In [4]:
```
#write program to perform circular shift on list to the right
arr = [1,2,3,4,5,6,7]
shiftNo = int(input("enter number of shift: "))
print(arr[-shiftNo:len(arr)]+arr[0:len(arr)-shiftNo])
```

```
enter number of shift: 2
[6, 7, 1, 2, 3, 4, 5]
```

In [10]:
```
#write program to transform a matrix represent as list of list
m=[[1,2,3],[4,5,6,],[7,8,9]]
print("orignal matrix")
for i in m:
    print(i)
print()
print("Transpose Matrix")
for i in range(len(m[0])):
    new=[row[i] for row in m]
    print(new)
```

```
orignal matrix
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Transpose Matrix
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]
```

In [11]:
```
m=[[1,2,3,4],[4,5,6,7],[7,8,9,5]] #
print("orignal matrix")
for i in m:
    print(i)
print()
print("Transpose Matrix")
for i in range(len(m[0])):
    new=[row[i] for row in m]
    print(new)
```

```
orignal matrix
[1, 2, 3, 4]
[4, 5, 6, 7]
[7, 8, 9, 5]

Transpose Matrix
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]
[4, 7, 5]
```

In [13]:
```
#write program to print element with frequence grater than given value k.
l=[1,1,1,1,2,2,2,2,3,3,5,5,5,6,7]
k=int(input("Enter value: "))
count=0
for i in range(len(l)):
    a=l.count(i)
    if a>k:
        print(i,end=" ")
```

```
Enter value: 2
1 2 5
```

In [17]:
```
#write program find index at which element of 2 list desen't match
l1=[1,2,3,4]
l2=[1,5,3,6]
index=[]
```

```
if len(l1)==len(l2):
    for i in range(len(l1)):
        if l1[i]!=l2[i]:
            index.append(i)
else:
    print("not equl")
print("list not match the index:",index)
```

```
list not match the index: [1, 3]
```

In [19]:
```
#write program count number of string where the string length is 3 or more and first
s=["abc","xyz","aba","2112","12341","12345"]
count=0
for i in s:
    if len(i)>=3 and i[0]==i[-1]:
        count+=1
print(count)
```

```
3
```

dictinary

In [20]:
```
d={}
print(type(d))
```

```
<class 'dict'>
```

In [21]:
```
d[10]="Apple"
d[20]="Banna"
d[30]="mango"
print(d)
```

```
{10: 'Apple', 20: 'Banna', 30: 'mango'}
```

In [30]:
```
d={10: 'Apple', 20: 'Banna', 30: 'mango'}
print(d[20])
# print(d[40])#key error
if 40 in d:
    print(d[40])
d[40]="Cherry"
print(d)
d[10]="strawberry"
print(d)
del(d[40])
print(d)
d.clear()
print(d)
# del d
# print(d)#name error
```

```
Banna
{10: 'Apple', 20: 'Banna', 30: 'mango', 40: 'Cherry'}
{10: 'strawberry', 20: 'Banna', 30: 'mango', 40: 'Cherry'}
{10: 'strawberry', 20: 'Banna', 30: 'mango'}
{}
```

In [31]:
```
squre={x:x*x for x in range(1,6)}
print(squre)
double={x:2*x for x in range (1,6)}
print(double)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
{1: 2, 2: 4, 3: 6, 4: 8, 5: 10}
```

In [33]:
```
d=dict([(10,"Apple"),(20,"banna"),(30,"cherry")])
print(d)
print(len(d))
```

```
{10: 'Apple', 20: 'banna', 30: 'cherry'}
3
```

In [36]:
```
s={1:20,2:30,3:40}#copy function
s1=s.copy()
s1[1]=60
print(s)
```

```
{1: 20, 2: 30, 3: 40}
```

In [38]:
```
d={10:"apple",20:"Banna"}#update
x={30:"apple",40:"Banna"}
d.update(x)
print(d)
print(x)
```

```
{10: 'apple', 20: 'Banna', 30: 'apple', 40: 'Banna'}
{30: 'apple', 40: 'Banna'}
```

In [39]:
```
d={10:"lucky",20:"arman",30:"aryan"}
print(d[10])
print(d.get(10))
print(d.get(40))
print(d.get(10,"guest"))
print(d.get(40,"guest"))
```

```
lucky
lucky
None
lucky
guest
```

In [43]:
```
d={10:"lucky",20:"arman",30:"aryan"}
print(d.popitem())
print(d)
d.clear()
# print(d.popitem())key error
```

```
(30, 'aryan')
{10: 'lucky', 20: 'arman'}
```

In [51]:
```
d={10:"lucky",20:"arman",30:"aryan"}
print(d.keys())
for i in d.keys():
    print(i)
print(d.values())
for i in d.values():
    print(i)
print(d.items())
for k,v in d.items():
    print(k,"->",v)
```

```
dict_keys([10, 20, 30])
10
20
30
dict_values(['lucky', 'arman', 'aryan'])
lucky
arman
aryan
dict_items([(10, 'lucky'), (20, 'arman'), (30, 'aryan')])
10 -> lucky
```

```
20 -> arman
30 -> aryan
```

In [52]:
```
d={10:"lucky",20:"arman",30:"aryan"}
print(d.setdefault(40,"Vishal"))
print(d)
print(d.setdefoult(10,"om"))
print(d)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-52-0ae64e932784> in <module>
      1 d={10:"lucky",20:"arman",30:"aryan"}
----> 2 print(d.setdefoult(40,"Vishal"))
      3 print(d)
      4 print(d.setdefoult(10,"om"))
      5 print(d)

AttributeError: 'dict' object has no attribute 'setdefoult'
```

In [5]:
```
s={40,10,20,30}
s.remove(30)
# print(s)
# s.remove(50)error
print(s)
s={40,10,20,30}
s.discard(30)
print(s)
```

```
{40, 10, 20}
{40, 10, 20}
```

In [6]:
```
s={10,20,30,40}
s.clear()
print(s)
```

```
set()
```

In [8]:
```
s={x*x for x in range(1,4)}
print(s)
s={2**x for x in range(1,6)}
print(s)
```

```
{1, 4, 9}
{32, 2, 4, 8, 16}
```

In [17]:
```
#write program to replace the words from dictionary your given a string replace its

# s="campusX is best for ds student"
# d={'best':"the best chennel",'DS':"Data-science"}
# x=s.replace("best"," the best chennal").replace("ds","Data-science")
# print(x)

s="campusX is best for ds student"
d={'best':"the best chennel",'DS':"Data-science"}
words=s.split()
l=[]
for i in words:
    if i in d:
        l.append(d[i])
    else:
        l.append(i)

print(" ".join(l))
```

```
campusX is the best chennel for ds student
```

given a list l of size n you need to count the number of special elements in the given list and element is special if removal of that element makes the list balance the list will be balanced if some of given index elements is equal to sum of all index elements also print the updated list after removal of special elements

In [23]:
```
# def count(lst):
#     special= []
#     for i in range(len(lst)):
#         temp= lst[:i] + lst[i+1:]
#         if sum(temp[:i]) == sum(temp[i:]):
#             special.append(lst[i])
#     updated = [x for x in lst if x not in special]
#     return len(special), updated
# l = [5,2,5,8]
# count, updated = count(l)
# print(f"Number of special elements: {count}")
# print(f"Updated list: {updated}")
```

```
Number of special elements: 0
Updated list: [5, 2, 5, 8]
```

In [31]:
```
# l=[5,5,2,5,8]
# k=[]
# sum1=0
# sum2=0
# for k in l[::2]:
#     sum1+=1
# for k in l[1::2]:
#     sum2+=k
# print(l)
# print(f"The special element is: {k}")
```

```
[5, 5, 2, 5, 8]
the special element is: 5
```

In [39]:
```
l=eval(input("enter list: "))
print("Original List: ",l)
count=0
for i in range(len(l)):
    c=l.copy()
    c.pop(i)
    sume=sum0=0
    for k in c[::2]:
        sume+=k
    for k in c[1::2]:
        sum0+=k
    if sume==sum0:
        print("index to be removed: ",i)
        count+=1
        print("List after removed of index: ", i ,"is:", l[:i] + l[i+1:])
print("Total special element: ",count)
```

```
enter list: [5,5,2,5,8]
Original List:  [5, 5, 2, 5, 8]
```

```
index to be removed:  0
List after removed of index:  0 is: [5, 2, 5, 8]
index to be removed:  1
List after removed of index:  1 is: [5, 2, 5, 8]
Total special element:  2
```

LAMBDA FUNCTION

In [48]:
```
#Normal funcation

def square(n):
    return n*n
print(square(5))
```

```
25
```

In [47]:
```
#lambda function
s= lambda x:x*x
print(s(4))
```

```
16
```

In [49]:
```
#sum
s=lambda a,b:a+b
print(s(10,20))
```

```
30
```

In [50]:
```
l=[1,2,3,4,5]
def double(x):
    return 2*x
l1=list(map(double,l))
print(l1)
```

```
[2, 4, 6, 8, 10]
```

In [51]:
```
#by using lambda
l=[1,2,3,4,5]
s=lambda x:2*x
l1=list(map(double,l))
print(l1)
```

```
[2, 4, 6, 8, 10]
```

In [56]:
```
l1=[1,2,3,4,5]
l2=[3,4,5,6,7]
l=list(map(lambda a,b: a+b,l1,l2))
print(l)
```

```
[4, 6, 8, 10, 12]
```

In [60]:
```
#reduce function lib is functool
from functools import *
l=[10,20,30,40,50]
sum=reduce(lambda x,y:x+y,l)
print(sum)
mul=reduce(lambda x,y:x*y,l)
print(mul)
```

```
150
12000000
```

In [63]:
```
#filter
def isEven(x):
    if x%2==0:
        return  True
```

```
    else:
        return False
l=[0,1,2,3,4,5,6,7,8,9,10]
l1=list(filter(isEven,l))
print(l1)
```

```
[0, 2, 4, 6, 8, 10]
```

In [17]:
```
l=[1,2,3,4,5]
s=set(l)
s.add(10)
print(5)

x=frozenset(l)

print(x)
```

```
5
frozenset({1, 2, 3, 4, 5})
```

In [3]:
```
#WRITER program to coverts string's given in list uppeer case by using map funcation
word=["apple","banna","cherry"]
l=list(map(str.upper,word))
print(l)
```

```
['APPLE', 'BANNA', 'CHERRY']
```

In [ ]:
```
<!-- #write the program to filter the words with greather than 5 -->
```

In [5]:
```
word=["apple","banna","cherry","abn"]
newWord=filter(lambda x:len(x)>=5,word)
for i in newWord:
    print(i)
```

```
apple
banna
cherry
```

In [9]:
```
#write program to find numbers divisible by 19 or 13 from a  list of function lambda
number=[1,2,10,13,7,19,10,25,36]
newNum=filter(lambda x:x%13==0 or x%19==0,number)
print(list(newNum))
```

```
[13, 19]
```

In [12]:
```
a=["appple","banna", "cherry","date"]
l=sorted(a,reverse=True)#key=len
print(l)
```

```
['date', 'cherry', 'banna', 'appple']
```

In [16]:
```
a=["appple","banna", "cherry","date"]

l=sorted(a,key=lambda x:x[-1])
print(l)
```

```
['banna', 'appple', 'date', 'cherry']
```

In [18]:
```
a=["apple","kiwi","date","banna","cherry"]
a.sort(key=len)
print(a)
```

```
['kiwi', 'date', 'apple', 'banna', 'cherry']
```

In [23]:
```
a=[(1,3),(2,2),(1,3)]
a.sort(key=lambda x:x[-1])
print(a)
```

```
[(2, 2), (1, 3), (1, 3)]
```

In [28]:
```
t="apple#banna#orange#cherry"
x=t.split("#",1)
print(x)
y=t.split("#",2)
print(y)
z=t.split("#",3)
print(z)
```

```
['apple', 'banna#orange#cherry']
['apple', 'banna', 'orange#cherry']
['apple', 'banna', 'orange', 'cherry']
```

In [29]:
```
l=["aryan","aman","Dhairy","Lukey"]
print(l[-1][-1])
```

```
y
```

In [30]:
```
min=lambda x,y:x if x<y else y
print(min(101*99,102*98))
```

```
9996
```

In [32]:
```
x={1,2,3,4,5}
y={3,4,5,6,7}
z={1,3,5,7,9}
print((x|y) or(x|y))
```

```
{1, 2, 3, 4, 5, 6, 7}
```

<!-- how many pizza you want to order-number form user

costomize pizza 1 size--> small(50),(medium(100),large(200) topping --> how many topping center number cron,tomato,onion,capscum--->20rs musroom,olivas,brocall-->50rs chees-->how many types you want(enter number) feta,mozarella,cheador --->60rs -->

In [34]:
```
orders = int(input("Enter the number of pizza that you want to order"))
pizzaAmounts = []
totalCost = 0
global totalCost

topping1 = ['corn', 'tomato', 'onion', 'capiscum']
topping2 = ['mushroom', 'olive', 'onion', 'broccoli']


def pizzaOrder():
    currentBill = 0
    print("Size")
    print("Small     Medium      Large")
    print('---------------------------------')
    print("50        100         200")
    pizzaSize = input("Enter the size of the pizza: ")
    if(pizzaSize.lower() == 'small'):
        currentBill += 50
    elif(pizzaSize.lower() == 'medium'):
        currentBill += 100
    else:
        currentBill += 200
    print('\n')
    noOfToppings = int(input("Enter the number of toppings that you want: "))
```

```
    for i in range(noOfToppings):
        print('\n')
        print("                TOPPINGS")
        print('---------------------------------')
        print("corn         tomato    onion     capiscum - 20 rs")
        print("mushroom     olive     onion     broccoli - 50 rs")
        print('---------------------------------')
        topping = input("Enter the topping that you want: ")
        if(topping in topping1):
            currentBill += 20
        else:
            currentBill += 50

    noOfCheese = int(input("Enter the number of Cheese type that you want: "))
    for i in range(noOfCheese):
        print('\n')
        print("          CHEESE TYPE")
        print('---------------------------------')
        print("feta       mozarella        chader - 60 rs")
        print('---------------------------------')
        cheese = input("Enter the cheese that you want: ")
#         print(cheese)
        currentBill += 60
    print('\n\n')
    print("            BILL")
    print('---------------------------------')
    print("Your pizza order was: ", currentBill)
    print('---------------------------------')
    pizzaAmounts.append(currentBill)



for i in range(orders):
    pizzaOrder()

totalBill = 0
for j in pizzaAmounts:
    totalBill += j
print('\n\n')
print("The total cost of the pizza was: ", totalBill)
```

```
Enter the number of pizza that you want to order4
Size
Small     Medium      Large
---------------------------------
50        100         200
Enter the size of the pizza: large


Enter the number of toppings that you want: 1


                TOPPINGS
---------------------------------
corn         tomato    onion     capiscum - 20 rs
mushroom     olive     onion     broccoli - 50 rs
---------------------------------
Enter the topping that you want: mushroom
Enter the number of Cheese type that you want: 1


          CHEESE TYPE
---------------------------------
feta       mozarella        chader - 60 rs
---------------------------------
Enter the cheese that you want: mozarell
```

```
           BILL
---------------------------------
Your pizza order was:  310
---------------------------------
Size
Small     Medium      Large
---------------------------------
50        100         200
Enter the size of the pizza: small


Enter the number of toppings that you want: 1


               TOPPINGS
---------------------------------
corn         tomato    onion     capiscum - 20 rs
mushroom     olive     onion     broccoli - 50 rs
---------------------------------
Enter the topping that you want: tomato
Enter the number of Cheese type that you want: 1


          CHEESE TYPE
---------------------------------
feta       mozarella        chader - 60 rs
---------------------------------
Enter the cheese that you want: feta



           BILL
---------------------------------
Your pizza order was:  130
---------------------------------
Size
Small     Medium      Large
---------------------------------
50        100         200
Enter the size of the pizza: medium


Enter the number of toppings that you want: 1


               TOPPINGS
---------------------------------
corn         tomato    onion     capiscum - 20 rs
mushroom     olive     onion     broccoli - 50 rs
---------------------------------
Enter the topping that you want: broccoli
Enter the number of Cheese type that you want: 1


          CHEESE TYPE
---------------------------------
feta       mozarella        chader - 60 rs
---------------------------------
Enter the cheese that you want: chader



           BILL
---------------------------------
Your pizza order was:  210
---------------------------------
```

```
Size
Small     Medium      Large
---------------------------------
50        100         200
Enter the size of the pizza: large


Enter the number of toppings that you want: 1


                TOPPINGS
---------------------------------
corn         tomato    onion     capiscum - 20 rs
mushroom     olive     onion     broccoli - 50 rs
---------------------------------
Enter the topping that you want: onion
Enter the number of Cheese type that you want: 1


          CHEESE TYPE
---------------------------------
feta       mozarella        chader - 60 rs
---------------------------------
Enter the cheese that you want: feta



            BILL
---------------------------------
Your pizza order was:  280
---------------------------------



The total cost of the pizza was:  930
```

In [36]:

```python
def calculate_pizza_price(size, toppings, cheese):
    cost = 0
    # Calculate pizza base price based on size
    if size == 'small':
        cost += 50
    elif size == 'medium':
        cost += 100
    else:
        cost += 200

    # Calculate topping prices
    topping_prices_20 = ['corn', 'tomato', 'onion', 'capsicum']
    topping_prices_50 = ['mushroom', 'olives', 'broccoli']

    for topping in toppings:
         cost += 20 if topping in topping_prices_20 else 50
    #Calculate cheese price
    cost += 50 * len(cheese)
    return cost

def order_pizza():
    pizzas = []
    number = int(input("How many pizzas you want to order: "))
    for i in range(number):
        toppings = []
        cheese = []
        print('Customize Pizza', i + 1)
        print('Select Pizza Size: small, medium or large')
        size = input('Select size: ')

        print('''Select Toppings:
```

```
                20 for 'corn', 'tomato', 'onion', 'capsicum'
                50 for 'mushroom', 'olives', 'broccoli' ''')
        toppings_count = int(input('How many toppings: '))

        for _ in range(toppings_count):
             toppings.append(input('Enter toppings: '))

        print('Select Cheese: mozarrella, feta, cheddar')
        cheese_count = int(input('How many cheese: '))

        for _ in range(cheese_count):
             cheese.append(input('Enter cheese: '))
        print()
        pizza_cost = calculate_pizza_price(size, toppings, cheese)
        pizzas.append((size, toppings, cheese, pizza_cost))
    return pizzas

def print_bill(pizzas):
     total = 0
     count = 1
     for pizza in pizzas:
         print('Pizza', count)
         print("Size:", pizza[0])
         print("Toppings:", pizza[1])
         print("Cheese:", pizza[2])
         print("Pizza cost:", pizza[3])
         total += pizza[3]
         count += 1
     print('Total bill amount:', total)
# Main program
pizza_order = order_pizza()
print_bill(pizza_order)
```

```
How many pizzas you want to order: 1
Customize Pizza 1
Select Pizza Size: small, medium or large
Select size: small
Select Toppings:
               20 for 'corn', 'tomato', 'onion', 'capsicum'
               50 for 'mushroom', 'olives', 'broccoli'
How many toppings: 1
Enter toppings: 2
Select Cheese: mozarrella, feta, cheddar
How many cheese: 3
Enter cheese: 1
Enter cheese: 2
Enter cheese: 3

Pizza 1
Size: small
Toppings: ['2']
Cheese: ['1', '2', '3']
Pizza cost: 250
Total bill amount: 250
```