Inner Loops:

```
- Loop within a loop
```

In [11]:

```python
n = int(input())
for i in range(1,n+1):
    for j in range(1,n+1):
        print("*",end="")
    print(end="\n")
```

```
4
****
****
****
****
```

In [9]:

```python
n = int(input())
for i in range(1,n+1):
    for j in range(1,n+1):
        if i==j or i+j==n+1:
            print("*",end="")
        else:
            print(" ",end="")
    print(end="\n")
```

```
7
*     *
 *   *
  * *
   *
  * *
 *   *
*     *
```

In [12]:

```
n = int(input())
for i in range(1,n+1):
    for j in range(1,n+1):
        if i==1 or i==n or j==1 or j==n:
            print("*",end="")
        else:
            print(" ",end="")
    print(end="\n")
```

```
5
*****
*   *
*   *
*   *
*****
```

In [16]:

```
n = int(input())
k = 0
for r in range(1,n+1):
    for c in range(1,n+1):
        print("{:02}".format(k+1),end=" ")
        k+=1
    print(end="\n")
```

```
3
01 02 03
04 05 06
07 08 09
```

Jumping Statements:

```
    - continue -> skip single value and prints all element
s
    - break -> exit of loop
    - pass -> nothing to print
    - return -> single or multiple values by using collect
ions
```

In [19]:

```python
k = int(input())
for h in range(1,k+1):
    if h==4 or h==7:
        continue
    else:
        print(h,end=" ")
```

```
10
1 2 3 5 6 8 9 10
```

In [20]:

```python
k = int(input())
for h in range(1,k+1):
    if h==4:
        break
    else:
        print(h,end=" ")
```

```
10
1 2 3
```

In [33]:

```python
k = 1
for r in range(1,6):
    for c in range(1,8):
        if k>=32:
            break
        else:
            if c==2:
                print("||",end=" ")
            elif r==4:
                print("##",end=" ")
            elif k%2==0:
                print("**",end=" ")
            elif k%3==0:
                print("()",end=" ")
            elif k<=10:
                print("::",end=" ")
            else:
                print("[]",end=" ")
            k+=1
    print(end="\n")
```

```
:: || () ** :: ** ::
** || ** [] ** [] **
() || [] ** [] ** ()
## || ## ## ## ## ##
[] || []
```

Functions:

```
- To Perform a specific task

- predefined -> print,sqrt,pow etc.,
- user defined functions
Syntax:

    def fun_name(arguments):
        //stmnts
        return
```

User defined functions

```
- With return type and with arguments
- With return type and without arguments
- Without return type and with arguments
- Without return type and without arguments
```

In [36]:

```python
# With return type and with arguments

def SumofDigits(k):
    s = 0
    while k!=0:
        r = k%10
        s+=r
        k=k//10
    return s

n = int(input())
print("Given number is : {}"
      " and its digit count is: {}"
      .format(n,SumofDigits(n)))
```

```
234
Given number is : 234 and its digit count is: 9
```

Functions arguments:

```
- Required argument
- Keyword argument
- Default argument and
- Value-length argument
```

In [38]:

```python
# Required arguments
def su(n,m):
    return n+m

k = int(input())
u = int(input())
su(k,u)
```

```
3
5
```

Out[38]:

```
8
```

In [45]:

```python
# Keyword argument:

def namedtls(n,a):
    print("Name is: {} and age is: {}".format(n,a))
    return

nam = input()
ag = int(input())
namedtls(nam,ag)
```

```
somu
23
Name is: somu and age is: 23
```

In [54]:

```python
# Default arguments

def Name(p,sal=25000):
    print("Entered name is: {} and sal is: {}"
          .format(p,sal))
    return

na = input()
Name('saral',sal=34500)
Name('arun')
```

```
rajesh
Entered name is: saral and sal is: 34500
Entered name is: arun and sal is: 25000
```

In [57]:

```python
# Value length arguments:

def NoP(*g):
    for e in g:
        print(e,end=" ")
    return

NoP(3,6,7,8,9,12,14,23,2,346)
```

```
3 6 7 8 9 12 14 23 2 346
```

Strings:

In [66]:

```python
# -> Slicing,indexed based,changes can be done

s = "Python Program"
print(s,type(s))
print(s[0])
print(s[2])
print(s[0:5])
print(s[5:8])
print(s[:5])
print(s[3:])
```

```
Python Program <class 'str'>
P
t
Pytho
n P
Pytho
hon Program
```

In [84]:

```python
s = "Python Program"
print(s[2:9:2])
print(s[9:2:-2])
print(s[-4])
print(s[-3:-5:-1])
print(s[-5:2:-1])
print(s[-5::-1])
print(s[-5::-2])
print(s[3::3])
print(s[-1::-1])
print(s[1:-1:1])
s
```

```
to r
oPnh
g
rg
orP noh
orP nohtyP
oPnhy
h oa
margorP nohtyP
ython Progra
```

Out[84]:

```
'Python Program'
```

In [90]:

```python
s = "Python Program"
print(len(s))
print(s[len(s)//2:])
```

```
14
Program
```

In [91]:

```python
print(dir(str))
```

```
['__add__', '__class__', '__contains__', '__delattr_
_', '__dir__', '__doc__', '__eq__', '__format__', '_
_ge__', '__getattribute__', '__getitem__', '__getnew
args__', '__gt__', '__hash__', '__init__', '__init_s
ubclass__', '__iter__', '__le__', '__len__', '__lt_
_', '__mod__', '__mul__', '__ne__', '__new__', '__re
duce__', '__reduce_ex__', '__repr__', '__rmod__', '_
_rmul__', '__setattr__', '__sizeof__', '__str__', '_
_subclasshook__', 'capitalize', 'casefold', 'cente
r', 'count', 'encode', 'endswith', 'expandtabs', 'fi
nd', 'format', 'format_map', 'index', 'isalnum', 'is
alpha', 'isascii', 'isdecimal', 'isdigit', 'isidenti
fier', 'islower', 'isnumeric', 'isprintable', 'isspa
ce', 'istitle', 'isupper', 'join', 'ljust', 'lower',
'lstrip', 'maketrans', 'partition', 'replace', 'rfin
d', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstr
ip', 'split', 'splitlines', 'startswith', 'strip',
'swapcase', 'title', 'translate', 'upper', 'zfill']
```

In [102]:

```python
ss = "pYthon PrOgRaM"
print(ss.capitalize())
print(ss.title())
print(ss.casefold())
print(ss.count('p'))
print(ss.swapcase())
print(ss.center(20))
```

```
Python program
Python Program
python program
1
PyTHON pRoGrAm
   pYthon PrOgRaM
```

In [118]:

```python
ss = "pYthon PrOgRaM"
print(ss.endswith('M'))
print(ss.startswith('P'))
print(ss.find('z'))
print(ss.index('P'))
```

```
True
False
-1
7
```

In [131]:

```python
s = 'wei2342'
print(s.isalnum())
d = 'dsbbfjgsf'
print(d.isalpha())
e = '1'
print(e.isascii())
h = '3445748723'
print(h.isdecimal())
l = '34234672'
print(l.isdigit())
```

```
True
True
True
True
True
```

In [152]:

```python
g = 'asdasd'
print(g.isidentifier())
k = 'adasdawsa'
print(k.islower())
m = 'ASDASDASD'
print(m.isupper())
dd = '1286381235123'
print(dd.isnumeric())
sp = " "
print(sp.isspace())
st = 'Python Workshop'
print(st.istitle())
```

```
True
True
True
True
True
True
```

In [177]:

```python
d = "python"
g = "Workshop"
h = " apssdc"
k = "Vemu "
print("@".join(g))
print(g)
print(",".join(d))
print(d)
print(d+" "+g)
print(h.rjust(20))
print(k.ljust(30))
```

```
W@o@r@k@s@h@o@p
Workshop
p,y,t,h,o,n
python
python Workshop
              apssdc
Vemu
```

In [187]:

```python
d = "Vemu college   "
ll = "      Raju"
print(d.strip())
print(ll.strip())
print(d.rstrip())
print(ll.lstrip())
```

```
Vemu college
Raju
Vemu college
Raju
```

In [185]:

```python
k = "asfkljaslj asdjf kjsadlkjf alkjsf lkasdflknuehr"
print(k.split())
print(k.split('a'))
```

```
['asfkljaslj', 'asdjf', 'kjsadlkjf', 'alkjsf', 'lkas
dflknuehr']
['', 'sfklj', 'slj ', 'sdjf kjs', 'dlkjf ', 'lkjsf l
k', 'sdflknuehr']
```

```
k = '1273kjaAR!#%'
Alphabets Capital case are: 2
Alphabets Lower case are: 3
Numbers are : 4
Special Characters are: 3
```

In [197]:

```python
st = input()
cp = lp = dg = sp = 0
# print(st,type(st),len(st))
for i in st:
    if i.isalpha():
        if i.isupper():
            cp+=1
        else:
            lp+=1
    elif i.isdigit():
        dg+=1
    else:
        sp+=1
print("Alphabets Capital case are: {}"
      .format(cp))
print("Alphabets Lower case are: {}".format(lp))
print("Numbers are: {}".format(dg))
print("Special Characters are: {}".format(sp))
```

```
qy123AS@#$
Alphabets Capital case are: 2
Alphabets Lower case are: 2
Numbers are: 3
Special Characters are: 3
```

Data Structures in Python:

```
collections, Iterators

   - List -> Changes can be done,
               [],
               list(),
               Indexing is done,
               heterogeneous data,
               slicing can be done,
               ordered data
   - Tuple -> Changes can't be done,
               (),
               tuple(),
               Indexing is done,
               heterogenous data,
               slicing can be done,
               ordered data
   - Set -> Changes can be done,
               {},
               set(),
               duplicate elements are removed,
               Indexing is not there,
               heterogenous data,
               slicing can't be done,
               Unordered data
   - Dictionary -> Changes can be done,
               key and value pairs can exists,
               {'k':34},
               indexing is there,
               duplicate elements are removed by keys,
               ordered data,
```

List:

In [202]:

```python
k = [23,'sfas',3.34786,False]
print(k)
print(type(k))
print(k[2])
print(k[1:3])
```

```
[23, 'sfas', 3.34786, False]
<class 'list'>
3.34786
['sfas', 3.34786]
```

In [203]:

```python
print(dir(list()))
```

```
['__add__', '__class__', '__contains__', '__delattr_
_', '__delitem__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__getit
em__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le_
_', '__len__', '__lt__', '__mul__', '__ne__', '__new
__', '__reduce__', '__reduce_ex__', '__repr__', '__r
eversed__', '__rmul__', '__setattr__', '__setitem_
_', '__sizeof__', '__str__', '__subclasshook__', 'ap
pend', 'clear', 'copy', 'count', 'extend', 'index',
'insert', 'pop', 'remove', 'reverse', 'sort']
```

In [205]:

```python
lis = [1,4,6,3,2]
lis1 = [5,8,1,9,0]
print(lis+lis1)
print(lis)
print(lis1)
```

```
[1, 4, 6, 3, 2, 5, 8, 1, 9, 0]
[1, 4, 6, 3, 2]
[5, 8, 1, 9, 0]
```

In [208]:

```
print(lis)
lis.append(67)
print(lis)
```

```
[1, 4, 6, 3, 2, 34]
[1, 4, 6, 3, 2, 34, 67]
```

In [209]:

```
h=lis.copy()
print(h)
print(lis)
```

```
[1, 4, 6, 3, 2, 34, 67]
[1, 4, 6, 3, 2, 34, 67]
```

In [213]:

```
print(lis.count(2))
print(lis)
lis.extend([34,56,67,2,3,59])
print(lis)
```

```
2
[1, 4, 6, 3, 2, 34, 67, 34, 56, 67, 2, 3, 59]
[1, 4, 6, 3, 2, 34, 67, 34, 56, 67, 2, 3, 59, 34, 5
6, 67, 2, 3, 59]
```

In [225]:

```
print(lis1)
print(lis1.index(0))
lis1.insert(6,400)
print(lis1)
```

```
[5, 100, 8, 1, 9, 400, 0]
6
[5, 100, 8, 1, 9, 400, 400, 0]
```

In [230]:

```
print(lis1)
lis1.remove(0)
print(lis1)
```

```
[5, 100, 8, 1, 9, 400, 0]
[5, 100, 8, 1, 9, 400]
```

In [232]:

```
print(lis1)
lis1.pop(0)
print(lis1)
```

```
[5, 100, 8, 1, 9]
[100, 8, 1, 9]
```

In [235]:

```
print(lis1)
lis1.reverse()
print(lis1)
lis1.sort()
print(lis1)
lis1.sort(reverse=True)
print(lis1)
```

```
[1, 8, 9, 100]
[100, 9, 8, 1]
[1, 8, 9, 100]
[100, 9, 8, 1]
```

In [236]:

```
print(lis1)
lis1.clear()
print(lis1)
```

```
[100, 9, 8, 1]
[]
```

In [238]:

```
lis1=[34,233465,456456,456]
print(lis1)
del lis1
```

```
[34, 233465, 456456, 456]
```

In [243]:

```
k = input().split()
print(k)
m = []
for i in k:
    m.append(int(i))
print(m)
```

```
23 4 6 7 8 9 0 0 7 75 54
['23', '4', '6', '7', '8', '9', '0', '0', '7', '75',
'54']
[23, 4, 6, 7, 8, 9, 0, 0, 7, 75, 54]
```

In [248]:

```
n = input().split()
el, ol = [], []
for i in n:
    if int(i)%2==0:
        el.append(int(i))
    else:
        ol.append(int(i))
el.sort()
ol.sort()
print(el)
print(ol)
print(len(el))
print(len(ol))
print(sum(el))
print(sum(ol))
```

```
34 4 5 6 7 12 0 9 8 3 5 6 77
[0, 4, 6, 6, 8, 12, 34]
[3, 5, 5, 7, 9, 77]
7
6
70
106
```

Tuple:

In [251]:

```
p = (34,'somu',56.00,True)
print(p)
print(type(p))
print(p[2])
print(p[1:3])
```

```
(34, 'somu', 56.0, True)
<class 'tuple'>
56.0
('somu', 56.0)
```

In [252]:

```
print(dir(tuple()))
```

```
['__add__', '__class__', '__contains__', '__delattr_
_', '__dir__', '__doc__', '__eq__', '__format__', '_
_ge__', '__getattribute__', '__getitem__', '__getnew
args__', '__gt__', '__hash__', '__init__', '__init_s
ubclass__', '__iter__', '__le__', '__len__', '__lt_
_', '__mul__', '__ne__', '__new__', '__reduce__', '_
_reduce_ex__', '__repr__', '__rmul__', '__setattr_
_', '__sizeof__', '__str__', '__subclasshook__', 'co
unt', 'index']
```

In [258]:

```
print(p)
print(p.index('somu'))
print(p.count(2))
```

```
(34, 'somu', 56.0, True)
1
0
```

In [262]:

```
k = input().split()
m = []
for i in k:
    m.append(int(i))
l = tuple(m)
print(l)
f = list(l)
f.sort()
g = tuple(f)
print(g)
```

```
2356 23 4 5 60 9 8 7 5 43 2 1
(2356, 23, 4, 5, 60, 9, 8, 7, 5, 43, 2, 1)
(1, 2, 4, 5, 5, 7, 8, 9, 23, 43, 60, 2356)
```

Set

In [263]:

```
s = {23,2,1,1.00,'ascii','aa','rajesh'}
print(s)
print(type(s))
s
```

```
{'aa', 1, 2, 'ascii', 'rajesh', 23}
<class 'set'>
```

Out[263]:

```
{1, 2, 23, 'aa', 'ascii', 'rajesh'}
```

In [265]:

```
s = {1,1,1,1,3,3,3,3,5,55,5,5,6,6,7,7}
print(s)
s
```

```
{1, 3, 5, 6, 7, 55}
```

Out[265]:

```
{1, 3, 5, 6, 7, 55}
```

In [266]:

```
print(dir(set))
```

```
['__and__', '__class__', '__contains__', '__delattr_
_', '__dir__', '__doc__', '__eq__', '__format__', '_
_ge__', '__getattribute__', '__gt__', '__hash__', '_
_iand__', '__init__', '__init_subclass__', '__ior_
_', '__isub__', '__iter__', '__ixor__', '__le__', '_
_len__', '__lt__', '__ne__', '__new__', '__or__', '_
_rand__', '__reduce__', '__reduce_ex__', '__repr__',
'__ror__', '__rsub__', '__rxor__', '__setattr__', '_
_sizeof__', '__str__', '__sub__', '__subclasshook_
_', '__xor__', 'add', 'clear', 'copy', 'difference',
'difference_update', 'discard', 'intersection', 'int
ersection_update', 'isdisjoint', 'issubset', 'issupe
rset', 'pop', 'remove', 'symmetric_difference', 'sym
metric_difference_update', 'union', 'update']
```

In [ ]:

```

```