 # Syntax: variable_name = [value1,value2,.....]
3 li = [1,2,"Python",55.4,True,"APSSDC"]
4 print(li)
5 print(type(li))
6 #If you want to access the list elements here we are having a
7 # three ways to access
8 #1.By using positive Indexing
9 #2.By using negative Indexing
10 #3.By using loop
```

```
[1, 2, 'Python', 55.4, True, 'APSSDC']
<class 'list'>
```

In [162]:

```
1 #1.By using positive Indexing
2 li[0]#It gives first element of the list
3 li[1]#It gives second element of the list
4 li[2]#It gives third element from the list
```

Out[162]:

'Python'

In [164]:

```
1 #2.By using negative Indexing
2 li[-1]#It gives last element from the list
3 li[-2]
```

Out[164]:

True

In [166]:

```
1 #If you want know the length of list there
2 # is a predefined method called len()
3 len(li)
```

Out[166]:

6

In [167]:

```
1 #3.By using loop
2 for i in range(len(li)):
3     print(li[i])
```

1
2
Python
55.4
True
APSSDC

In [168]:

```
1 for i in li:
2     print(i)
```

1
2
Python
55.4
True
APSSDC

In [174]:

```
1 #List Slicing
2 print(li)
3 li[0:5:2]
```

[1, 2, 'Python', 55.4, True, 'APSSDC']

Out[174]:

[1, 'Python', True]

In [175]:

```
1 li[::-1]
```

Out[175]:

```
['APSSDC', True, 55.4, 'Python', 2, 1]
```

In [176]:

```
1 myli = []
```

In [178]:

```
1 #myli[0] = 1
2 dir(list)
```

Out[178]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [179]:

```
1 myli.append(1)
```

In [180]:

```
1 myli
```

Out[180]:

```
[1]
```

In [181]:

```
1 myli.append(60)
```

In [182]:

```
1 myli
```

Out[182]:

```
[1, 60]
```

In [183]:

```
1 for i in range(10,20):  
2     myli.append(i)  
3 myli
```

Out[183]:

```
[1, 60, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [184]:

```
1 #If you want find the maximum elemnt in the list there is a  
2 # predefined method called max()  
3 max(myli)
```

Out[184]:

```
60
```

In [190]:

```
1 #If you want find the minimum elemnt in the list there is a  
2 # predefined method called min()  
3 min(myli)  
4
```

Out[190]:

```
1
```

In [192]:

```
1 # m = ["APSSDC", "apssdc", "Python"]  
2 # min(m)  
3 # max(m)  
4 # dir(list)
```

In [201]:

```
1 li = [1,20,40,20,1,70,80]
2 li.count(20)
```

Out[201]:

2

In [203]:

```
1 li.sort()
```

In [204]:

```
1 li
```

Out[204]:

[1, 1, 20, 20, 40, 70, 80]

In [205]:

```
1 li
```

Out[205]:

[1, 1, 20, 20, 40, 70, 80]

In [214]:

```
1 s = [40,20,50,70,80]
2 s = sorted(s,reverse = True)
```

In [215]:

```
1 s
```

Out[215]:

[80, 70, 50, 40, 20]

In [217]:

```
1 s.index(80)
```

Out[217]:

0

In [218]:

```
1 myli.clear()
```

In [219]:

```
1 myli
```

Out[219]:

```
[]
```

In [220]:

```
1 s
```

Out[220]:

```
[80, 70, 50, 40, 20]
```

In [221]:

```
1 s.pop()
```

Out[221]:

```
20
```

In [222]:

```
1 s
```

Out[222]:

```
[80, 70, 50, 40]
```

In [224]:

```
1 s.insert(3,90)
```

In [225]:

```
1 s
```

Out[225]:

```
[80, 70, 50, 90, 40]
```

In [226]:

```
1 s.remove(50)
```

In [227]:

```
1 s
```

Out[227]:

```
[80, 70, 90, 40]
```


In [228]:

```
1 li
```

Out[228]:

```
[1, 1, 20, 20, 40, 70, 80]
```

In [229]:

```
1 s.extend(li)
```

In [230]:

```
1 s
```

Out[230]:

```
[80, 70, 90, 40, 1, 1, 20, 20, 40, 70, 80]
```

In [231]:

```
1 li
```

Out[231]:

```
[1, 1, 20, 20, 40, 70, 80]
```

In [233]:

```
1 n = li.copy()
```

In [234]:

```
1 n
```

Out[234]:

```
[1, 1, 20, 20, 40, 70, 80]
```

In [235]:

```
1 k = "Good Afternoon APSSDC"  
2 k
```

Out[235]:

```
'Good Afternoon APSSDC'
```

In [236]:

```
1 list(k)
```

Out[236]:

```
['G',  
'o',  
'o',  
'd',  
,  
,  
'A',  
'f',  
't',  
'e',  
'r',  
'n',  
'o',  
'o',  
'n',  
,  
,  
'A',  
'P',  
'S',  
'S',  
'D',  
'C']
```

In [237]:

```
1 b = k.split(" ")  
2 b
```

Out[237]:

```
['Good', 'Afternoon', 'APSSDC']
```

In [238]:

```
1 a = k.split()  
2 a
```

Out[238]:

```
['Good', 'Afternoon', 'APSSDC']
```

In [239]:

```
1 s = "Hello-Apssdc-How-are-you"  
2 n = s.split("-")
```

In [240]:

```
1 n
```

Out[240]:

```
['Hello', 'Apssdc', 'How', 'are', 'you']
```

In [243]:

```
1 s = "Hello-Apssdc-How-are-you"
2 n = s.split("e")
```

In [244]:

```
1 n
```

Out[244]:

```
['H', 'llo-Apssdc-How-ar', '-you']
```

In [245]:

```
1 # 10 30 40 70 90 60
2 l = input().split()
3 l
```

```
10 30 40 70 90 60
```

Out[245]:

```
['10', '30', '40', '70', '90', '60']
```

In [250]:

```
1 #li = [20,80,100,101,700,500,700,500,20,10] find the max valu in the list
2 li = [20,80,100,101,700,500,700,500,20,10]
3 unique = []
4 for i in li:
5     if i not in unique:
6         unique.append(i)
7 print(unique)
8 sli = sorted(unique,reverse = True)
9 print(sli[1])
```

```
[20, 80, 100, 101, 700, 500, 10]
500
```

In [251]:

```
1 k = input().split()
2 k
```

```
40 20 80 50
```

Out[251]:

```
['40', '20', '80', '50']
```

In [253]:

```
1 intvalues = []
2 for i in k:
3     intvalues.append(int(i))
4 print(intvalues)
```

[40, 20, 80, 50]

In [254]:

```
1 intvalues
```

Out[254]:

[40, 20, 80, 50]

In [255]:

```
1 sum(intvalues)
```

Out[255]:

190

In [256]:

```
1 intvalues
```

Out[256]:

[40, 20, 80, 50]

In [257]:

```
1 intvalues.clear()
```

In [258]:

```
1 intvalues
```

Out[258]:

[]

In []:

```
1
```

Tuple

- Tuple is a collection of heterogeneous data and it accepts indexing also
- Tuple is Immutable (It can't be changeable)
- Declaration of tuple : By using normal paranthesis i.e '()'

In [263]:

```
1 #Creation of tuple
2 #Variblename = (Value1,valu2,...)
3 #If you want to create the tuple here we are having three ways i.e
4 # 1.tuple()
5 # 2.variblename = ()
6 # 3.comma seperated values(Ex:1,2,3,4,5)
7
```

In [264]:

```
1 1,2,3,4,5
```

Out[264]:

(1, 2, 3, 4, 5)

In [265]:

```
1 t = ("Mastan",12,50.5,True,"APSSDC")
2 t
```

Out[265]:

('Mastan', 12, 50.5, True, 'APSSDC')

In [266]:

```
1 #Accessing elements from tuple
2 #If you want to access the lements from the tuple here we are
3 # having a three ways i.e:
4 # 1.By using Positive Indexing
5 # 2.By using negative Indexing
6 # 3.By using for loop
```

In [270]:

```
1 # 1.By using Positive Indexing
2 print(type(t))
3 t[0]#It gives first element from the tuple
4 t[1]#It gives second element from the tuple
5
```

<class 'tuple'>

Out[270]:

12

In [271]:

```
1 # 2.By using negative Indexing
2 t[-1]#It gives last element from the tuple
```

Out[271]:

'APSSDC'

In [273]:

```
1 t[0:3]
```

Out[273]:

('Mastan', 12, 50.5)

In [274]:

```
1 t[::-1]
```

Out[274]:

('APSSDC', True, 50.5, 12, 'Mastan')

In [275]:

```
1 dir(tuple)
```

Out[275]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'count',
 'index']
```

In [277]:

```
1 t.index('Mastan')
```

Out[277]:

```
0
```

In [278]:

```
1 t.count('APSSDC')
```

Out[278]:

```
1
```

In [279]:

```
1 v = ()
```

In [280]:

```
1 type(v)
```

Out[280]:

tuple

In [282]:

```
1 40,50,70
```

Out[282]:

(40, 50, 70)

In [283]:

```
1 for i in t:  
2     print(i)
```

Mastan

12

50.5

True

APSSDC

In [285]:

```
1 li = [10,30,40,50]  
2 s = tuple(li)
```

In [286]:

```
1 s
```

Out[286]:

(10, 30, 40, 50)

In [287]:

```
1 list(s)
```

Out[287]:

[10, 30, 40, 50]

In []:

```
1
```


