Assignment 2 : AM624D (DS: Tools and Techniques)

Submitted By : Sidhant satapathy

Reg. No. : 21-27-06 (Data Science)

1. add of two no

In [4]:
```
a,b=input("Enter two numbers:").split(" ")#it takes as string which is separated by space
sum=int(a)+int(b)#Then it converts string to integer both the input
print("Addition of given numbers is:",sum)
```

```
Enter two numbers:23 90
Addition of given numbers is: 113
```

2.Grade of a student

In [8]:
```
ML,DL,CV,IP,DTT,BD=input("Enter marks of alll subjects respectively:").split(",")

sum=(float(ML)+float(DL)+float(CV)+float(IP)+float(DTT)+float(BD))/6
print("Avg mark is:",sum)
if sum>90:
    print("Grade:O")
elif sum>80 and sum<=90:
    print("Grade:E")
elif sum>70 and sum<=80:
    print("Grade:A")
elif sum>60 and sum<=70:
    print("Grade:B")
elif sum>50 and sum<=60:
    print("Grade:C")
elif sum>40 and sum<=50:
    print("Grade:D")
elif sum<=40:
    print("Grade:F")
```

```
Enter marks of alll subjects respectively:90,89,64,51,77,50
Avg mark is: 70.16666666666667
Grade:A
```

3.Largest among 3 numbers

In [16]:
```
a,b,c = input("Enter marks of all subjects respectively:").split(" ")
a=int(a)
b=int(b)
c=int(c)
if (a>b) and (a>c):
    print("Greatest:",a)
elif (b>c) and (b>a):
    print("greatest:",b)
else:
    print("greatest:",c)
```

```
Enter marks of all subjects respectively:33 55 10
greatest: 55
```

4.pattern

In [57]:
```
n=int(input("Enter the no of lines:"))
for i in range(n):
    print("*"*(i+1))
```

```
Enter the no of lines:5
*
**
***
```

```
****
*****
```

5.Diamond pattern

In [60]:
```
n=int(input("Enter the height of diamond:"))
for i in range(n):
    print(" "*(n-i-1)+"* "*(i+1))
for j in range(n-1):
    print(" "*(j+1)+"* "*(n-1-j))
```

```
Enter the no of lines:5
    *
   * *
  * * *
 * * * *
* * * * *
 * * * *
  * * *
   * *
    *
```

6.pattern

In [5]:
```
n=int(input("Enter the height of diamond:"))
for i in range(n):
    print("*"*(i+1))
for j in range(n-1):
    print("*"*(n-1-j))
```

```
Enter the height of diamond:3
*
**
***
**
*
```

7.list

In [19]:
```
number=[1,2,3.4,5,3.4,"asawahan"]
print(len(number))
print(number[0])
number=number+["fall of kiev"]
print(number)
number.append(89)
print(number)
print(number[-2])
number.remove(3.4)
print(number)
number.reverse()
print(number)
list=[22,12,87,78,90,55]
list.sort()
print(list)
```

```
6
1
[1, 2, 3.4, 5, 3.4, 'asawahan', 'fall of kiev']
[1, 2, 3.4, 5, 3.4, 'asawahan', 'fall of kiev', 89]
fall of kiev
[1, 2, 5, 3.4, 'asawahan', 'fall of kiev', 89]
[89, 'fall of kiev', 'asawahan', 3.4, 5, 2, 1]
[12, 22, 55, 78, 87, 90]
```

8.dict

In [2]:
```
#dict are unordered collection of data in key:value pair
#How to create dictionary
user={"name":"sidhant",
       "age":40}
print(user,type(user))#both must be in quptation like string
```

```
#second method to crate dict
user1=dict(name="java",est=1991)
print(user1)

#how to access data from dict
#there is no index in dict
print(user['name'],user1['est'])

#dict can store string,list,dictionary
user_info={
    'name':'sidhant',
    "clg":'diat',
    'fav_actress':['indian','russian']
}
print(user_info,"\n",type(user_info['fav_actress']))
# Accessing keys for the dictionary
print(user.keys())
# Accessing keys for the dictionary
print(user_info.values())
# Printing all the items of the Dictionary
print(user_info.items())
```

```
{'name': 'sidhant', 'age': 40} <class 'dict'>
{'name': 'java', 'est': 1991}
sidhant 1991
{'name': 'sidhant', 'clg': 'diat', 'fav_actress': ['indian', 'russian']}
 <class 'list'>
dict_keys(['name', 'age'])
dict_values(['sidhant', 'diat', ['indian', 'russian']])
dict_items([('name', 'sidhant'), ('clg', 'diat'), ('fav_actress', ['indian', 'russian'])])
```

10.tuple

In [27]:

```
#tuples are faster than lists
example=('mon-22','tue-19','wed-05')
#no append,no pop,no remove,no inser
#you can perform count,index,len,slicing
print(example)
print(example[:2])

#looping in tuples
mixing=(1,22,3,5,9.78)
for i in mixing:
    print(i)
print(mixing)

#tuple with one element
num=(1)#not a tuple,but integer
print(type(num))
num=(1,)#now it is atuple=so you have to add comma
wods=("stve",)#for tuple add comma
print(type(num),type(wods))

#tuple without parenthesis
guitar="bthoven","yamaha","tyler"
print(type(guitar))

#tuple unpacking
guitarist=("mangol","jamal","landscape")
g1,g2,g3=(guitarist)
print(g1,g2,g3)

#list inside tuple
favrt=(22,["radhe",'syham'])
print(favrt[1],type(favrt),type(favrt[1]))
print(favrt[1].pop(0))
```

```
('mon-22', 'tue-19', 'wed-05')
('mon-22', 'tue-19')
1
22
3
5
9.78
(1, 22, 3, 5, 9.78)
```

```
<class 'int'>
<class 'tuple'> <class 'tuple'>
<class 'tuple'>
mangol jamal landscape
['radhe', 'syham'] <class 'tuple'> <class 'list'>
radhe
```

11.median

In [32]:
```
list = [4, 5, 9, 10, 14, 17]
list.sort()
mid = len(list) // 2
res = (list[mid] + list[~mid]) / 2
print("Median of the list is:",res)
```

```
Median of the list is: 9.5
```

12.mode

In [42]:
```
import statistics as st1
statistics.mode([4, 1, 2, 2, 3, 5])
```

Out[42]:
```
2
```

13.mean

In [44]:
```
import statistics
statistics.mean([4, 8, 6, 5, 3, 2, 8, 9, 2, 5])
```

Out[44]:
```
5.2
```

14.nested_list

In [4]:
```
list_1 = [1,3,92,75.4,'millions',21]
print(list_1)
list_2 = [22,132,190]
list_1.append(list_2)
print(f"Appending an entire list to another List : {list_1}")#Nested list
list_1 = list_1 + list_2
print("Adding Two lists : ",list_1)
```

```
[1, 3, 92, 75.4, 'millions', 21]
Appending an entire list to another List : [1, 3, 92, 75.4, 'millions', 21, [22, 132, 190]]
Adding Two lists :  [1, 3, 92, 75.4, 'millions', 21, [22, 132, 190], 22, 132, 190]
```

15.slicing

In [9]:
```
list = [2,1,19,12,16,25,8,23]
#LIST SLICING :  List[ Initial : End : IndexJump ]
print(list[1:])
print(list[::])
print(list[:8:2])
```

```
[1, 19, 12, 16, 25, 8, 23]
[2, 1, 19, 12, 16, 25, 8, 23]
[2, 19, 16, 8]
```

16.min & max of list

In [13]:
```
list1 = [2,1,19,12,16,25,8,23]

print(min(list1))
print(max(list1))
```

```
1
25
```

17.count or occurence of list

In [23]:
```
list_count = [0,56.90,66,14,66,0,19,56,32,74,0,0,66,0]
a= int(input("Enter the number : "))
print(list_count.count(a))
```

```
Enter the number : 0
5
```

18.2d_list

In [27]:
```
list_2D = [[1,2,3,11,12,13], [5,15,25,35], [10,6,2]]
print("2D list :",list_2D)
for i in list_2D:
    print(i)
```

```
2D list : [[1, 2, 3, 11, 12, 13], [5, 15, 25, 35], [10, 6, 2]]
[1, 2, 3, 11, 12, 13]
[5, 15, 25, 35]
[10, 6, 2]
```

19.2d list

In [33]:
```
td_list = [[1,2,3,11,12,13], [5,15,25,35], [10,6,2]]

# Method 1
for i in range(len(td_list)):
    for j in range(len(td_list[i])):
        print(td_list[i][j], end=' ')
    print()

print("\n")

# Method 2
for i in td_list:
    for j in i :
        print(j, end = " ")
    print()
```

```
1 2 3 11 12 13
5 15 25 35
10 6 2


1 2 3 11 12 13
5 15 25 35
10 6 2
```

20.spilt & join in list

In [35]:
```
#1 Splitting a list
list = [12,19,21,29]
print('List : ',list)
print('After Splitting:')
middle_index = int(len(list)/2)
first_half = list[:middle_index]
second_half = list[middle_index:]
print('first half : ',first_half)
print('second half : ',second_half)

#2 Joining
print("Joining Operation : ")
list1 = ["Hello", "World" ]
list2 = [1, 2, 3]
print('List1 :',list1)
print('List2 :',list2)
list3 = list1 + list2
print(list3)
```

```
List :  [12, 19, 21, 29]
After Splitting:
first half :  [12, 19]
```

```
second half :  [21, 29]
Joining Operation :
List1 : ['Hello', 'World']
List2 : [1, 2, 3]
['Hello', 'World', 1, 2, 3]
```