

```
In [1]: print("Hello! Sonu")
```

Hello! Sonu

```
In [1]: a=input("enter 1st number:")  
b=input("enter 2nd number:")  
c=input("enter 3rd number:")  
if(a==b)and(b==c):  
    print("All are equal.")  
elif(a>b)and(a>c):  
    print("1st number is maximum.")  
elif(b>c)and(b>a):  
    print("2nd number is maximum.")  
else:  
    print("3rd number is maximum.")  
if(a==b)and(b==c):  
    print("All are equal.")  
elif(a<b)and(a<c):  
    print("1st number is minimum.")  
elif(b<a)and(b<c):  
    print("2nd number is minimum.")  
else:  
    print("3rd number is minimum.")
```

```
enter 1st number:20  
enter 2nd number:30  
enter 3rd number:45  
3rd number is maximum.  
1st number is minimum.
```

```
In [26]: import math  
a=input("enter 1st side:")  
b=input("enter 2nd side:")  
print("hypoteneous of the triangle is:")  
print(math.sqrt(int(a)**2 + int(b)**2))
```

```
enter 1st side:12  
enter 2nd side:5  
hypoteneous of the triangle is:  
13.0
```

```
In [27]: p=input("enter principle:")  
r=input("enter rate of interest:")  
n=input("enter number of year:")  
print((int(p)*int(r)*int(n))/100)
```

```
enter principle:1000  
enter rate of interest:5  
enter number of year:2  
100.0
```

```
In [31]: r=input("radius of circle:")  
print("area of circle is:")  
print((math.pi) * int(r)**2)  
print("circumference of circle is:")  
print(2*math.pi*int(r))
```

```
radius of circle:5
area of circle is:
78.53981633974483
circumference of circle is:
31.41592653589793
```

In [39]:

```
a=input("enter 1st side:")
b=input("enter 2nd side:")
c=input("enter 3rd side:")
if (a==b==c):
    print("equilateral.")
elif((a==b)and(b!=c)) or ((a==c)and(c!=b))or((b==c)and(a!=b)):
    print("isosceles.")
elif((a!=b)and(b!=c)) or ((a!=c)and(c!=b))or((b!=c)and(a!=b)):
    print("scalen.")
```

```
enter 1st side:12
enter 2nd side:23
enter 3rd side:12
isosceles.
```

In [74]:

```
a=input("enter 1st number:")
b=input("enter 2nd number:")
c=int(a)+int(b)
print(" addition of %s and %s is : %s"%(a,b,c))
```

```
enter 1st number:20
enter 2nd number:30
addition of 20 and 30 is : 50
```

In [2]:

```
a=20
b=10
c=a+b
print("addition of a:{0} and b:{1} is c:{2}".format(a,b,c))
```

```
addition of a:20 and b:10 is c:30
```

In [3]:

```
max(10,20,30,40,12)
```

Out[3]: 40

In [37]:

```
name="bskjbfk"
name.isalnum()
```

Out[37]: True

In [44]:

```
a=23
b=56
c=a+b
print("additon of %d and %d is: %d"%(a,b,c))
str=("addition of {2} and {1} is: {0}")
print(str.format(a,b,c))
```

```
additon of 23 and 56 is: 79
addition of 79 and 56 is: 23
```

```
In [45]: name="""sonu  
monu  
sahdasf  
fasfas"""  
len(name)
```

```
Out[45]: 24
```

```
In [55]: array=[12,52,1,2,20,30]  
print(len(array))
```

```
6
```

```
In [54]: str=" fj kh "  
print(str.strip())  
len(str)
```

```
fj kh
```

```
Out[54]: 7
```

```
In [56]: array=[12,52,1,2,20,30]  
len(array)
```

```
Out[56]: 6
```

```
In [76]: str=" fj kh "  
str.strip()  
print(str)
```

```
fj kh
```

```
In [11]: a=input("enter the first number:")  
b=input("enter the first number:")  
ch=input("Enter the choice:\n 1 for addition\n 2 for subtraction\n 3 for multiplication  
 4 for division  
 5 for power")  
if(int(ch)==1):  
    print(int(a)+int(b))  
elif(ch==2):  
    print(int(a)-int(b))  
elif(ch==3):  
    print(int(a)*int(b))  
elif(ch==4):  
    print(int(a)/int(b))  
elif(ch==5):  
    print(a**b)  
else:  
    print("invalid choice.")
```

```
enter the first number:10  
enter the first number:50
```

```
Enter the choice:  
 1 for addition  
 2 for subtraction  
 3 for multiplication  
 4 for division  
 5 for power
```

```
2  
invalid choice.
```

In [34]:

```
import math  
x=input("enter the first number:")  
y=input("enter the second number")  
print(math.pi)  
print(math.e)  
print(math.ceil(int(x)))  
print(math.floor(int(x)))  
print(math.factorial(int(x)))  
print(math.pow(int(x),int(y)))  
print(math.fabs(int(x)))  
print(math.log2(int(x)))  
print(math.sqrt(int(x)))  
print(math.exp(int(x)))  
print(math.radians(int(x)))  
print(math.degrees(int(x)))  
print(math.copysign(int(x),int(y)))  
print(math.cos(int(x)))  
print(math.sin(int(x)))  
print(math.tan(int(x)))
```

```
enter the first number:45  
enter the second number2  
3.141592653589793  
2.718281828459045  
45  
45  
1196222208654801945619631614956577150643837337600000000000  
2025.0  
45.0  
5.491853096329675  
6.708203932499369  
3.4934271057485095e+19  
0.7853981633974483  
2578.3100780887044  
45.0  
0.5253219888177297  
0.8509035245341184  
1.6197751905438615
```

In [3]:

```
a="sonu yadav"  
for i in a:  
    print(i)
```

```
s  
o  
n  
u  
  
y  
a  
d  
a  
v
```

In [26]:

```
a="sonu yadav"  
i=0  
while(i<len(a)):
```

```
print(a[i])
i=i+1
```

s
o
n
u
.
y
a
d
a
v

In [37]:

```
import random
print(random.randint(10,50))
```

32

In [77]:

```
import random
a="sonu monu asu"
print(random.choice(a))
```

m

In [42]:

```
a=[1,50,3,4,2,30]
print(random.choice(a))
```

30

In [43]:

```
print(random.random())
#it will print random number between 0 and 1
```

0.831264436718998

In [44]:

```
print(random.randrange(10))
#it will genreate random number between 0 and 9
```

5

In [62]:

```
print(random.randrange(10,101,2))
#it will print even number between 10 and 100
```

14

In [99]:

```
#reverse the given string
a="sonu monu"
str=""
i=len(a)-1
while i>=0:
    str= str + a[i]
    i=i-1
print(str)
```

unom unos

In [103...]

```
#print the even characters after reversing
a="sonu yadav"
str=""
i=len(a)-1
if i%2!=0:
    while i>=0:
        str=str + a[i]
        #print(a[i])
        i=i-2
    print(str)
else:
    while i>=0:
        str=str + a[i]
        #print(a[i-1])
        i=i-2
    print(str)
```

vdyuo

In [106]:

```
#print the odd characters after reversing
a="sonu yadav"
str=""
i=len(a)-1
if i%2==0:
    while i>=0:
        str=str + a[i]
        i=i-2
    print(str)
else:
    while i>=0:
        str=str + a[i]
        i=i-2
    print(str)
```

vdyuo

In [8]:

```
name="this is a dog in zoo"
name[4:12:3]
```

Out[8]:

' d'

In [6]:

```
var=[1,5,23,7,9]
son=["a","f","fr","dsg"]
list=[var,son]
print(list[0][3])
print(list)
```

7

[[1, 5, 23, 7, 9], ['a', 'f', 'fr', 'dsg']]

In [18]:

```
var.append(258)
```

In [19]:

```
var
```

Out[19]: [1, 5, 23, 7, 9, 258]

```
In [20]:  
for i in range(10,23):  
    var.append(i)
```

```
In [21]:  
var
```

```
Out[21]: [1, 5, 23, 7, 9, 258, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]
```

```
In [22]:  
var.insert(5,10)
```

```
In [23]:  
var
```

```
Out[23]: [1, 5, 23, 7, 9, 10, 258, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]
```

```
In [26]:  
var.pop()
```

```
Out[26]: 10
```

```
In [27]:  
var
```

```
Out[27]: [1, 5, 7, 9, 258]
```

```
In [28]:  
var.pop()
```

```
Out[28]: 258
```

```
In [29]:  
var.pop(2)
```

```
Out[29]: 7
```

```
In [30]:  
var
```

```
Out[30]: [1, 5, 9]
```

```
In [78]:  
a={'phon':8340406282,'name':'sonu','address':'bihar',101:'id'}  
a['phon']
```

```
Out[78]: 8340406282
```

```
In [79]:  
a.keys()
```

```
Out[79]: dict_keys(['phon', 'name', 'address', 101])
```

```
In [80]:  
a.items()
```

```
Out[80]: dict_items([('phon', 8340406282), ('name', 'sonu'), ('address', 'bihar'), (101, 'id')])
```

```
In [81]: a.values()
```

```
Out[81]: dict_values([8340406282, 'sonu', 'bihar', 'id'])
```

```
In [9]: a[101]
```

```
Out[9]: 'id'
```

```
In [11]: for var in a:  
         print(a[var])
```

```
8340406282  
sonu  
bihar  
id
```

```
In [12]: for var in a:  
         print(var)
```

```
phon  
name  
address  
101
```

```
In [14]: for var in a:  
         print(a.items())
```

```
dict_items([('phon', 8340406282), ('name', 'sonu'), ('address', 'bihar'), (101, 'id')])  
dict_items([('phon', 8340406282), ('name', 'sonu'), ('address', 'bihar'), (101, 'id')])  
dict_items([('phon', 8340406282), ('name', 'sonu'), ('address', 'bihar'), (101, 'id')])  
dict_items([('phon', 8340406282), ('name', 'sonu'), ('address', 'bihar'), (101, 'id')])
```

```
In [16]: len(a)
```

```
Out[16]: 4
```

```
In [17]: choice ={'s1':'maths', 's2':'phy', 's3':'chem', 's4':{1: 'c', 2:'java',3:'c++'}}
```

```
In [18]: len(choice)
```

```
Out[18]: 4
```

```
In [19]: choice['s4'][3]
```

```
Out[19]: 'c++'
```

```
In [20]: len(choice['s4'])
```

```
Out[20]: 3
```

```
In [21]: len(choice) + len(choice['s4'])
```

```
Out[21]: 7
```

```
In [22]: choice['s5']='art'
```

```
In [23]: choice
```

```
Out[23]: {'s1': 'maths',
           's2': 'phy',
           's3': 'chem',
           's4': {1: 'c', 2: 'java', 3: 'c++'},
           's5': 'art'}
```

```
In [24]: choice.update({'s6': 'PE'})
```

```
In [25]: choice
```

```
Out[25]: {'s1': 'maths',
           's2': 'phy',
           's3': 'chem',
           's4': {1: 'c', 2: 'java', 3: 'c++'},
           's5': 'art',
           's6': 'PE'}
```

```
In [31]: dummy={1: '1', 2: '2'}
```

```
In [32]: choice.update(dummy)
```

```
In [33]: choice
```

```
Out[33]: {'s1': 'maths',
           's2': 'phy',
           's3': 'chem',
           's4': {1: 'c', 2: 'java', 3: 'c++'},
           's5': 'art',
           's6': 'PE',
           1: '1',
           2: '2'}
```

```
In [34]: choice.pop('s4')
```

```
Out[34]: {1: 'c', 2: 'java', 3: 'c++'}
```

```
In [36]: choice.clear()
```

In [37]: choice

Out[37]: {}

In [31]:

```
#1. Count the occurrence of an element from the given list.
import random
list=[10, 20, 5, 10, 8, 10, 3, 0.3, 10, 20, 8, 19, 18, 15, 2, 1, 3, 2, 13, 5, 0, 12, 0, 11, 0, 20]
#a=input("enter the number:")
a=random.randint(0,20)
print(a)
c=list.count(a)
print(c)
```

3

2

In [51]:

```
#2. Print all the sub-lists from a given list.
```

T

In [52]:

```
choice ={'s1':'maths', 's2':'phy', 's3':'chem', 's4':{1: 'c', 2:'java',3:'c++'}}
for x,y in choice.items():
    print(x,y)
```

s1 maths
s2 phy
s3 chem
s4 {1: 'c', 2: 'java', 3: 'c++'}

In [62]:

```
a=10
id(a)
```

Out[62]: 140732138334288

In [63]:

```
b=20
id(b)
```

Out[63]: 140732138334608

In [61]:

```
a is not b
```

Out[61]: True

In [8]:

```
A = ['blah','blahm','blahn']
B = ['1','2','3']
for i in A:
    for j in B:
        print(i,j)
```

blah 1
blah 2
blah 3

```
blahm 1  
blahm 2  
blahm 3  
blahn 1  
blahn 2  
blahn 3
```

In [68]:

```
sum=0  
for i in range(0,101):  
    if i%2==0:  
        sum=sum + i  
print(sum)
```

2550

In [78]:

```
sum=0  
n=input("entr the number:")  
for i in range(1,int(n) + 1):  
    sum=sum + 1/i  
sum
```

entr the number:5

Out[78]: 2.283333333333333

In [79]:

```
sum=0  
n=input("enter the number:")  
for i in range(1,int(n) + 1):  
    sum=sum + 1/(i*i)  
sum
```

enter the number:3

Out[79]: 1.3611111111111112

In [3]:

```
s=0  
n=input("enter the decimal number:")  
while n != 0:  
    x=int(n)%2  
    s=s*10  
    s=s+x  
    n=int(n)/2  
print(s)
```

enter the decimal number:8

10

In [4]:

```
a=('a',10,'sf',52,22)
```

In [5]:

```
a
```

Out[5]: ('a', 10, 'sf', 52, 22)

In [9]:

```
b=['s','dfs',20,56]
```

```
In [10]: b.insert(2,10)
```

```
In [11]: b
```

```
Out[11]: ['s', 'dfs', 10, 20, 56]
```

```
In [12]: b.append(123)
```

```
In [21]: x='sfasjfnao'  
y='mkonj'  
if(x==y):  
    print("true")  
print("false")
```

```
false
```

```
In [ ]:
```

```
In [84]: import random  
print("#Start the rock,paper and cissor game#")  
a=input("Enter 1st player name:")  
r='rock'  
p='paper'  
c='cissor'  
count=0;  
b=input("Choose your choise->rock or paper or cissor: ")  
i=random.randint(0,3)  
if(i==0):  
    d='rock'  
elif(i==1):  
    d='paper'  
else:  
    d='cissor'  
print("Computer choosen choice is:%s"%(d))  
if(d==b):  
    print("Draw")  
elif(b==r and d==c):  
    print("winner is: %s"%(a))  
elif(b==r and d==p):  
    print("winner is: Computer")  
elif(b==p and d==c):  
    print("winner is: Computer")  
elif(b==p and d==r):  
    print("winner is: %s"%(a))  
elif(b==c and d==p):  
    print("winner is: %s"%(a))  
else:  
    print("winner is: Computer")
```

```
#Start the rock,paper and cissor game#  
Enter 1st player name:sonu  
Choose your choise->rock or paper or cissor: rock
```

Computer choosen choice is:rock
Draw

```
In [45]: x=(1,2,3,4,4,5,1,2,6,3,6,6,1)
y=list(x)
for i in y:
    print(i, " -> ",y.count(i))
    y.remove(i)
```

```
-----  
TypeError                                     Traceback (most recent call last)
<ipython-input-45-84037c03ca4e> in <module>
      1 x=(1,2,3,4,4,5,1,2,6,3,6,6,1)
----> 2 y = list(x)
      3 for i in y:
      4     print(i, " -> ",y.count(i))
      5     y.remove(i)

TypeError: 'list' object is not callable
```

```
In [8]: choice ={'s1':'maths', 's2':'phy', 's3':'chem', 's4':{1: 'c', 2:'java',3:'c++'}}
key=input("enter the key which value you want in the dictionary:")
for i in choice:
    if key==i:
        print("value of the given key is : %s"%(choice[i]))
```

```
enter the key which value you want in the dictionary:s4
value of the given key is : {1: 'c', 2: 'java', 3: 'c++'}
```

```
In [7]: x=[1,2,85,6,[5,36,2,14],5,3,[5,2,89,52]]
for i in range(0,len(x)):
    if(type(x[i])==list):
        print(x[i])
```

```
[5, 36, 2, 14]
[5, 2, 89, 52]
```

```
In [86]: a=["abd","bhde"]
x=""
for i in range(0,len(a)):
    print(a[i][::-1],end=" ")
```

```
dba edhb
```

```
In [49]: a="sdjabhf"
x=""
for i in range(len(a)-1,-1,-1):
    print(a[i],end="")
```

```
fhbajds
```

```
In [120... a=["abd","bhde"]
x=""
for i in range(0,len(a)):
    y=a[i]
```

```

for j in range(len(y[i])-1,-1,-1):
    x=x+a[j]
print(x)

```

```

IndexError                                     Traceback (most recent call last)
<ipython-input-120-335ac1449bf3> in <module>
      4     y=a[i]
      5     for j in range(len(y)-1,-1,-1):
----> 6         x=x+a[j]
      7     print(x)
      8

IndexError: list index out of range

```

In [59]:

```

a=['a','v','S']
for i in range(0,len(a)):
    print((a[i]).upper())

```

A
V
S

In [75]:

```

a={'a':1,'b':2,'c':2,'d':5,'e':7,'f':1,'g':10,'h':11,'i':5,'j':12,'k':9,'l':11,'m':7,'n':0}
m=0
n=0
print("1st player->")
x=input("Enter the word:")
print("2nd player->")
y=input("Enter the word:")
for i in range(0,len(x)):
    m=m+a[x[i]]
for j in range(0,len(y)):
    n=n+a[y[j]]
if(m>n):
    print("1st player win.")
elif(m<n):
    print("2nd player win.")
else:
    print("Both player have same score.")

```

1st player->
Enter the word:and
2nd player->
Enter the word:mjs
2nd player win.

In [79]:

```

import array as a
arr=a.array("i",[2,3,45,23,10,])
for i in arr:
    print(i)

```

2
3
45
23
10

In [84]:

```
import array as a
arr=a.array("i",[2,3,45,23,10,])
arr.insert(2,15)
for i in arr:
    print(i)
```

```
2
3
15
45
23
10
```

In [12]:

```
import array as b
arr=b.array("i",[2,3,45,23,10,])
arr.insert(2,15)
arr.remove(3)
for i in arr:
    print(i)
```

```
2
15
45
23
10
```

In [91]:

```
arr.pop()
for i in arr:
    print(i)
```

```
2
15
45
23
```

In [93]:

```
arr.pop(3)
for i in arr:
    print(i)
```

```
2
15
45
```

In [94]:

```
arr.append(16)
```

```
2
15
45
16
```

In [100...]

```
print(arr.count(45))
```

```
1
```

```
In [15]: print(arr.reverse())
```

None

```
In [102... for i in arr:  
    print(i)
```

16
45
15
2

```
In [103... for i in arr:  
    print(5*i)
```

80
225
75
10

```
In [117... import array as a  
arr=a.array("i",[2,3,45,23,10,])  
x=a.array("i",[])  
y=a.array("i",[])  
for i in arr:  
    print(i)  
a=input("enter the pivot index:")  
for i in range(0,int(a)):  
    x.append(arr[i])  
for j in range(int(a)+1,len(arr)):  
    y.append(arr[j])  
for i in x:  
    print(i)  
for j in y:  
    print(j)
```

2
3
45
23
10
enter the pivot index:2
2
3
23
10

```
In [114... import array as a  
arr=a.array("i",[2,3,45,23,10,])  
for i in range(0,len(arr)-1):  
    for j in range(0,len(arr)-1-i):  
        if arr[j]>arr[j+1]:  
            arr[j]=arr[j]+arr[j+1]  
            arr[j+1]=arr[j]-arr[j+1]  
            arr[j]=arr[j]-arr[j+1]  
  
for i in arr:  
    print(i)
```

```
2
3
10
23
45
```

In [1]:

```
def area_of_triangle(l,b):
    return l*b
area_of_triangle(10,20)
```

Out[1]: 200

In [6]:

```
import math
def area_of_circle(r):
    return math.pi*r**2
def area_of_cylinder(h,r):
    return area_of_circle(r)*2 + 2*math.pi*r*h
area_of_cylinder(10,5)
```

Out[6]: 471.23889803846896

In [22]:

```
list=[10,20,40,30,50,10,6,20,0]
def manipulate(s,x):
    if(s=='append'):
        return list.append(int(x))
    elif(s=='remove'):
        return list.remove(int(x))
    elif(s=='pop'):
        return list.pop(int(x))

s=input("Enter the choice: append,remove,pop:")
x=input("Enter the index or append value:")
manipulate(s,x)
list
```

Enter the choice: append,remove,pop:remove
Enter the index or append value:50

Out[22]: [10, 20, 40, 30, 10, 6, 20, 0]

In [1]:

```
pi=3.142
def fun(r):
    return pi*r**2
fun(int(input("Enter the radius of the circle:")))
```

Enter the radius of the circle:23

Out[1]: 1662.118

In [7]:

```
def sum(a,*b):
    c=a
    for i in range(0,len(b)):
        c=c+b[i]
    print(c)
sum(10,20,30,40,50,20,10,60,100)
```

340

In [11]:

```
class student:
    def __init__(self, id, name, **x):
        average=0
        for i in range(0, len(x)):
            self.id = id
            self.name = name
            self.x[i] = x[i]
            average = average + x[i]
    def compute_average(self):
        print("average marks of: " + self.name + " is " + str(average))
```

In [12]:

```
obj = student("15", "sonu", phy="12", che="56", math="50")
obj.compute_average()
```

KeyError

Traceback (most recent call last)

```
<ipython-input-12-0fdbf580487b> in <module>
----> 1 obj = student("15", "sonu", phy="12", che="56", math="50")
      2 obj.compute_average()
```

```
<ipython-input-11-14e39b24eb90> in __init__(self, id, name, **x)
      5         self.id = id
      6         self.name = name
----> 7         self.x[i] = x[i]
      8         average = average + x[i]
      9     def compute_average(self):
```

KeyError: 0

In [3]:

```
class monu:
    pass
```

In [41]:

```
#r=read the file
#w=write the file
#a=append the file
#x=create the file
f=open("C:/Users/PRADEEP COMPUTER/Desktop/dum.txt", "r")
#print(f.read())
print(f.readlines())
```

```
['asdf\n', 'sdf\n', 'dfg\n', 'dsfg\n', 'dsf ']
```

In [19]:

```
f=open("C:/Users/PRADEEP COMPUTER/Desktop/dum.txt", "r")
#f.read()
count=0
for i in f:
    count=count + 1
```

In [33]:

```
print(count)
```

5

```
In [68]: f=open("C:/Users/PRADEEP COMPUTER/Desktop/dum.txt","r")
count_vowels=0
count_consonents=0
count_digit=0
for i in f:
    for j in i:
        if j=='a' or j=='e' or j=='i' or j=='o' or j=='u' :
            count_vowels=count_vowels + 1
        elif j=='0' or j=='1' or j=='2' or j=='3' or j=='4' or j=='5' or j=='6' or j=='7' or j=='8' or j=='9':
            count_digit=count_digit + 1
        else:
            count_consonents=count_consonents + 1
```

```
In [69]: print(count_vowels)
print(count_consonents)
print(count_digit)
```

15
42
34

```
In [77]: def substring (s,n):
    count = 0
    count1=0
    for j in range(n):
        if s[j]=='a' or s[j]=='e' or s[j]=='i' or s[j]=='o' or s[j]=='u' :
            for i in range(j+1,n+1):
                x=s[j:i]
                count=count+1
        else:
            for i in range(j+1,n+1):
                y=s[j:i]
                count1=count1 + 1
    print(p1+" score is %d"%count)
    print(p2+" score is %d"%count1)
    if count>count1 :
        print(p1+" is winner.")
    else:
        print(p2+" is winner.")

s=input("enter the string:")
p1=input("enter the 1st player name:")
p2=input("enter the 2nd player name:")
substring(s,len(s))
```

enter the string:banana
enter the 1st player name:sonu
enter the 2nd player name:monu
sonu score is 9
monu score is 12
monu is winner.

```
In [177... stack=[]
def fun(x,y):

    s=""
    stack=stack.append(s)
    s1=""
    s2=""
```

```

if int(x)==1:
    s=s+y
    print("update string is:",s)
    #stack=stack.append(s)
elif int(x)==2:
    for j in range(len(s)-int(y),len(s)):
        s1=s1+s[j]
    #chars to delete
    for i in range(0,len(s) - int(y)):
        s2=s2+s[i] #krrp
    s=s2
    stack=stack.append(s)
elif int(x)==3:
    print(s)
elif int(x)==4:
    #s=s+[stack.pop()]
    print("undo",stack.pop())
    #s = stack.peek()
    print(s)
l=input("print the number of times you want to perform the operations:") #5
for i in range(0,int(l)):
    m=input("1 for append, 2 for delete , 3 for print , 4 for undo:")
    n=input("enter the value if you want to append or enter the number of elements to d
fun(m,n)

```

print the number of times you want to perform the operations:3
 1 for append, 2 for delete , 3 for print , 4 for undo:1
 enter the value if you want to append or enter the number of elements to delete:sfdad

```

UnboundLocalError                                     Traceback (most recent call last)
<ipython-input-177-4b74d0d1c31f> in <module>
      30     m=input("1 for append, 2 for delete , 3 for print , 4 for undo:")
      31     n=input("enter the value if you want to append or enter the number of eleme
nts to delete:")
--> 32     fun(m,n)

<ipython-input-177-4b74d0d1c31f> in fun(x, y)
      3
      4     s=[]
--> 5     stack=stack.append(s)
      6     s1=[]
      7     s2=[]

```

UnboundLocalError: local variable 'stack' referenced before assignment

In [130...]

```
f=open("rahul.txt","w")
f.write("sfbasjbf")
f.close()
```

In [131...]

```
import os
os.remove("C:/Users/PRADEEP COMPUTER/Desktop/sanu1.txt")
```

In [158...]

```
"""1. Write a function in python to count the number of lines from a text
file which is not starting with an alphabet "T". split(" ")"""

count=0
f=open("C:/Users/PRADEEP COMPUTER/Desktop/x.txt","r")
s=f.readlines()
```

```

for i in s:
    if not (i[0]=="t" or i[0]=="T"):
        count=count + 1
print("total number of line which is not starting with t is : ")
print(count)

```

total number of line which is not starting with t is :
5

In [176]:
s="bfjkasdhfilhds"
print(s[::-1])

sdhlfhdsakjfb

In [183]:
print(s[::-2])

shihskf

In [122]:
f=open("C:/Users/PRADEEP COMPUTER/Desktop/sanu.txt","r")
count=0
var=f.read()
v = var.split()
for i in v:
 if len(i)!=0:
 if i[-1]=='e':
 count=count+1
print(count)

3

In [56]:
a="dsjkdf asfaiu dahasuif"
print(len(a))
print(a[len(a)-1])
print(a[::-1])
print(a[:-1])
str=a.split(" ")
for i in str:
 print(i)

22
f
fiusahad uiafsa fdkjsd
dsjkdf asfaiu dahasuif
dsjkdf
asfaiu
dahasuif

In [167]:
n=int(input("enter the number:"))
var=97
dic={}
for i in range(1,n+1):
 dic[i]=chr(var)
 var=var+1
print(dic.items())

enter the number:2
dict_items([(1, 'a'), (2, 'b')])

In [178...]

```

list1=[10,12,15,46,89,123]
start=0
flag=1
end=len(list1)-1
key=int(input("enter the key:"))
for i in range(0,len(list1)):
    mid=int((start+end)/2)
    if key==list1[mid]:
        print("given key's index is:",mid)
        flag=0
        break
    elif key>list1[mid]:
        start=mid+1
    elif key < list1[mid]:
        end=mid-1

if(flag==1):
    print("given key is not present in the list.")

```

enter the key:2
given key is not present in the list.

In [179...]

```

def appen(n):
    stack.append(n)
stack=[]
def fun(x,y):
    if int(m)==1:
        appen(n)

l=input("print the number of times you want to perform the operations:") #5
for i in range(0,int(l)):
    m=input("1 for append, 2 for delete , 3 for print , 4 for undo:")
    n=input("enter the value if you want to append or enter the number of elements to d
fun(m,n)

```

```

KeyboardInterrupt                                     Traceback (most recent call last)
<ipython-input-179-a82a964ebb43> in <module>
      6     appen(n)
      7
----> 8 l=input("print the number of times you want to perform the operations:") #5
      9 for i in range(0,int(l)):
     10     m=input("1 for append, 2 for delete , 3 for print , 4 for undo:")

~\anaconda3\lib\site-packages\ipykernel\kernelbase.py in raw_input(self, prompt)
    858             "raw_input was called, but this frontend does not support input
requests."
    859         )
--> 860         return self._input_request(str(prompt),
    861             self._parent_ident,
    862             self._parent_header,

~\anaconda3\lib\site-packages\ipykernel\kernelbase.py in _input_request(self, prompt, id
ent, parent, password)
    902             except KeyboardInterrupt:
    903                 # re-raise KeyboardInterrupt, to truncate traceback
--> 904                 raise KeyboardInterrupt("Interrupted by user") from None
    905             except Exception as e:
    906                 self.log.warning("Invalid Message:", exc_info=True)

```

KeyboardInterrupt: Interrupted by user

In [191..

```
b2\xeeD\xc1vW\xdf\x10\x9fZ\x19v\x89\xf5\xd4\xee\xb3R\xe1\xe2p\xd4\x8493\x17\x10\xe2H\xb6
\xb3{\xe6\x15b\xae,\x16\xd3\x95\xab\xd1%Po\x8b\xcdK\x12\xc5j\xb8\xd5\xbf\x9c\y\x98\xe9
q\xea\x15*\x04\xe7R{^#Q\xc1}\xb8"\xdb\xf63\x1f\xfe\x1b\xa2\x182\x96R?\xcd5\xfe\xa6;Z\xad
jz`\x86\x06\xe2y\xfe!\xc4\x970\xf3\xa3\x03\x9d\xaf5\xb7\x0b`\x9c\x86\xab\x11\xb6f*\x9c\x
05\\\'x10\xd5\x18\x83\x92%\x98\xd4\xf4\xe0\xxa7\xc5\xaa\x9a\xb6Fam\xaa3|gDU\xf4\xea\x90/\x
80o\x1d\xaa8\xf8;H\xecC\x03\xcd\xdes5\x7f\x04S\xadP\x1ec\xef:o\x14\xba\xe9i\xfe)\xa6\xad~
a\xc9\xef:4\xbe\xa1U\x0bs\x01\x89\xc8\xd2\xfc\x1dJ\x9a\x04f]&\x7f\x84\x91E\xa9\xd5a\x17
\xc3\xf0\x9e\xdd}7\x1a\xa7R\xaa7\xca\xe0\xe6u\xb5<G\xff\x00(\xc7\x9a\xf8\xbc\xf1T~\x1d\x
a\x6pV\xaa3\xbc\xeaS\xd0j*/+;\x05\xdb\xdef\xceW\x1e;\xe2^8\xf4\xeb5\xae\xb9\xbc\xf3\xab
\xc7k\xd4\x06\xf9\x9f0\xab\xf0\x96\x9bRo\xaa9R\xde\xf2\xd3\xe0\x9e\x18\x82\xe2\x85\xcc
\xdf\xf2q\x1a\xf2\xb9\x8f\x99\x8e)X0f\xab\x7f\xac\xe9\xd0\xe2\x1f\x88P\x19\xff\x00y\xee
\xe9\xfc\x1f\xc3\xb9\x875\x00G\x99\xd3\xd3/|\xc2)\x81m:\x03\xed\x17\xe5\xe5\x8f0\x01F\x8
d:\x86\xec\xc0\xce\xc6\x83\x85\xd2\xaaT\xdc\x01\xed=\xd6\x9f\x83p\xda`\x11I\x07\xb0\x9bE
\r\x05@\'\xa6\xa0\xceW\xe4\xd6\xbf\xfc<\xf7\x0f\xe1:z\x1b\xad\xe7oM\xaa5\xd3\xdf,\xb6\xf0
&\x84\xfc\x15\xed\xc8\xb6\xf3\xe?\'\n\x06\x02\x819\xdbh\x89J\x80Qb-\x1a\x81:0\xb4\x14}=\'x
f6\G\xaa3\xd0*2&t\xc6\xbe\x1e\xe8j\x0b71\x13\xb4*\x8eP\x04\xe0\xd2\xafB\x9f\xe5}\a\xd4\x
e2t\xd11\x1c\x137\xf6\xe7c\xb0\xd5\xc2\x8d\xe6w\xd5(;\'\xc5\xaa\xe3!I,\xd8\x13\x93\xaa\xf
8\x9a\x8a\xdc\x07\x04\xf4\xcc\xb3\x9d1\xeb\xeb\xeb\xd01%\xb0\`\'\x89q@t\xce\xca\xd6^\'\xf3
\xe7\{\xe3\x01G\xd4TbZ\xd6\xbf\xe7u\xbf\x1c\xb1\xd0\x8aDr\x99\xd2|W\x1b\xe6K_I\xe1z\x9
5z\xc0\xde\xe5\x8c\xf5\xfaZ\xaa1T\x02w\x9f\x0f\xf8[\xe2%\xd4j\xe8\xaa9{-\xf2L\xfa\xce\x9bX
\x8dMY\x9b\x16\x98\xeb\x9b\x0f\x92e\xf4\xeaW\xd4\x015\x94\x1c\x19\x9e\xbdnV\x173\x9b\xaa
\xd4\x86\xd4\x82\x0e\x01\x88\xd4\xea\x836\xf6\x11\x8egk\xab^\'\xa2\xb6\xc2]\rB\xf2\x92\xc7
\xe6\x02yn#\xf1\x05\x04w^pB`\x1b\xeeg\x16\x97\xc5h\xd5j\xd3\x0e.\x08\xcd\xf0&\x7\xc7o\x
e1q\xf4*\xda\xb0)\x8f5\xed\xb4\xc4\xba\x80W\xe68\xf3<\xd7\xf7\xf5#N\xc1\xb7\xf30j\xfe#Zh
\x148\x1dL\xb3\x8ac\xd4\xebub\x9a\x1f\x9a\xc4\x0b\xceF\x97\x8e\xab\xf3#\x91u\xeb\xe1\xf
5\xbf\x16%J\x95~\|\'\r\xe7\x02\xa7\x1d\xf4\xd1\x99\|\x9b\xdeu\xe7\xe2\xfd\xaa\xab\xaa
\xe3(\xf4\x8f+\x81\xdey\xrv\xaa\x9bjK\x07\x01\xbe\xd3\xc77\x1d\xaa)\xdb\x98\x96`i\x87S\x
c4+\xd6~\r\xd3\x9f\x8f\x0b^\'\xe9\xaa\xaa\x1\xe5/\x07\x8a\x8b\x96f\xc6\xfb\xcf\x08*\xeb\x
19\x8d\x94\x9b\xc6\xaa2kj\xdcr\x93\xed/\x845\xe9\xf5\xba\xe1V\x91!\xbe\xb3\xcd\xf5\x0e\x
\x8b\xe3\xde\x1a\xe98\x81\x1f\x94\x98\'\x85\xeb\x1c\xdd\xe6\xe6F\xd7\x98\x9d\xecL\xd7J
\x88\xb8\xe672\xe9\xf0\xbdI{\x05\xc4\xd70\x84j\xc3\x00\xaa\x93%\xab\r\xaa1\xaa6EPI\xbd\xfa
M\x94\xb4\xd4\x85\xb9\xb3\x0e\x9f\r\xd55\xaa\x01\x17i\x7f\x86\xd5\xaa2\x9b\xaa3\xf6\x98\x
f4\xdbU->\x9d\x8e@\x13Jp\xdaX\xb1\x9c\xea\x14\xf5\x8dT(\xa6Tu3\xafOK\xaa\x08\t\xbd\xfd\x
e6j\x9b0\x85\xd1#-\x7f\x1d\xaa5\xff\x00r\xd2m\xaa\xbd=J\x9d\x89\x1ef\x8aU\xeb\xaa\xb1Bm\x
f5\x99\xca\x9b\x19\xabpT\x17\xe4\x13\x0e\xaa\x85\xba\xaa\xb9\xf6\x9e\x80j\xebZ\xde\x
91\xb1\xeb\x05\xaa\xd4k\x86\xaa7\x9e\xd15)<c\xe8]^\'\xec\r\xaa3\xe8i\x99\x88\x16\xb0\x1b\xcf
J\xcb\xce\xc4zY\x89m3\xae};\t\xbdLq\x97J\xca\xd7\x11\xe2\x8d\x85\x8f]\x70\xf0\x95\x9dAJ
X\xf3\x88\xaa6\xd0j11\x9b\xf4\x8d)\x9c\xb2\xc8\x1c\x0b`MT\x99\x1a\xc3\x93\x1ec\x87\x0c\x
f~\fa\xf4\x8e\xaa\xc3k\xde\xe5o\x16\xc5\x02\xd2\xaa7k\x9b\x0e\xd1\x94\xd2\x8d\xaeT\x0fx\x
6\xe1\xda\x82\x06@#6\x06Zp\xddG(6\x1e33\xaa=:\V\x16\x162\x96\x8d\x03|\x81\xd2hN\x1f\|\x
\x97\xe9y\x1b\x86\xd5\xb6\xf6\x12j\xaa5=%\x06\x8b[\xf6\x9a\x17A\xaa\xea\xc0\xcc\xb4u\x
9\xf9U\xac\x06/6\x7fv\xd5\xe5\xb0\xaa\x7fc%\xbff\xda`j\xe9\xb4\xcb\x85\x00\xf7\x99\xda\x
6\x9dE\xca\x80#\xaa\xf0\xfa\xe8\xaa\xf3\xdf\xac\xaa\x1c5\xea-\xdd\xba\xc4\xaa`P\xb0\x00
\x02<\xc2\x02\x85\x85\xec<M\x1f\xddM|\xb8\x8f^\'\x1c\xab`H\x93b\xb9\xac\xb4se\xbf\xbc\xcd
\xaa8*\x89\xf2%\xef\xb6`i\xb8}\xd8r\xb0\xf2a\x7fv\x96\x16\xe6\x12\xf9C\x1c\x1d55`\'\x19\xc
1$\xf8\xaa}4\xc5\x97)\x04\xeb\x9e\x1c\xaba\xce\xbd\x88\xbc/\xee\xf5\x02\xc1\x85\xfb\xde<
\x8c\xc7 r*\x7y !\x1d\xfc\xf4\x9d\n\xbc=I\x16q(\xf0\xe4\xb0\x1c\xe3\xda]\x88\xe7\x92\x
62/\xb6\x00\x9b\xd3\x86\xaa5\xcd\xde\xff\x00M\xaa1\xaa6\n\xd9\xb6\x8f(\xac\x88\xaa\x16\xf
6\x90\xb2\x85\x06\xd3p\xd0+\x10\x03Kn\x18\x00\xbf66\xdeMG5\x18\x17\xdb\xac\xd2\x94\x95\x
85\xf6\x8c\x1a\x03\0\xc4\xd3\x94\xfdv2[\xfd\xaf\xaa\x96\x88\x0c\x06,c\x8d5\x03\x9b\xb6=
\xe0-\'0\xe7\xfa\xc6\xad\x1b\x8f\xcf\x8fy\x0fB\xaa7\xdb\x00\xaa.w;L\xcbD#\|=\'\x8c\x95H\x
b\xbd\xe3\x17Z\x95\x97\x03i\xaa2\x99N\|\x80g\x1d\x1a\xed\xf9\xb1\xda<\|\x8f\xcd\x890\xd7Z
\xf4\xc2\x8c\x8f\xaa4\xaa1Z\xaa\xf5\x19\xeb\x05\x18\xaf\xe7\xb4\x03D\xb6\xf5\x92x\xff\x0
0c\xba5h-\xcaw\x8c\x1a\xf0\x17\xb1\xf38t\xaa\x10\x00/\xb6#\x05&k\xd9\xee;\x7\x8azu\x7f
\x18\xadrn~\xb1\xaa\xab\x0b\xce\x11R\xr\xb9\xaa\x00A\'\x9a\xc6<\x07r\xaa\xb5\x06/2T\xe2j
\xaa4\x81\x99\xcc(\xd6,[\&\x07\xe1\x01\xc9rL\xbe\x13\xf2k\xaa\xfcM\xaa\x02\xaa\x8/\x13\xcfP\x
9b\x9e\xb9\x81\xaa7D\xaa1L\xf5c%Z\xccN6\x96I\r\xaa7#\xb1qs\x89\xaa5\x13\x9c^\'\xff\x00i\xcbf\x
aaG\xcb\x88tZ\xb2\xe7\x9a,\x86\xd7I\xaa95\xf0m\x15P2\x01\x9c\xc2\xaa\xcf\xb9k\x9f&Bns\x93
$\x98k\xff\xd9'
```

In [217...]

```
f=open("C:/Users/PRADEEP COMPUTER/Desktop/asu.jpg","rb")
var = (f.read())
z1 =open("C:/Users/PRADEEP COMPUTER/Desktop/monu.jpg","wb")
z1.write(var)
```

```
z1.close()
f.close()
```

In [218...]

```
z =open("C:/Users/PRADEEP COMPUTER/Desktop/548191.jpg","rb")
var=f.read()
for i in var:
v=var.reverse()
z1 =open("C:/Users/PRADEEP COMPUTER/Desktop/mon.jpg","wb")
z1.write(v)
```

File "<ipython-input-218-7ef7e4de0a64>", line 4
 v=var.reverse()
 ^

IndentationError: expected an indented block

In [219...]

```
import os
os.remove("C:/Users/PRADEEP COMPUTER/Desktop/monu.jpg")
```

In [248...]

```
z =open("C:/Users/PRADEEP COMPUTER/Desktop/rahul.txt","w")
z.write("well,thjs js the word by me.")
z.close()
z1=open("C:/Users/PRADEEP COMPUTER/Desktop/rahul.txt","r")
x=z1.read()
x=x.replace('j','i')

print(x)
```

well,this is the word by me.

In [260...]

```
string="fanaj dgre dsfi dgjdi"
x=string.split(" ")
for i in x:
  s=list(i)
  print(s)
```

```
['f', 'a', 'n', 'a', 'j']
['d', 'g', 'r', 'e']
['d', 's', 'f', 'i']
['d', 'g', 'j', 'd', 'i']
```

In [30]:

```
word = 'sonu'
print(list(word))
```

TypeError Traceback (most recent call last)
<ipython-input-30-277df6c046d3> in <module>
 1 word = 'sonu'
----> 2 print(list(word))

TypeError: 'list' object is not callable

In [327...]

```
def fun(self,arr,n):
  for i in range(1,n+1):
    self.arr = arr.count(i)
arr=[10,2,10,2,10,5,6,6,6,1,7,2]
n=10
```

```
fun(arr,n)
#for i in range(n):
#    print(i)
```

TypeError Traceback (most recent call last)
<ipython-input-327-da5fa494a632> in <module>
 4 arr=[10,2,10,2,10,5,6,6,6,1,7,2]
 5 n=10
----> 6 fun(arr,n)
 7 #for i in range(n):
 8 # print(i)

TypeError: fun() missing 1 required positional argument: 'n'

In [322...]

```
def area(x1,x2,x3,y1,y2,y3):
    return (1/2)*(x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2))
x1=int(input("1st coordinate of x:"))
x2=int(input("2nd coordinate of x:"))
x3=int(input("3rd coordinate of x:"))
y1=int(input("1st coordinate of y:"))
y2=int(input("2nd coordinate of y:"))
y3=int(input("3rd coordinate of y:"))
print(area(x1,x2,x3,y1,y2,y3))
```

```
1st coordinate of x:1
2nd coordinate of x:2
3rd coordinate of x:3
1st coordinate of y:3
2nd coordinate of y:5
3rd coordinate of y:2
-2.5
```

In [37]:

```
s="sonu"
str=list(s)
```

TypeError Traceback (most recent call last)
<ipython-input-37-10290c4b2aac> in <module>
 1 s="sonu"
----> 2 str=list(s)
 3
 4

TypeError: 'list' object is not callable

In [362...]

```
print(str)
```

```
gg
```

In [4]:

```
a=int(input(10))
b=int(input(20))
print("addition of %d and %d is %d"%(a,b,a+b))
```

```
1010
2020
addition of 10 and 20 is 30
```

In [21]:

```
import math  
math.floor(2.13)
```

Out[21]: 2

In [24]:

```
print("a"+"bc")
```

abc

In [36]:

```
word="sonu"  
print(list(word))
```

TypeError

Traceback (most recent call last)

```
<ipython-input-36-148b17f84476> in <module>  
      1 word="sonu"  
----> 2 print(list(word))
```

TypeError: 'list' object is not callable

In [38]:

```
set1={1,2,5,3,4,6,7,8}  
set2={2,5,3,4,1,6,7,8,9}  
set3=set1.intersection(set2)  
print(set3)
```

{1, 2, 3, 4, 5, 6, 7, 8}

In [39]:

```
set4=set1.union(set2)  
print(set4)
```

{1, 2, 3, 4, 5, 6, 7, 8, 9}

In [20]:

```
import numpy as np  
a=np.array([3,5,48,-2])  
b=np.array([12,5,48,20])  
print(a+b)  
print(a-b)  
print(a*b)  
print(a/b)  
print(a**b)  
print(a%b)
```

```
[15 10 96 18]  
[-9 0 0 -22]  
[ 36 25 2304 -40]  
[ 0.25 1. 1. -0.1 ]  
[ 531441 3125 0 1048576]  
[ 3 0 0 18]
```

In [32]:

```
c=np.array([[1,2,3],[4,25,9]])  
print(np.transpose(c))
```

```
[[ 1 4]  
[ 2 25]  
[ 3 9]]
```

```
In [8]: print(c.T)
```

```
[[ 1  3 20]
 [ 2 25 23]]
```

```
In [13]: #it will multiply the matrices and then sum them values
print(np.inner(a,b))
```

```
2325
```

```
In [14]: print(np.dot(a,b))
```

```
2325
```

```
In [17]: c.diagonal()
```

```
Out[17]: array([ 1, 25])
```

```
In [18]: c.trace()
```

```
Out[18]: 26
```

```
In [19]: sum(c.diagonal())
```

```
Out[19]: 26
```

```
In [34]: print(np.add(a,b))
print(np.subtract(a,b))
print(np.divide(a,b))
print(np.dot(a,b))

print(np.sqrt(b))
print(np.multiply(a,b))
print(np.sum(a))
print(np.sum(a, axis=0))
#axis=0 is going to be coloum axix
print(np.sum(c, axis=0))
print(np.sum(c))
#axis=1 is going to be row axis
print(np.sum(c, axis=1))
```

```
[15 10 96 18]
[-9 0 0 -22]
[ 0.25 1. 1. -0.1 ]
2325
[3.46410162 2.23606798 6.92820323 4.47213595]
[ 36 25 2304 -40]
54
54
[ 5 27 12]
44
[ 6 38]
```

```
In [36]: print(a.sort())
```

None

In [56]:

```
c=np.array([[1,2,3],[4,25,9],[12,-23,-5]])
#by default row wise sorting is done
c.sort()
print(c)
#colum wise sorting using axis=0
c.sort(axis=0)
print(c)
#row wise sorting in 2D array using axis=1
c.sort(axis=1)
print(c)
```

```
[[ 1  2  3]
 [ 4  9 25]
 [-23 -5 12]]
[[-23 -5  3]
 [ 1  2 12]
 [ 4  9 25]]
[[-23 -5  3]
 [ 1  2 12]
 [ 4  9 25]]
```

In [46]:

```
print(c.max())
```

```
25
```

In [47]:

```
print(a.max())
```

```
48
```

In [48]:

```
print(c.max(axis=0))
```

```
[ 4  9 25]
```

In [60]:

```
print(c.max(axis=1))
```

```
[ 3 12 25]
```

In [50]:

```
print(c.min())
```

```
-23
```

In [51]:

```
#calculating the mean
print(c.mean())
```

```
3.111111111111111
```

In [52]:

```
#calculating standard deviation
print(c.std())
```

```
12.233328287414441
```

In [53]:

```
print(c.std(axis=0))
```

```
[12.08304597  5.71547607  9.03081146]
```

```
In [54]: print(c.std(axis=1))
```

```
[10.87300429 4.96655481 8.9566859 ]
```

```
In [59]: print(c.mean(axis=0))
print(c.mean(axis=1))
```

```
[-6. 2. 13.33333333]
[-8.33333333 5. 12.66666667]
```

```
In [58]: print(a.mean())
```

```
13.5
```

```
In [65]: a=np.array([[1,3,4],[54,20,30],[52,1,30]])
print(a[0:2,1:3])
a=np.array([10,10,10,30])
a1=a[0:3]
print(a1,a)
a1[:]=10
print(a1,a)
```

```
[[ 3 4]
 [20 30]]
[10 10 10] [10 10 10 30]
[10 10 10] [10 10 10 30]
```

```
In [87]: ar=np.arange(0,10,2)
ar1=np.arange(10)
print(ar,ar1)
```

```
[0 2 4 6 8] [0 1 2 3 4 5 6 7 8 9]
```

```
In [75]: arr=np.mgrid[0:5,0:4]
print(arr)
```

```
[[[0 0 0 0]
 [1 1 1 1]
 [2 2 2 2]
 [3 3 3 3]
 [4 4 4 4]]
```

```
[[[0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]]]
```

```
In [92]: #liinspace divide the element in 11 parts
print(np.linspace(10,12,11))
```

```
[10. 10.2 10.4 10.6 10.8 11. 11.2 11.4 11.6 11.8 12. ]
```

```
In [82]: np.arange(12).reshape(3,4)
```

```
array([[ 0, 1, 2, 3],
```

```
Out[82]: [[ 4,  5,  6,  7],  
          [ 8,  9, 10, 11]])
```

```
In [83]: np.arange(9).reshape(3,3)
```

```
Out[83]: array([[0, 1, 2],  
                 [3, 4, 5],  
                 [6, 7, 8]])
```

```
In [95]: print(np.linspace(np.arange(9).reshape(3,3),2))
```

```
[[[0.          1.          2.          ]]  
 [3.          4.          5.          ]]  
 [6.          7.          8.          ]]  
  
 [[[0.04081633 1.02040816 2.          ]]  
 [2.97959184 3.95918367 4.93877551]  
 [5.91836735 6.89795918 7.87755102]]]  
  
 [[[0.08163265 1.04081633 2.          ]]  
 [2.95918367 3.91836735 4.87755102]  
 [5.83673469 6.79591837 7.75510204]]]  
  
 [[[0.12244898 1.06122449 2.          ]]  
 [2.93877551 3.87755102 4.81632653]  
 [5.75510204 6.69387755 7.63265306]]]  
  
 [[[0.16326531 1.08163265 2.          ]]  
 [2.91836735 3.83673469 4.75510204]  
 [5.67346939 6.59183673 7.51020408]]]  
  
 [[[0.20408163 1.10204082 2.          ]]  
 [2.89795918 3.79591837 4.69387755]  
 [5.59183673 6.48979592 7.3877551]]]  
  
 [[[0.24489796 1.12244898 2.          ]]  
 [2.87755102 3.75510204 4.63265306]  
 [5.51020408 6.3877551 7.26530612]]]  
  
 [[[0.28571429 1.14285714 2.          ]]  
 [2.85714286 3.71428571 4.57142857]  
 [5.42857143 6.28571429 7.14285714]]]  
  
 [[[0.32653061 1.16326531 2.          ]]  
 [2.83673469 3.67346939 4.51020408]  
 [5.34693878 6.18367347 7.02040816]]]  
  
 [[[0.36734694 1.18367347 2.          ]]  
 [2.81632653 3.63265306 4.44897959]  
 [5.26530612 6.08163265 6.89795918]]]  
  
 [[[0.40816327 1.20408163 2.          ]]  
 [2.79591837 3.59183673 4.3877551]  
 [5.18367347 5.97959184 6.7755102]]]  
  
 [[[0.44897959 1.2244898 2.          ]]  
 [2.7755102 3.55102041 4.32653061]  
 [5.10204082 5.87755102 6.65306122]]]  
  
 [[[0.48979592 1.24489796 2.          ]]  
 [2.75510204 3.51020408 4.26530612]  
 [5.02040816 5.7755102 6.53061224]]]
```

```
[[0.53061224 1.26530612 2.          ]
 [2.73469388 3.46938776 4.20408163]
 [4.93877551 5.67346939 6.40816327]]  
  
[[0.57142857 1.28571429 2.          ]
 [2.71428571 3.42857143 4.14285714]
 [4.85714286 5.57142857 6.28571429]]  
  
[[0.6122449  1.30612245 2.          ]
 [2.69387755 3.3877551  4.08163265]
 [4.7755102  5.46938776 6.16326531]]  
  
[[0.65306122 1.32653061 2.          ]
 [2.67346939 3.34693878 4.02040816]
 [4.69387755 5.36734694 6.04081633]]  
  
[[0.69387755 1.34693878 2.          ]
 [2.65306122 3.30612245 3.95918367]
 [4.6122449  5.26530612 5.91836735]]  
  
[[0.73469388 1.36734694 2.          ]
 [2.63265306 3.26530612 3.89795918]
 [4.53061224 5.16326531 5.79591837]]  
  
[[0.7755102  1.3877551  2.          ]
 [2.6122449  3.2244898  3.83673469]
 [4.44897959 5.06122449 5.67346939]]  
  
[[0.81632653 1.40816327 2.          ]
 [2.59183673 3.18367347 3.7755102 ]
 [4.36734694 4.95918367 5.55102041]]  
  
[[0.85714286 1.42857143 2.          ]
 [2.57142857 3.14285714 3.71428571]
 [4.28571429 4.85714286 5.42857143]]  
  
[[0.89795918 1.44897959 2.          ]
 [2.55102041 3.10204082 3.65306122]
 [4.20408163 4.75510204 5.30612245]]  
  
[[0.93877551 1.46938776 2.          ]
 [2.53061224 3.06122449 3.59183673]
 [4.12244898 4.65306122 5.18367347]]  
  
[[0.97959184 1.48979592 2.          ]
 [2.51020408 3.02040816 3.53061224]
 [4.04081633 4.55102041 5.06122449]]  
  
[[1.02040816 1.51020408 2.          ]
 [2.48979592 2.97959184 3.46938776]
 [3.95918367 4.44897959 4.93877551]]  
  
[[1.06122449 1.53061224 2.          ]
 [2.46938776 2.93877551 3.40816327]
 [3.87755102 4.34693878 4.81632653]]  
  
[[1.10204082 1.55102041 2.          ]
 [2.44897959 2.89795918 3.34693878]
 [3.79591837 4.24489796 4.69387755]]  
  
[[1.14285714 1.57142857 2.          ]
 [2.42857143 2.85714286 3.28571429]
 [3.71428571 4.14285714 4.57142857]]  
  
[[1.18367347 1.59183673 2.          ]]
```

```
[2.40816327 2.81632653 3.2244898 ]  
[3.63265306 4.04081633 4.44897959]]
```

```
[[1.2244898 1.6122449 2. ]  
[2.3877551 2.7755102 3.16326531]  
[3.55102041 3.93877551 4.32653061]]
```

```
[[1.26530612 1.63265306 2. ]  
[2.36734694 2.73469388 3.10204082]  
[3.46938776 3.83673469 4.20408163]]
```

```
[[1.30612245 1.65306122 2. ]  
[2.34693878 2.69387755 3.04081633]  
[3.3877551 3.73469388 4.08163265]]
```

```
[[1.34693878 1.67346939 2. ]  
[2.32653061 2.65306122 2.97959184]  
[3.30612245 3.63265306 3.95918367]]
```

```
[[1.3877551 1.69387755 2. ]  
[2.30612245 2.6122449 2.91836735]  
[3.2244898 3.53061224 3.83673469]]
```

```
[[1.42857143 1.71428571 2. ]  
[2.28571429 2.57142857 2.85714286]  
[3.14285714 3.42857143 3.71428571]]
```

```
[[1.46938776 1.73469388 2. ]  
[2.26530612 2.53061224 2.79591837]  
[3.06122449 3.32653061 3.59183673]]
```

```
[[1.51020408 1.75510204 2. ]  
[2.24489796 2.48979592 2.73469388]  
[2.97959184 3.2244898 3.46938776]]
```

```
[[1.55102041 1.7755102 2. ]  
[2.2244898 2.44897959 2.67346939]  
[2.89795918 3.12244898 3.34693878]]
```

```
[[1.59183673 1.79591837 2. ]  
[2.20408163 2.40816327 2.6122449 ]  
[2.81632653 3.02040816 3.2244898 ]]
```

```
[[1.63265306 1.81632653 2. ]  
[2.18367347 2.36734694 2.55102041]  
[2.73469388 2.91836735 3.10204082]]
```

```
[[1.67346939 1.83673469 2. ]  
[2.16326531 2.32653061 2.48979592]  
[2.65306122 2.81632653 2.97959184]]
```

```
[[1.71428571 1.85714286 2. ]  
[2.14285714 2.28571429 2.42857143]  
[2.57142857 2.71428571 2.85714286]]
```

```
[[1.75510204 1.87755102 2. ]  
[2.12244898 2.24489796 2.36734694]  
[2.48979592 2.6122449 2.73469388]]
```

```
[[1.79591837 1.89795918 2. ]  
[2.10204082 2.20408163 2.30612245]  
[2.40816327 2.51020408 2.6122449 ]]
```

```
[[1.83673469 1.91836735 2. ]  
[2.08163265 2.16326531 2.24489796]]
```

```
[2.32653061 2.40816327 2.48979592]]
[[1.87755102 1.93877551 2.          ]
 [2.06122449 2.12244898 2.18367347]
 [2.24489796 2.30612245 2.36734694]]
[[1.91836735 1.95918367 2.          ]
 [2.04081633 2.08163265 2.12244898]
 [2.16326531 2.20408163 2.24489796]]
[[1.95918367 1.97959184 2.          ]
 [2.02040816 2.04081633 2.06122449]
 [2.08163265 2.10204082 2.12244898]]
[[2.          2.          2.          ]
 [2.          2.          2.          ]
 [2.          2.          2.          ]]]]
```

In [105...]

```
#it will give the only 1 random number between 0 to 1
print(np.random.rand())
#it will give the 5 random number between 0 to 1
print(np.random.rand(5))
```

```
0.5912417813464794
[0.0359773 0.67727129 0.51608413 0.79142754 0.75869143]
```

In [111...]

```
print(2*np.random.rand())
```

```
1.446603639485933
```

In [109...]

```
#it will create the matrix of random number
print(np.random.rand(3,4)*100)
```

```
[[30.48964394 28.98877735 15.74649929 28.92862179]
 [75.50819695 11.04818998 68.81668149 27.27228136]
 [26.55856249 34.75053779 57.35929039 8.32514023]]
```

In [115...]

```
print(np.random.randint(2,10,4))
```

```
[8 7 6 5]
```

In [117...]

```
print(np.random.randint(2,10,12).reshape(3,4))
```

```
[[4 9 4 8]
 [3 6 9 9]
 [8 6 9 9]]
```

In [127...]

```
print(np.random.randn(5,5))
```

```
[[ 1.39169308 -0.4318332   0.33254864  0.2322018   1.94093059]
 [ 0.86265344 -0.69405557 -1.45395374 -0.62456982 -0.27212333]
 [ 1.93238172  0.13411526  0.36458976 -0.44947144  1.19326784]
 [-0.50871379 -1.20381499 -0.6550584   -0.7932702   -1.13687841]
 [ 0.28947373  0.04294532 -0.47901568  1.47340832 -2.29816451]]
```

In [2]:

```
import pandas as pd
s2=pd.Series([1,2,3,4,5],index=['jan','feb','march','apr','may'])
```

```
print(s2['feb'])
print(s2[:2])
```

```
2
jan    1
feb    2
dtype: int64
```

In [148...]

```
s1=pd.Series([1,2,3,40])
s2=pd.Series([8,6,9,7])
print(s1+s2)
print(s1-s2)
print(s1*s2)
print(s1/s2)
print(s1//s2)
print(s1%s2)
```

```
0      9
1      8
2     12
3     47
dtype: int64
0     -7
1     -4
2     -6
3     33
dtype: int64
0      8
1     12
2     27
3    280
dtype: int64
0    0.125000
1    0.333333
2    0.333333
3    5.714286
dtype: float64
0      0
1      0
2      0
3      5
dtype: int64
0      1
1      2
2      3
3      5
dtype: int64
```

```
max(s1)
```

Out[151... 40

In [152...]

```
min(s2)
```

Out[152... 6

In [153...]

```
s1.max()
```

Out[153... 40

```
In [158... df=pd.DataFrame()
a1=np.array([12,32])
a2=np.array([15,63,95])
a3=np.array([10,52,41,63,10])
pd.DataFrame(a1)
```

Out[158... 0

0 12**1** 32

```
In [159... pd.DataFrame([a1,a2])
```

Out[159... 0 1 2

0 12 32 NaN**1** 15 63 95.0

```
In [164... pd.DataFrame([a1,a2,a3],columns=['A','B','C','D','E'])
```

Out[164... A B C D E

0 12 32 NaN NaN NaN**1** 15 63 95.0 NaN NaN**2** 10 52 41.0 63.0 10.0

```
In [172... print(df)
print(df.count())
print(len(df)-df.count())
```

```
Empty DataFrame
Columns: []
Index: []
Series([], dtype: int64)
Series([], dtype: int64)
```

```
In [11]: import pandas as pd
df = pd.read_csv("C:/Users/PRADEEP COMPUTER/Desktop/book.csv")
print(df)
```

	studets	rollno
0	sonu	1
1	monu	2
2	asu	3

In [14]: df.tail()

Out[14]: studets rollno

	studets	rollno
0	sonu	1
1	monu	2
2	asu	3

In [15]: df.head(10)

Out[15]: studets rollno

	studets	rollno
0	sonu	1
1	monu	2
2	asu	3

In [16]: df.tail(2)

Out[16]: studets rollno

	studets	rollno
1	monu	2
2	asu	3

In [17]: df.count()

Out[17]: studets rollno 3
dtype: int64

In [19]: dict={'name':['a','b','c'],'address':['x','y','z']}
df=pd.DataFrame(dict)
print(df)

	name	address
0	a	x
1	b	y
2	c	z

In [24]: df.to_csv('path')

In [27]: dict={'name':['a','b','c'],'address':['x','y','z']}
df=pd.DataFrame(dict,index=['A','B','C','D'],columns=['col_1','col_2'])
#df.to_csv('path')
print(df)

TypeError

Traceback (most recent call last)

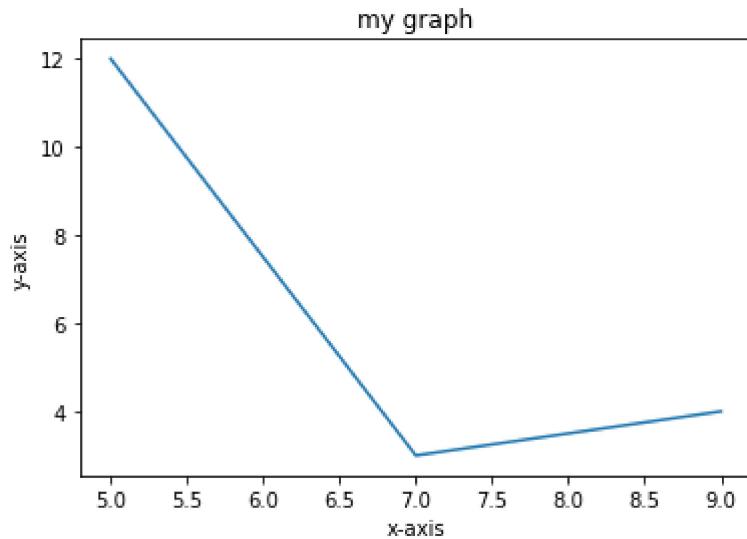
```
<ipython-input-27-582ba841e39e> in <module>
      1 dict={'name':['a','b','c'],'address':['x','y','z']}
----> 2 df=pd.DataFrame(dict,index=['A','B','C','D'],columns=['col_1','col_2'])
      3 #df.to_csv('path')
```

```
4 print(df)
```

TypeError: __init__() got an unexpected keyword argument 'columns'

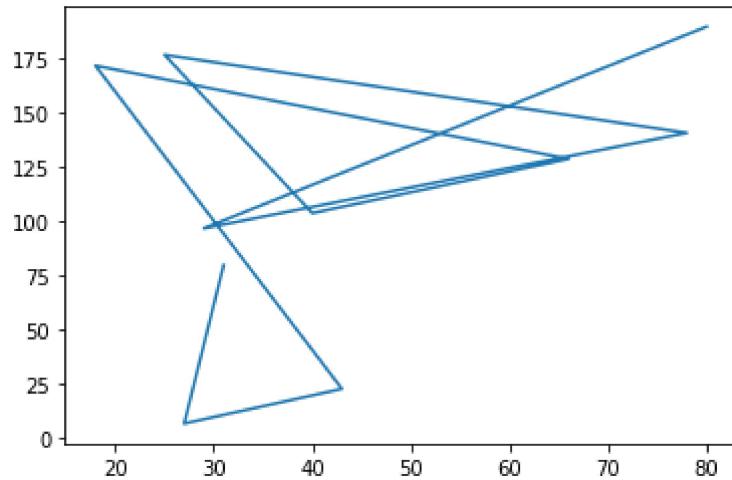
In [2]:

```
from matplotlib import pyplot as p
p.plot([5,7,9],[12,3,4])
p.title("my graph")
p.xlabel("x-axis")
p.ylabel("y-axis")
p.show()
```



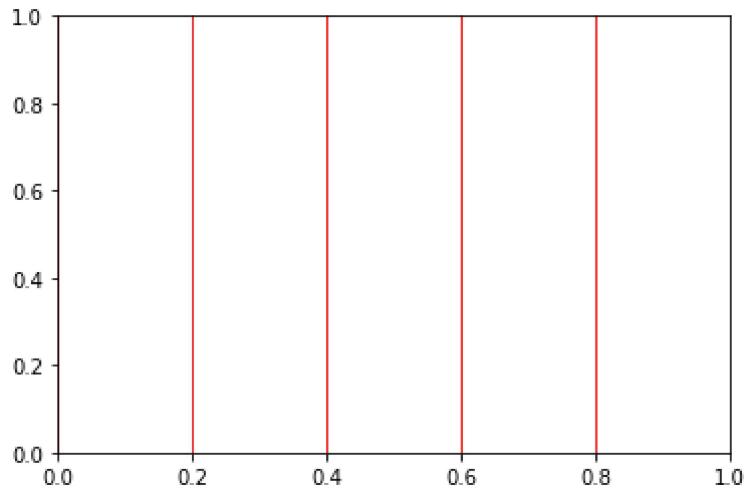
In [6]:

```
import numpy as np
x=np.random.randint(1,100,10)
y=np.random.randint(1,200,10)
p.plot(x,y)
p.show()
```

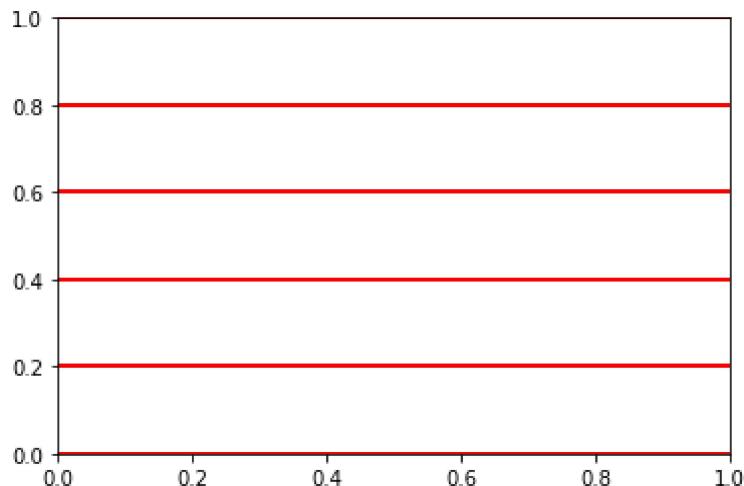


In [8]:

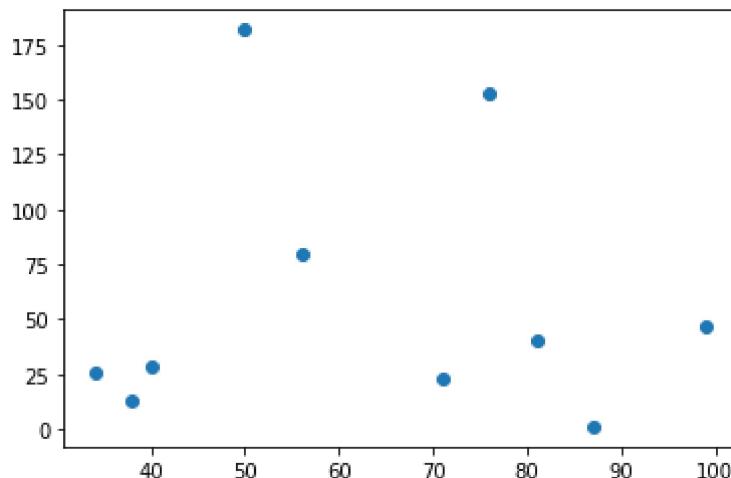
```
p.grid(axis='x', color='red')
```



```
In [12]: p.grid(axis='y', color='red', linewidth="2")
```

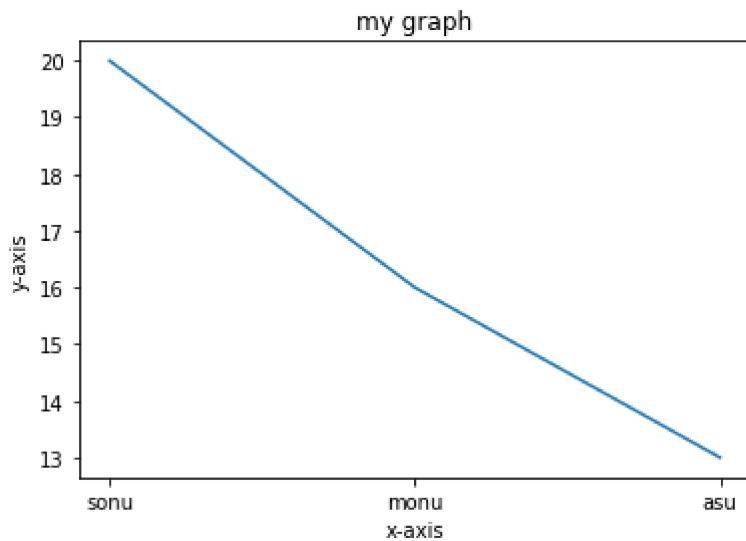


```
In [13]: import numpy as np  
x=np.random.randint(1,100,10)  
y=np.random.randint(1,200,10)  
p.scatter(x,y)  
p.show()
```



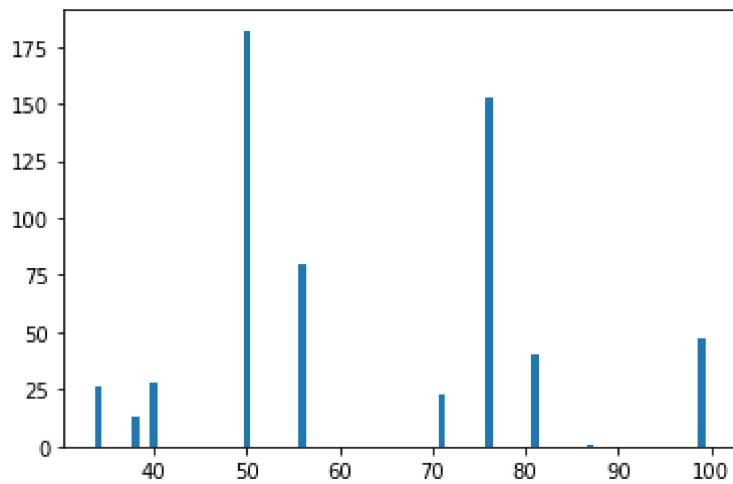
```
In [54]:
```

```
import numpy as np
from matplotlib import pyplot as p
list1=['sonu','monu','asu']
list2=[20,16,13]
p.plot(list1,list2)
p.title("my graph")
p.xlabel("x-axis")
p.ylabel("y-axis")
p.show()
```



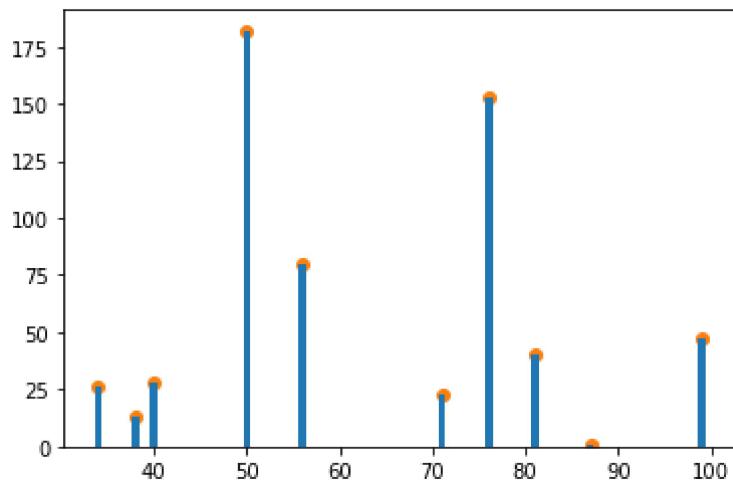
In [29]:

```
p.bar(x,y)
p.show()
```



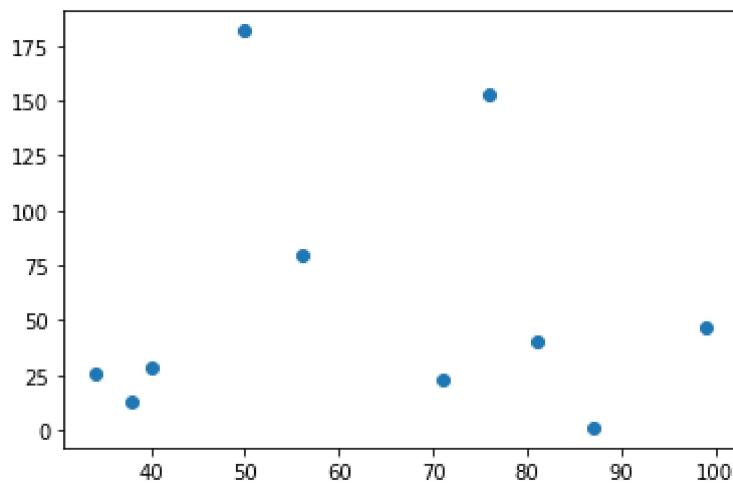
In [32]:

```
p.bar(x,y)
p.scatter(x,y)
p.show()
```



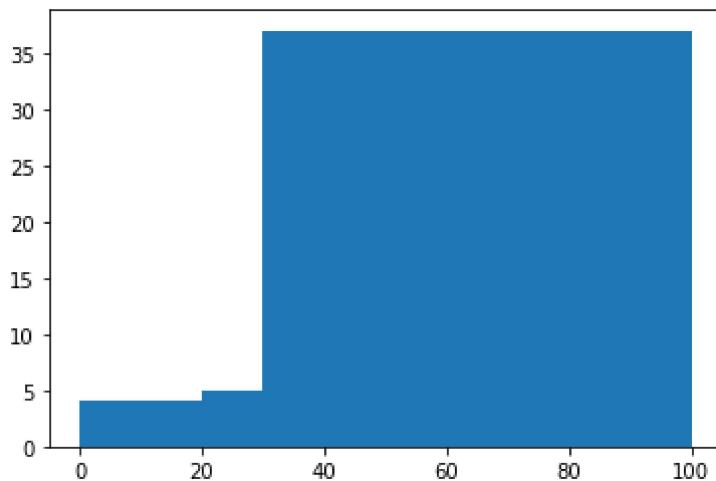
```
In [31]: p.scatter(x,y)
```

```
Out[31]: <matplotlib.collections.PathCollection at 0x1a74cf525b0>
```



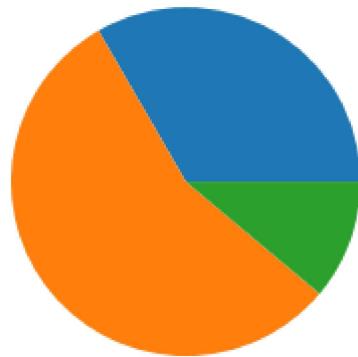
```
In [27]: age=np.random.randint(1,100,50)
bin=[0,10,20,30,100]
p.hist(age,bin)
```

```
Out[27]: (array([ 4.,  4.,  5., 37.]),
 array([ 0, 10, 20, 30, 100]),
 <BarContainer object of 4 artists>)
```



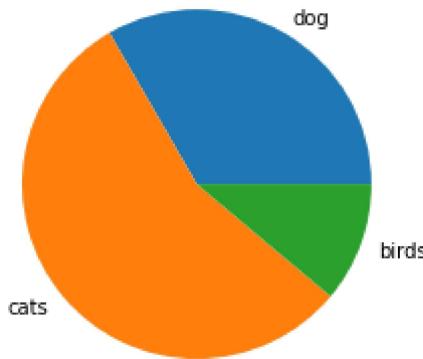
```
In [34]: x=np.array([30,50,10])
p.pie(x)
```

```
Out[34]: ([<matplotlib.patches.Wedge at 0x1a74d217280>,
<matplotlib.patches.Wedge at 0x1a74d217700>,
<matplotlib.patches.Wedge at 0x1a74d217b80>],
[Text(0.5499999702695115, 0.9526279613277875, ''),
Text(-0.842648784452574, -0.7070664933799364, ''),
Text(1.0336619699471659, -0.37622191840048963, '')])
```



```
In [38]: x=np.array([30,50,10])
y=["dog","cats","birds"]
p.pie(x,labels=y)
```

```
Out[38]: ([<matplotlib.patches.Wedge at 0x1a74d3f7bb0>,
<matplotlib.patches.Wedge at 0x1a74d4050d0>,
<matplotlib.patches.Wedge at 0x1a74d405550>],
[Text(0.5499999702695115, 0.9526279613277875, 'dog'),
Text(-0.842648784452574, -0.7070664933799364, 'cats'),
Text(1.0336619699471659, -0.37622191840048963, 'birds')])
```

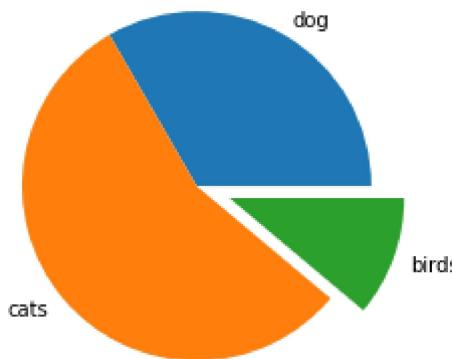


In [41]:

```
x=np.array([30,50,10])
y=["dog","cats","birds"]
p.pie(x,labels=y,explode=[0,0,0.2])
```

Out[41]:

```
([<matplotlib.patches.Wedge at 0x1a74d1a2f10>,
 <matplotlib.patches.Wedge at 0x1a74d1cdaf0>,
 <matplotlib.patches.Wedge at 0x1a74d1cd370>],
 [Text(0.5499999702695115, 0.9526279613277875, 'dog'),
 Text(-0.842648784452574, -0.7070664933799364, 'cats'),
 Text(1.2216005099375598, -0.444625903564215, 'birds')])
```

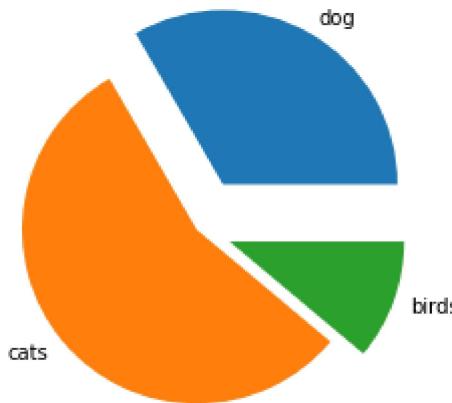


In [42]:

```
x=np.array([30,50,10])
y=["dog","cats","birds"]
p.pie(x,labels=y,explode=[0.3,0,0.2])
```

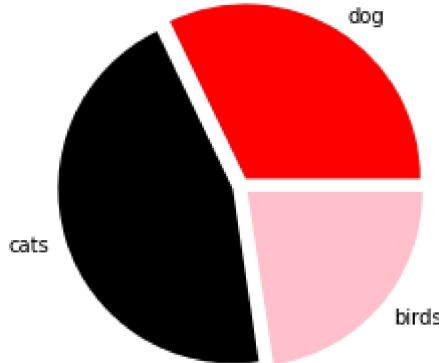
Out[42]:

```
([<matplotlib.patches.Wedge at 0x1a74cf31e50>,
 <matplotlib.patches.Wedge at 0x1a74cf31ee0>,
 <matplotlib.patches.Wedge at 0x1a74cf313d0>],
 [Text(0.6999999621611965, 1.2124355871444568, 'dog'),
 Text(-0.842648784452574, -0.7070664933799364, 'cats'),
 Text(1.2216005099375598, -0.444625903564215, 'birds')])
```



```
In [50]: x=np.array([27,38,19])
y=["dog","cats","birds"]
p.pie(x,labels=y,explode=[0.05,0.05,0.05],colors=['red','black','pink'])
```

```
Out[50]: ([<matplotlib.patches.Wedge at 0x1a74e50d6a0>,
<matplotlib.patches.Wedge at 0x1a74e50db80>,
<matplotlib.patches.Wedge at 0x1a74e50dfa0>],
[Text(0.6118369010165581, 0.9737328209290548, 'dog'),
Text(-1.0989087326991127, -0.33896843097496626, 'cats'),
Text(0.8716674782728758, -0.7501305268560969, 'birds')])
```



```
In [66]: import pandas as pd
import numpy as np
d = pd.read_csv("C:/Users/PRADEEP COMPUTER/Desktop/student-por.csv")
print(df)
```

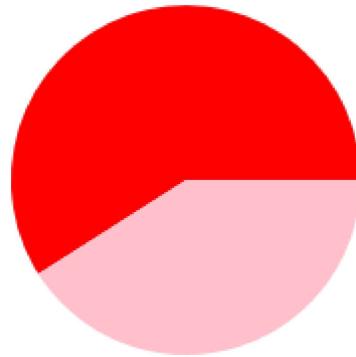
	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	\
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	
1	GP	F	17	U	GT3	T	1	1	at_home	other	
2	GP	F	15	U	LE3	T	1	1	at_home	other	
3	GP	F	15	U	GT3	T	4	2	health	services	
4	GP	F	16	U	GT3	T	3	3	other	other	
..
644	MS	F	19	R	GT3	T	2	3	services	other	
645	MS	F	18	U	LE3	T	3	1	teacher	services	
646	MS	F	18	U	GT3	T	1	1	other	other	
647	MS	M	17	U	LE3	T	3	1	services	services	
648	MS	M	18	R	LE3	T	3	2	services	other	

	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	...	4	3	4	1	1	3	4	0	11	11
1	...	5	3	3	1	1	3	2	9	11	11
2	...	4	3	2	2	3	3	6	12	13	12
3	...	3	2	2	1	1	5	0	14	14	14
4	...	4	3	2	1	2	5	0	11	13	13
..
644	...	5	4	2	1	2	5	4	10	11	10
645	...	4	3	4	1	1	1	4	15	15	16
646	...	1	1	1	1	1	5	6	11	12	9
647	...	2	4	5	3	4	2	6	10	10	10
648	...	4	4	1	3	4	5	4	10	11	11

[649 rows x 33 columns]

In [67]:

```
x=d['sex'].value_counts()
p.pie(x,colors=['red','pink'])
p.show()
print(x)
```

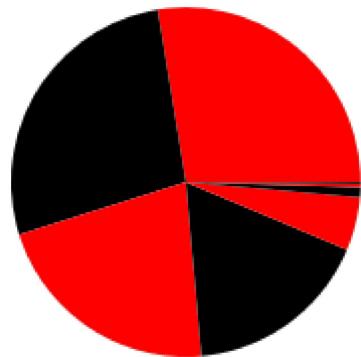


F	383
M	266
Name: sex, dtype: int64	

In [75]:

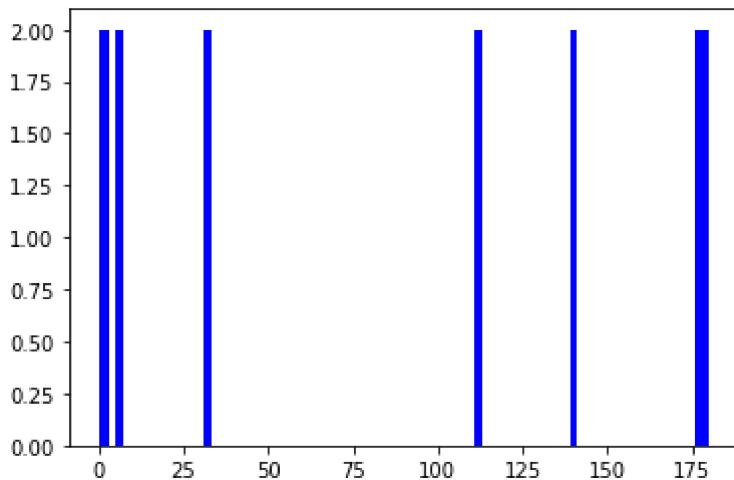
```
x=d['age'].value_counts()
print(x)
p.pie(x,colors=['red','black'])
p.show()
```

17	179
16	177
18	140
15	112
19	32
20	6
21	2
22	1
Name: age, dtype: int64	



In [78]:

```
x=d['age'].value_counts()
p.bar(x,height=2,color='blue',width=2)
p.show()
```



In [92]:

```
x=d['age'].value_counts()
y=d['sex'].value_counts()
p.scatter(x,y)
p.show()
```

```
-----  
ValueError                                     Traceback (most recent call last)  
<ipython-input-92-06c1c7af7170> in <module>  
      1 x=d['age'].value_counts()  
      2 y=d['sex'].value_counts()  
----> 3 p.scatter(x,y)  
      4 p.show()  
  
~\anaconda3\lib\site-packages\matplotlib\pyplot.py in scatter(x, y, s, c, marker, cmap,  
      norm, vmin, vmax, alpha, linewidths, verts, edgecolors, plotnonfinite, data, **kwargs)  
  2888         verts=cbook.deprecation._deprecated_parameter,  
  2889         edgecolors=None, *, plotnonfinite=False, data=None, **kwargs):  
-> 2890     _ret = gca().scatter(  
  2891         x, y, s=s, c=c, marker=marker, cmap=cmap, norm=norm,  
  2892         vmin=vmin, vmax=vmax, alpha=alpha, linewidths=linewidths,  
  
~\anaconda3\lib\site-packages\matplotlib\__init__.py in inner(ax, data, *args, **kwargs)  
 1445     def inner(ax, *args, data=None, **kwargs):  
 1446         if data is None:
```

```

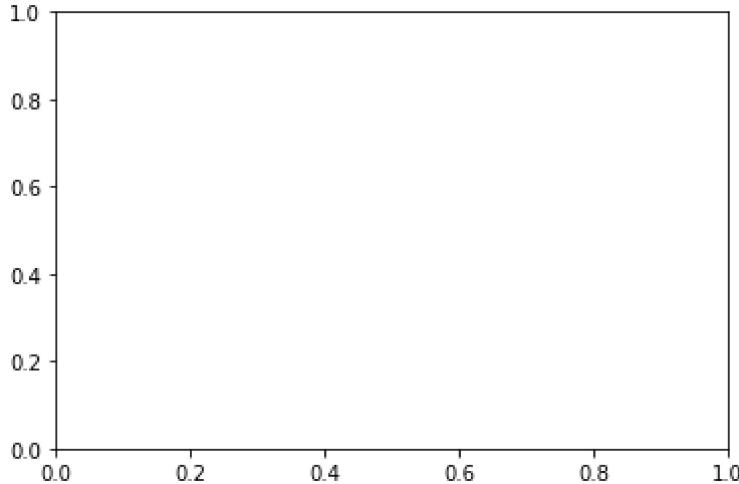
-> 1447             return func(ax, *map(sanitize_sequence, args), **kwargs)
1448
1449     bound = new_sig.bind(ax, *args, **kwargs)

~\anaconda3\lib\site-packages\matplotlib\cbook\deprecation.py in wrapper(*inner_args, **inner_kwargs)
    409                     else deprecation_addendum,
    410                         **kwargs)
--> 411         return func(*inner_args, **inner_kwargs)
    412
    413     return wrapper

~\anaconda3\lib\site-packages\matplotlib\axes\_axes.py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, verts, edgecolors, plotnonfinite, **kwargs)
    439         y = np.ma.ravel(y)
    440         if x.size != y.size:
-> 441             raise ValueError("x and y must be the same size")
    442
    443         if s is None:

```

ValueError: x and y must be the same size



In [96]:

```

import seaborn as s
x=d.corr()
print(x)

```

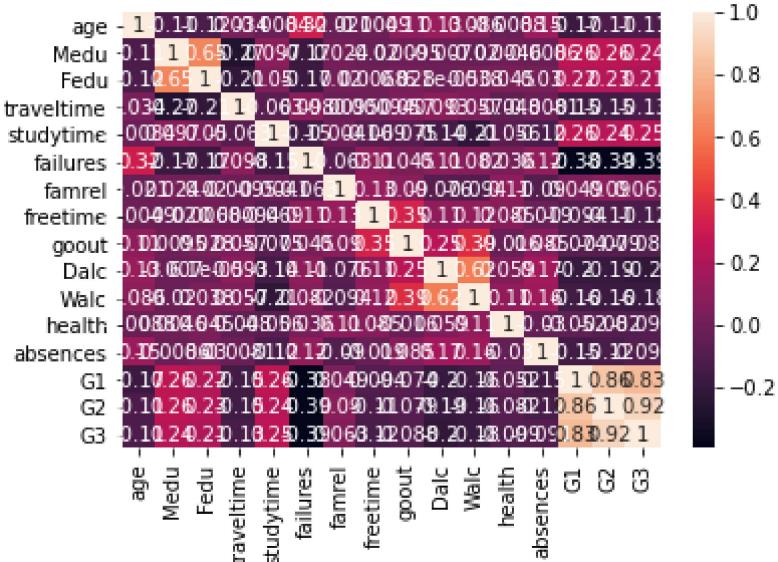
	age	Medu	Fedu	traveltime	studytime	failures	\
age	1.000000	-0.107832	-0.121050	0.034490	-0.008415	0.319968	
Medu	-0.107832	1.000000	0.647477	-0.265079	0.097006	-0.172210	
Fedu	-0.121050	0.647477	1.000000	-0.208288	0.050400	-0.165915	
traveltime	0.034490	-0.265079	-0.208288	1.000000	-0.063154	0.097730	
studytime	-0.008415	0.097006	0.050400	-0.063154	1.000000	-0.147441	
failures	0.319968	-0.172210	-0.165915	0.097730	-0.147441	1.000000	
famrel	-0.020559	0.024421	0.020256	-0.009521	-0.004127	-0.062645	
freetime	-0.004910	-0.019686	0.006841	0.000937	-0.068829	0.108995	
goout	0.112805	0.009536	0.027690	0.057454	-0.075442	0.045078	
Dalc	0.134768	-0.007018	0.000061	0.092824	-0.137585	0.105949	
Walc	0.086357	-0.019766	0.038445	0.057007	-0.214925	0.082266	
health	-0.008750	0.004614	0.044910	-0.048261	-0.056433	0.035588	
absences	0.149998	-0.008577	0.029859	-0.008149	-0.118389	0.122779	
G1	-0.174322	0.260472	0.217501	-0.154120	0.260875	-0.384210	
G2	-0.107119	0.264035	0.225139	-0.154489	0.240498	-0.385782	
G3	-0.106505	0.240151	0.211800	-0.127173	0.249789	-0.393316	
	famrel	freetime	goout	Dalc	Walc	health	\
age	-0.020559	-0.004910	0.112805	0.134768	0.086357	-0.008750	

Medu	0.024421	-0.019686	0.009536	-0.007018	-0.019766	0.004614
Fedu	0.020256	0.006841	0.027690	0.000061	0.038445	0.044910
traveltime	-0.009521	0.000937	0.057454	0.092824	0.057007	-0.048261
studytime	-0.004127	-0.068829	-0.075442	-0.137585	-0.214925	-0.056433
failures	-0.062645	0.108995	0.045078	0.105949	0.082266	0.035588
famrel	1.000000	0.129216	0.089707	-0.075767	-0.093511	0.109559
freetime	0.129216	1.000000	0.346352	0.109904	0.120244	0.084526
goout	0.089707	0.346352	1.000000	0.245126	0.388680	-0.015741
Dalc	-0.075767	0.109904	0.245126	1.000000	0.616561	0.059067
Walc	-0.093511	0.120244	0.388680	0.616561	1.000000	0.114988
health	0.109559	0.084526	-0.015741	0.059067	0.114988	1.000000
absences	-0.089534	-0.018716	0.085374	0.172952	0.156373	-0.030235
G1	0.048795	-0.094497	-0.074053	-0.195171	-0.155649	-0.051647
G2	0.089588	-0.106678	-0.079469	-0.189480	-0.164852	-0.082179
G3	0.063361	-0.122705	-0.087641	-0.204719	-0.176619	-0.098851

	absences	G1	G2	G3
age	0.149998	-0.174322	-0.107119	-0.106505
Medu	-0.008577	0.260472	0.264035	0.240151
Fedu	0.029859	0.217501	0.225139	0.211800
traveltime	-0.008149	-0.154120	-0.154489	-0.127173
studytime	-0.118389	0.260875	0.240498	0.249789
failures	0.122779	-0.384210	-0.385782	-0.393316
famrel	-0.089534	0.048795	0.089588	0.063361
freetime	-0.018716	-0.094497	-0.106678	-0.122705
goout	0.085374	-0.074053	-0.079469	-0.087641
Dalc	0.172952	-0.195171	-0.189480	-0.204719
Walc	0.156373	-0.155649	-0.164852	-0.176619
health	-0.030235	-0.051647	-0.082179	-0.098851
absences	1.000000	-0.147149	-0.124745	-0.091379
G1	-0.147149	1.000000	0.864982	0.826387
G2	-0.124745	0.864982	1.000000	0.918548
G3	-0.091379	0.826387	0.918548	1.000000

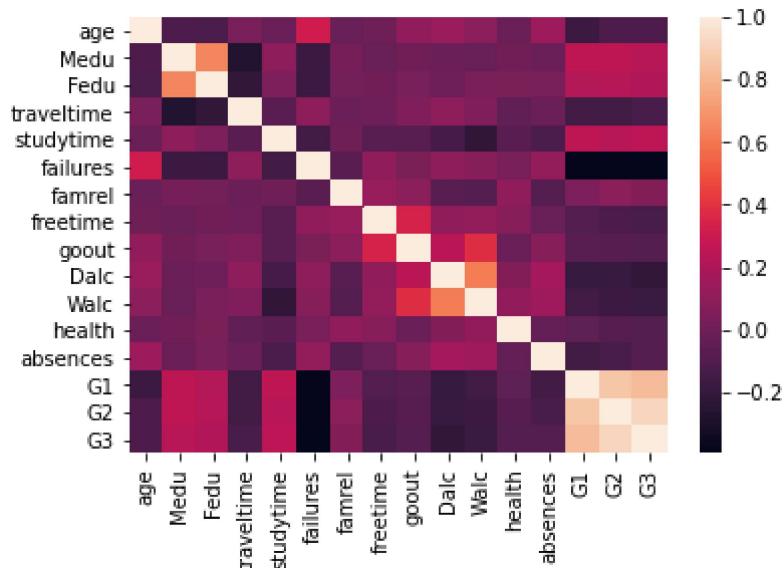
In [104...]: `s.heatmap(x,xticklabels=x.columns,yticklabels=x.columns,annot=True)`

Out[104...]: <AxesSubplot:>



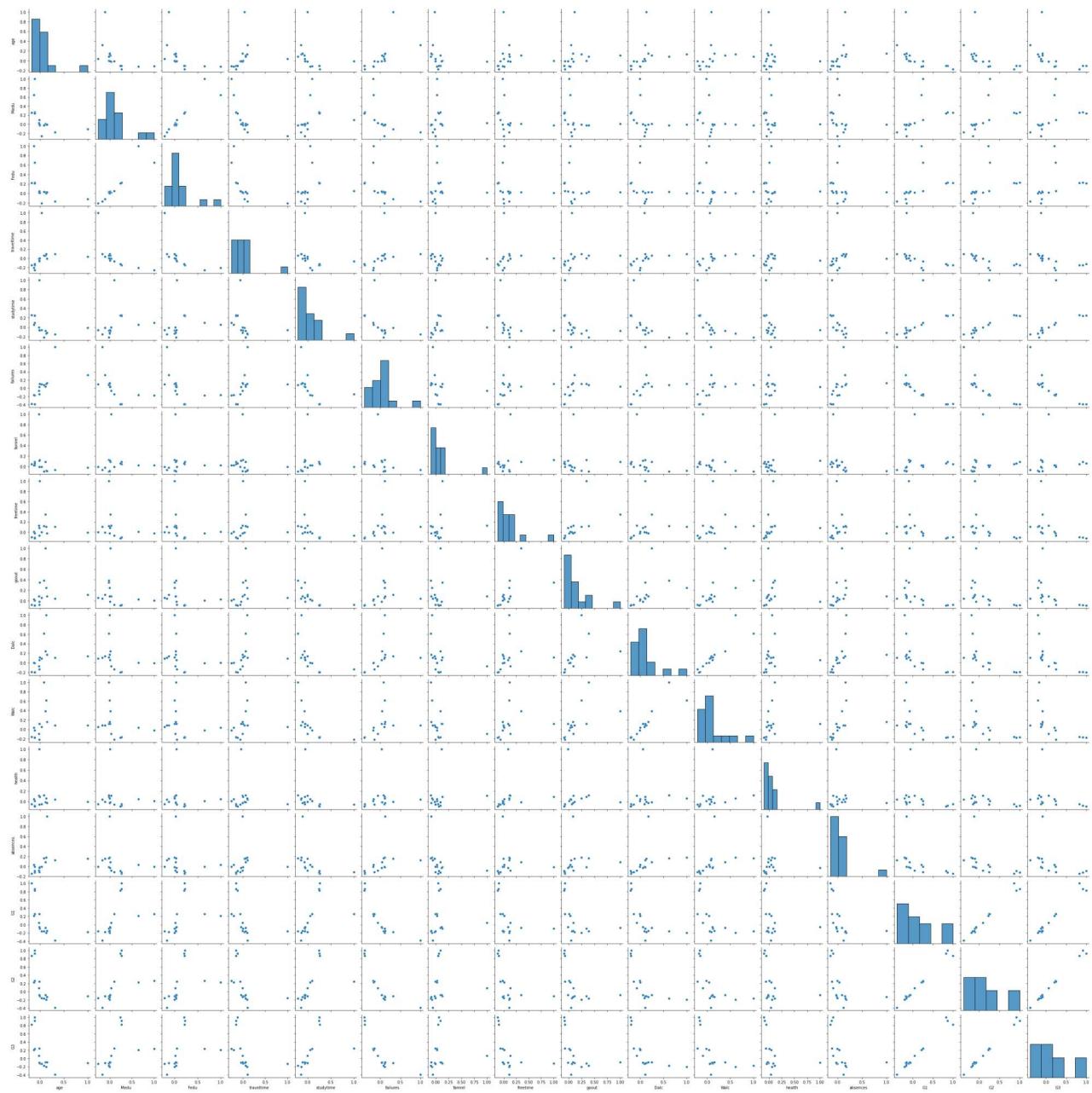
In [105...]: `s.heatmap(x,xticklabels=x.columns,yticklabels=x.columns,annot=False)`

Out[105...]: <AxesSubplot:>



In [99]:
s.pairplot(x)

Out[99]: <seaborn.axisgrid.PairGrid at 0x1a7516b6790>



In []: