In [1]:
```
ord("C")
```

Out[1]:
```
67
```

In [2]:
```
for i in range(1,5):
    for j in range(i,5):
        print("$",end=" ")
    for k in range(1,i+1):
        print("*",end=" ")
    print(" ",end="\n")
```

```
$ $ $ $ *
$ $ $ * *
$ $ * * *
$ * * * *
```

In [3]:
```
# while loop
# while (condition)

#for i in range(1,5)
```

In [4]:
```
#list , tuple , dictionary,set
```

In [5]:
```
# List: collection of element , which has index position
# mutuable datatype
# ordered collection of element

mylist=[10,20,"naina"]
print(type(mylist))
```

```
<class 'list'>
```

In [6]:
```
mylist[0:15]
```

Out[6]:
```
[10, 20, 'naina']
```

In [7]:
```
mylist[0]=100
```

In [8]:
```
mylist.append(50)
print(mylist)
```

```
[100, 20, 'naina', 50]
```

In [28]:
```
# in list we can'nt add 2 element at same time
mylist.append(50,100)
print(mylist)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 2
      1 # in list we can'nt add 2 element at same time
----> 2 mylist.append(50,100)
      3 print(mylist)

TypeError: list.append() takes exactly one argument (2 given)
```

In [ ]:
```
#extend ke use se multiple elements same time me add kr skte h
# kese bhe data pr loop chala skte h vo iterable he
```

In [29]:
```
mylist.extend("hey")
print(mylist)
```

```
[100, 20, 'naina', 50, 'h', 'e', 'y']
```

In [30]:
```
mylist.extend(["hey",99])
print(mylist)
```

```
[100, 20, 'naina', 50, 'h', 'e', 'y', 'hey', 99]
```

In [31]:
```
# in  this help it will tell about that function
help(mylist)
```

```
 |      The reverse flag can be set to sort in descending order.
 |
 |  ----------------------------------------------------------------------
 |  Class methods defined here:
 |
 |  __class_getitem__(...) from builtins.type
 |      See PEP 585
 |
 |  ----------------------------------------------------------------------
 |  Static methods defined here:
 |
 |  __new__(*args, **kwargs) from builtins.type
 |      Create and return a new object.  See help(type) for accurate signa
ture.
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes defined here:
 |
 |  __hash__ = None
```

In [32]:
```
# pop delete the last element and we can give value also to delete the value fi
print(mylist)
mylist.pop()
print(mylist)
```

```
[100, 20, 'naina', 50, 'h', 'e', 'y', 'hey', 99]
[100, 20, 'naina', 50, 'h', 'e', 'y', 'hey']
```

In [33]:
```
print(mylist)
mylist.remove("naina")
print(mylist)
```

```
[100, 20, 'naina', 50, 'h', 'e', 'y', 'hey']
[100, 20, 50, 'h', 'e', 'y', 'hey']
```

In [34]:
```
print(mylist)
del mylist[1]
print(mylist)
```

```
[100, 20, 50, 'h', 'e', 'y', 'hey']
[100, 50, 'h', 'e', 'y', 'hey']
```

In [35]:
```
mlist=[1,2,3,'yash','sankhla']
for i in mlist:
    print(i)
```

```
1
2
3
yash
sankhla
```

In [36]:
```
newlist=[]
mlist=[1,2,3,'yash','sankhla']
for i in mlist:
    if (type(i) == int):
        newlist.append(i**2)

print(newlist)
```

```
[1, 4, 9]
```

In [37]:
```
num=153
x=num
total=0

while(num>0):
    rem=num%10
    total+=rem**3
    num=num//10

if(total==x):
    print("armstrong")
else:
    print("not armstrong")
```

```
armstrong
```

In [ ]:
```
# TUPLE
# IMMUTABLE
```

In [21]:
```
mytuple=10
print(type(mytuple))
```

```
<class 'int'>
```

In [24]:
```
mytuple=10,20
print(type(mytuple))
```

```
<class 'tuple'>
```

In [23]:
```
mytuple=10,20,"hello"
print(type(mytuple))
```

```
<class 'tuple'>
```

In [27]:
```
mytuple+(60,70)
```

Out[27]:
```
(10, 20, 60, 70)
```

In [40]:
```
# because of memory address
mytuple=(10,20)
print(id(mytuple))

mytuple=mytuple+(60,70)
print("after",id(mytuple))

print(mytuple)
```

```
1856215290496
after 1856203229120
(10, 20, 60, 70)
```

In [ ]:
```
#dictionary
# collection of element key : value
# key:identifier
# value is data
# ordered collection of element
```

In [42]:
```
mydictionary={10:"tushar",20:"abhishek"}
```

In [43]:
```
mydictionary
```

Out[43]:
```
{10: 'tushar', 20: 'abhishek'}
```

In [44]:
```
mydictionary[10]
```

Out[44]:
```
'tushar'
```

In [47]:
```
#update
mydictionary[10]="aman"
print(mydictionary)
```

```
{10: 'aman', 20: 'abhishek'}
```

In [49]:
```
mydictionary[30]="aman"
print(mydictionary)
```

```
{10: 'aman', 20: 'abhishek', 30: 'aman'}
```

In [58]:
```
# delete
print(mydictionary)

#x=mydictionary.pop(10)
print(mydictionary)
```

```
{20: 'abhishek', 30: 'aman'}
{20: 'abhishek', 30: 'aman'}
```

In [59]:
```
#"hello"
#{"h":1, "e" : 1}

mydict={}
for i in "hello":
    mydict[i]=1
print(mydict)
```

```
{'h': 1, 'e': 1, 'l': 1, 'o': 1}
```

In [60]:
```
mydictionary={10:"tushar",20:"abhishek",10:"YASH"}
print(mydictionary)
```

```
{10: 'YASH', 20: 'abhishek'}
```

In [63]:
```
mydictionary={10:"tushar",20:"abhishek",10:"YASH"}
"yash" in mydictionary
```

Out[63]:
```
False
```

In [64]:
```
mydictionary={10:"tushar",20:"abhishek",10:"YASH"}
"YASH" in mydictionary
```

Out[64]:
```
False
```

In [78]:
```
mydict={}
for char in "hello hey how are you":
    if(char in "aeiou"):
        if char not in mydict:
            mydict[char]=1
        else:
            mydict[char]+=1
print(mydict)
```

```
{'e': 3, 'o': 3, 'a': 1, 'u': 1}
```

In [81]:
```
mydict
```

Out[81]:
```
{'e': 3, 'o': 3, 'a': 1, 'u': 1}
```

In [82]:
```
for i in mydict:
    print(i,mydict[i])
```

```
e 3
o 3
a 1
u 1
```

In [85]:
```
mydictionary
for i in mydictionary:
    print(i,mydictionary[i])
```

```
10 YASH
20 abhishek
```

In [86]:
```
mydictionary
for i in mydictionary:
    print(i)
```

```
10
20
```

In [89]:
```
mydictionary
for i in mydictionary:
    print(mydictionary[i])
```

```
YASH
abhishek
```

In [ ]:

In [ ]:

In [ ]: