CSL126 File

Name- Rishav Chopra
Class- Btech CSE 2nd sem
Roll no.- 01
submitted to- Dr. Butta singh

Airthmetic Operator:-

1. (+) Addition: Adds values on either side of operator.

In [1]:
```
a=6
b=4
x=a+b
print(x)
```

```
10
```

1. (-) Subtract: subtract right hand operand by left hand operand.

In [3]:
```
a=6
b=4
x=a-b
print(x)
```

```
2
```

1. (*) Multiplies: Multiplies on either side of operator.

In [4]:
```
a=6
b=7
x=a*b
print(x)
```

```
42
```

1. (/) Divison: Divides left hand operand by right hand operand

In [5]:
```
a=2
b=4
x=b/a
print(x)
```

```
2.0
```

1. (%) Modulus: Divides left hand operand with right hand and returns reminder

In [6]:
```
a=4
b=5
x=a%b
print(x)
```

```
4
```

1. (**) Exponents: Performs exponential (power) calculations on operators.

In [7]:
```
a=4
b=6
x=a**b
print(x)
```

```
4096
```

1. (//) Floor division: The division of operands where the result is the quotien in which the digits after decimal point are removed.

In [8]:
```
a=7
b=5
x=a//b
print(x)
```

```
1
```

Assignment Operator

1. (=) Equal opertor: Assignes values from right side oprands to left side operands.

In [9]:
```
x=4
x=a
print(a)
```

```
7
```

1. (+=) Add AND: Its adds right operands to the left operands and assigns the results.

In [10]:
```
x=7
y=6
x+=y
x
```

Out[10]:
```
13
```

1. (-=) SubtractAND: It subtracts right operand from the left operand and assign the results.

In [11]:
```
x=7
y=3
x-=y
x
```

Out[11]:
```
4
```

1. (*=) Multiply AND: Its multiplies right operand with left and prints the result.

In [12]:
```
x=5
y=6
x*=y
x
```

Out[12]:
```
30
```

1. (/=) Divide AND: It divides left operand with right operand and print the result.

In [13]:
```
x=7
y=6
x/=y
x
```

Out[13]:
```
1.1666666666666667
```

1. (**=) Exponent AND: Performs exponential (power) calculation to operator and assigm the values.

In [14]:
```
x=7
```

```
y=6
x**=y
x
```

Out[14]: 117649

1. (//=) Floor division: It performs division on operators and assign values toleft operator.

In [15]:
```
x=7
y=3
x//=y
x
```

Out[15]: 2

Comparison Operators

1. (==) Equal: If the values of two operands are equal, then conditions become true.

In [16]:
```
x="a"
y="b"
x==y
```

Out[16]: False

1. (!=) Not equal: If the values are not equal, the condiotions become true.

In [18]:
```
a="a"
b="b"
a!=b
```

Out[18]: True

1. (>) Greater than operator: If the value of left operand is greater than right operand, than the condition is true.

```
x=4
```

In [19]:
```
y=5
x>y
```

Out[19]:
```
False
```

1. (<) less than operator: If the value of left operand is less than right operand, then the condition is true.

In [20]:
```
x=4
y=8
x<y
```

Out[20]:
```
True
```

1. (>=) Greater than equal to: If the value of left operand is greater than or equal to the right hand, then the condition is true.

In [22]:
```
x=5
y=6
x>=y
```

Out[22]:
```
False
```

1. (<=) less than equal to: If the value of left operand is less than or equal to the right hand, then the condition is true.

In [23]:
```
x=5
y=6
x<=y
```

Out[23]:
```
True
```

Identity Operator

1. (is): Evaluates to true if the variable on either side of the operator point to the same object and false otherwise.

In [26]:
```
x="ram"
a=x
x is a
```

Out[26]: True

1. (is not): Evaluates to false if the variable on either side of the operator point to the same object and true otherwise.

In [28]:
```
x="ram"
y="laxman"
x is not y
```

Out[28]: True

Logical Operator

1. (AND) Logical AND: If both operands are true and condition becomes true.

In [30]:
```
x=6
x>5 and x<7
```

Out[30]: True

1. (OR) Logical OR: If any of the two operands are true

In [31]:
```
x=8
x>10 or x<20
```

Out[31]: True

1. (NOT) Logical NOT: Used to reverse the logic of operands.

In [37]:
```
x=7
x>7 and x<12
```

Out[37]: False

Membership Operators

1. (IN) in operator: Evaluates true if it finds out a variable in specified sequence otherwise false.

In [38]:
```
x="rishav"
y="r"
y in x
```

Out[38]: True

1. (NOT in) not in operator: Evaluates true if it does not finds out a variable in specified sequence otherwise false.

In [40]:
```
x="rishav"
y="z"
x not in y
```

Out[40]: True

Built in string meathods

1. String capitalize() method.

it returns the copy of string with only its first charcter captalized

In [46]:
```
str1="rishav"
str1.capitalize()
```

Out[46]: 'Rishav'

2. String center() method

the meathod center() returns centred in astring of length width. Padding is done usingfill char.

In [47]:
```
x="rishav"
x.center(20,"*")
```

Out[47]: '*******rishav*******'

3. String ends with() Method

it gives the result true if the string ends with specific suffix, otherwise false

In [53]:
```
x="rishav"
x.endswith("av",4,5)
```

Out[53]: False

In [49]:
```
x="rishav"
x.endswith("v")
```

Out[49]: True

4. String isalnum() Method

This checks wheather string consists of alnumeric characters

In [55]:
```
x="1234"
x.isalnum()
```

Out[55]: True

5. String isalpha() Method.

This checks wheather the string consists of alphabetical character.

In [57]:
```
x="rishav_chopra"
x.isalpha()
```

Out[57]: False

6. String isdigit() Method.

The meathod checks wheather the string consists of digits only.

In [58]:
```
x="1234"
x.isdigit()
```

Out[58]: True

7. String is lower() Method.

This checks wheather all are in lower case or not

In [60]:
```
x="rishav"
x.islower()
```

Out[60]: True

8. String isspace() Method.

This checks wheather the string consists of white spaces or not.

In [63]:
```
x=" "
x.isspace()
```

Out[63]: True

9. String isupper() Method

This meathod checks wheather string has uppercase characters.

In [65]:
```
x="BELLA CAIO"
x.isupper()
```

Out[65]: True

10. Stringjoin() Method

This meathod returns a string in which the string meathods are joined.

In [75]:
```
str1=["a","b","c","d","e","f"]
str1="*".join(str1)
print(str1)
```

```
a*b*c*d*e*f
```

11. String lower() Method.

This meathod returns the copy of string in which all case-based have been resolved.

In [80]:
```
x="RISHAV"
x.lower()
```

Out[80]:
```
'rishav'
```

12. Stringmax() Method.

The max() returns the alphabetical character from the string str.

In [82]:
```
str=("a","b","c")
x=max(str)
print(x)
```

```
c
```

13. String min() method.

The min() meathod the min alphabetial character from the string.

In [83]:
```
str=("a","b","c")
x=min(str)
print(x)
```

```
a
```

14. String replace() Method.

The replace() meathod returns a copy of the string in which the occurrences of old have been replaced with new.

In [89]:
```
x="i love this planet"
y=x.replace("planet","country")
print(y)
```

```
i love this country
```

15. String strip() Method

In [92]:
```
txt= "Banana"
x= txt.strip()
print("out of all fruits,", x ,"is my fav.")
```

```
out of all fruits, Banana is my fav.
```

16. String maketrans() Method.

In [93]:
```
txt="i like bananas"
y= txt.maketrans("b"," ")
print(txt.translate(y))
```

```
i like  ananas
```

17. string split() Method.

In [94]:
```
txt="Bella Caio"
txt.split()
```

Out[94]:
```
['Bella', 'Caio']
```

18. String Splitlines() Methods

In [99]:
```
txt= "i love\nmy country"
txt.splitlines()
```

Out[99]:
```
['i love', 'my country']
```

19. String Swapcase() Methods.

In [101…
```
txt=" Hello my Name is rishav"
txt.swapcase()
```

Out[101…
```
' hELLO MY nAME IS RISHAV'
```

20. string zfill() Method.

In [103…
```
txt="50"
x= txt.zfill(10)
print(x)
```

```
0000000050
```

21. string isdecimal() Method.

In [107…
```
x="23"
x.isdecimal()
```

Out[107… True

List

*List are defined using parantheses([list] and commas) *We can change lists in place. *List are mutable or modified.

In [110…
```
x=["a","1",23,"xyc"]
x
```

Out[110… ['a', '1', 23, 'xyc']

1. Joining List.

In [111…
```
lst1=[1,2,3,4]
lst2=[5,6,7,8]
lst1+lst2
```

Out[111… [1, 2, 3, 4, 5, 6, 7, 8]

2. Replicating or repeating list.

In [112…
```
lst1=[1,2,2,3,4]
s=lst1*3
```

In [113…
```
s
```

Out[113… [1, 2, 2, 3, 4, 1, 2, 2, 3, 4, 1, 2, 2, 3, 4]

3. Slicing the list.

In [114…
```
lst=["r","i","s","h","a","v"]
lst[0:2]
```

Out[114…
```
['r', 'i']
```

4. Append the list.

In [115…
```
lst1=[1,2,3,4,5]
lst1.append(6)
lst1
```

Out[115…
```
[1, 2, 3, 4, 5, 6]
```

In [118…
```
lst1=[1,2,3,7,5,6]
lst1[3]=4
```

In [117…
```
lst1
```

Out[117…
```
[1, 2, 3, 4, 5, 6]
```

5. The index method.

In [120…
```
lst1=[1,2,3,4,5,6,7,8]
lst1.index(3)
```

Out[120…
```
2
```

6. The Extend Method.

In [121…
```
lst1=["a","B","c"]
lst2=["D","e","f"]
lst1.extend(lst2)
```

In [122…
```
lst1
```

Out[122… ['a', 'B', 'c', 'D', 'e', 'f']

7. The insert Method.

In [123…
```
lst1=[1,2,3,4,5,6,7]
lst1.insert(8,9)
```

In [124…
```
lst1
```

Out[124… [1, 2, 3, 4, 5, 6, 7, 9]

8. The pop Method.

In [125…
```
lst1=["a","b","c","d"]
lst1.pop()
lst1
```

Out[125… ['a', 'b', 'c']

9. The remove Method

In [126…
```
x=[1,2,3,4,5,6,7]
x.remove(2)
```

In [127…
```
x
```

Out[127… [1, 3, 4, 5, 6, 7]

10. The clear Method

In [128…
```
lst=[1,2,3,4,5,6,7]
lst.clear()
```

In [129…
```
lst
```

Out[129… []

11. The count Method

In [132…
```
x=[1,2,3,4,5,6,7]
x.count(5)
```

Out[132…
```
1
```

12. The reverse Method

In [133…
```
x=[1,2,3,4,5,6]
x.reverse()
```

In [134…
```
x
```

Out[134…
```
[6, 5, 4, 3, 2, 1]
```

13. The sort Method

In [140…
```
lst=[1,2,3,4,7,5,4,]
lst.sort()
```

In [141…
```
lst
```

Out[141…
```
[1, 2, 3, 4, 4, 5, 7]
```

14. Sorting in reverse order

In [139…
```
lst.sort(reverse=True)
```

In [142…
```
lst
```

Out[142…
```
[1, 2, 3, 4, 4, 5, 7]
```

Dictionary

Keys can be any immutable type. Values can be any type. A single dictionary can store values of diff types.

In [1]:
```
x={"user":"rishav" , "password":"root312"}
x["user"]
```

Out[1]:
```
'rishav'
```

In [2]:
```
x["password"]
```

Out[2]:
```
'root312'
```

In [3]:
```
x["user"]="raj"
x
```

Out[3]:
```
{'user': 'raj', 'password': 'root312'}
```

In [4]:
```
Dict2={"user":"rishav" , "password":"root312" ,"department":"CSE" , "Mark_math": 92}
```

In [6]:
```
Dict2["Mark_math"]=95
Dict2
```

Out[6]:
```
{'user': 'rishav',
 'password': 'root312',
 'department': 'CSE',
 'Mark_math': 95,
 'Mark math': 95}
```

In [7]:
```
Dict2["mark physics"]=87
```

In [8]:
```
Dict2
```

Out[8]:
```
{'user': 'rishav',
 'password': 'root312',
 'department': 'CSE',
 'Mark_math': 95,
 'Mark math': 95,
 'mark physics': 87}
```

Del function

In [9]:
```
dict2={"user":"rishav" , "password":"root312" ,"department":"CSE" , "Mark_math": 92}
```

In [19]:
```
del dict2
```

In [20]:
```
dict2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-20-522e1a9638e7> in <module>
----> 1 dict2

NameError: name 'dict2' is not defined
```

Clear funtion

In [22]:
```
dict2={"user":"rishav" , "password":"root312" ,"department":"CSE" , "Mark_math": 92}
```

In [23]:
```
dict2.clear()
```

In [16]:
```
dict2
```

Out[16]:
```
{}
```

Key values and item Method

In [24]:
```
dict2={"user":"rishav" , "password":"root312" ,"department":"CSE" , "Mark_math": 92}
```

In [25]:
```
dict2.keys()
```

Out[25]:
```
dict_keys(['user', 'password', 'department', 'Mark_math'])
```

In [26]:
```
dict2.values()
```

Out[26]:
```
dict_values(['rishav', 'root312', 'CSE', 92])
```

In [28]:
```
dict2.items()
```

Out[28]:
```
dict_items([('user', 'rishav'), ('password', 'root312'), ('department', 'CSE'), ('Mark_math', 92)])
```

In [29]:
```
dict1=dict2.items()
```

In [30]:
```
dict1
```

Out[30]:
```
dict_items([('user', 'rishav'), ('password', 'root312'), ('department', 'CSE'), ('Mark_math', 92)])
```

Str() Functions

In [31]:
```
s=str(dict2)
```

In [32]:
```
s
```

Out[32]:
```
"{'user': 'rishav', 'password': 'root312', 'department': 'CSE', 'Mark_math': 92}"
```

In [33]:
```
s[0]
```

Out[33]:
```
'{'
```

Copy() Methods

In [34]:
```
dict3=dict2.copy()
```

In [35]:
```
dict3
```

Out[35]:
```
{'user': 'rishav', 'password': 'root312', 'department': 'CSE', 'Mark_math': 92}
```

In [36]:
```
dict2.clear()
dict2
```

Out[36]:
```
{}
```

In [37]:
```
dict3
```

Out[37]:
```
{'user': 'rishav', 'password': 'root312', 'department': 'CSE', 'Mark_math': 92}
```

Fromkeys() Meathod

In [40]:
```
s=["ram","age","gender"]
dict5=dict.fromkeys(s)
```

In [41]:
```
dict5
```

Out[41]:
```
{'ram': None, 'age': None, 'gender': None}
```

In [43]:
```
dict5=dict.fromkeys(s,1)
```

In [44]:
```
dict5
```

Out[44]:
```
{'ram': 1, 'age': 1, 'gender': 1}
```

Get() Method

In [45]:
```
dict5
```

Out[45]:
```
{'ram': 1, 'age': 1, 'gender': 1}
```

In [46]:
```
dict5.get("ram")
```

Out[46]:
```
1
```

In [49]:
```
dict3
```

Out[49]:
```
{'user': 'rishav', 'password': 'root312', 'department': 'CSE', 'Mark_math': 92}
```

In [50]:
```
dict3.get("Mark_math")
```

Out[50]:
```
92
```

In [52]:
```
sd=dict3.get("height")
```

In [53]:
```
print(sd)
```

```
None
```

In [54]:
```
x={"height":"6.0 feet"}
```

In [55]:
```
dict3.update(x)
```

In [56]:
```
dict3
```

Out[56]:
```
{'user': 'rishav',
 'password': 'root312',
 'department': 'CSE',
 'Mark_math': 92,
 'height': '6.0 feet'}
```

1.) WAP to offer discount (D) on purchase (p)

In [64]:
```
p=int(input('total purchase'))
if p>1000:
        d=p*0.1
        print('congrats you have won a discount of =',d)
        print('total amount=',p-d,'total saving=',d)
elif p>700 and p<1000:
            d=p*0.05
            print('congrats you have won a discount of =',d)
```

```
total purchase1234
congrats you have won a discount of = 123.4
total amount= 1110.6 total saving= 123.4
```

2.) WAP to accept a number between 1 and 100 and check it is odd or even

In [72]:
```
 x=int(input('enter a number of your wish'))
if x>=1 and x<=100:
        if x%2==0:
            print(x,'even number')
elif x>100:
        print('out of range')
```

```
else:
    print('odd number')
```

```
enter a number of your wish78
78 even number
```

3.) WAP to accept a number between 1 and 100 and check using nested if statement

Number is divisible by 2 and 3 Number is divisible by 2 but not by 3 Number is divisible by 3 but not by2 Number is not divisible by 2 and 3

In [77]:
```
n=int(input("Enter a number "))
if n>1 and n<100:
    if n%2==0 and n%3==0:
        print("Given number is divisible by 2 and 3 ")
    elif n%2==0 and n%3!=0:
        print("Given number is divisible by 2 but not by 3 ")
    elif n%2!=0 and n%3==0:
        print("Given number is not divisible by 2 but is divisible by 3 ")
    else:
        print("Given number is nither divisible by2 nor by 3 ")
else:
    print("Given number is invalid")
```

```
Enter a number 34
Given number is divisible by 2 but not by 3
```

4.) WAP to check given char is vowel.

In [83]:
```
n=input("Enter a single alphabetical character: ")
if n=="a" or n=="e" or n=="i" or n=="o" or n=="u":
    print("Given alphabet is vowel ")
else:
    print("Given alphabet is a consonent")
```

```
Enter a single alphabetical character: R
Given alphabet is a consonent
```

5.) WAP to transpose a matrix.

In [84]:
```
s=[[1,2,3],[4,3,4],[4,3,6]]
trans=[[0,0,0],[0,0,0],[0,0,0]]
for i in range(0,len(s)):
    for j in range(0,len(s[0])):
        trans[j][i]=s[i][j]
print(trans)
```

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

6.) WAP to add a matrix.

In [89]:
```
a=[[1,2,3],[4,5,6]]          # addition of two matrices
b=[[7,8,9],[1,5,6]]
sum1=[[0,0,0],[0,0,0]]
for i in range(0,len(a)):
    for j in range(0,len(a[0])):
        sum1[i][j]=a[i][j]+b[i][j]
print(sum1)
```

```
[[8, 10, 12], [5, 10, 12]]
```

7.) Table of variable n.

In [1]:
```
n=int(input("Enter a number: "))
i=1
for i in range(1,21):
    print(n,"x",i,"=", i*n)
```

```
Enter a number: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
```

```
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84
7 x 13 = 91
7 x 14 = 98
7 x 15 = 105
7 x 16 = 112
7 x 17 = 119
7 x 18 = 126
7 x 19 = 133
7 x 20 = 140
```

8.) WAP Print a table using While:

In [4]:
```
n=int(input("Enter a number: "))
i=1
while i<=10:
    print(n,"x",i,"=",i*n)
    i=i+1
```

```
Enter a number: 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
```

9.) WAP to print list in reverse order.

In [5]:
```
lst=[1,2,3,4,5,6,7,8]
for i in range(len(lst)-1,-1,-1):
    print(lst[i])
```

```
8
7
```

```
6
5
4
3
2
1
```

10.) WAP using break statement.

In [10]:
```
lst=["p","y","t","h"]
for i in range(0,len(lst)):
    print(lst[i])
    if lst[i]=="h":
        break
```

```
p
y
t
h
```

11.) WAP to print pythagorean numbers.

In [12]:
```
def pythagorean_function(n):
    import math
    for a in range(1,n+1):
        for b in range(1,n+1):
            c=a**2+b**2
            c_sqrtint=int(math.sqrt(c))
            c_sqrt=math.sqrt(c)
            if (c_sqrtint-c_sqrt==0):
                print(a,b,c_sqrtint)
```

In [13]:
```
pythagorean_function(10)
```

```
3 4 5
4 3 5
```

```
6 8 10
8 6 10
```

12.) WAP to test pythagoras theorem

In [14]:
```
def pythagoras_func(a,b):
        c=a**2+b**2
        import math
        c_sqrt=math.sqrt(c)
        if c_sqrt-int(c_sqrt)==0:
            print('its a pythagoras triangle')
        else:
            print('not a pythagoras triangle')
```

In [15]:
```
pythagoras_func(3,4)
```

```
its a pythagoras triangle
```

13.) WAP to print fibonacci series

In [16]:
```
def fibonacci_ser(n):
    a,b=0,1
    result=[]
    while b<n:
        result.append(b)
        a,b=b,a+b
    print(result)
```

In [17]:
```
fibonacci_ser(30)
```

```
[1, 1, 2, 3, 5, 8, 13, 21]
```

14.) WAP to check given year is a leap and print a suitable message correctly

In [19]:
```
def leap_yr(n):
```

```
    if n%4==0:
        print(n,'is a leap year')
    else:
        print(n,'not a leap year')
```

In [20]:
```
leap_yr(2020)
```

```
2020 is a leap year
```

15.) WAP to find the sum of odd and even numbers up to a user defined range

In [26]:
```
def number_num(n):
        e=0
        o=0
        for i in range(1,n):
            if i%2==0:
                print('even number',i)
                e=e+i
            else:
                print('odd number',i)
                o=o+i
        print(e)
        print(o)
```

In [27]:
```
number_num(9)
```

```
odd number 1
even number 2
odd number 3
even number 4
odd number 5
even number 6
odd number 7
even number 8
20
16
```

16.) WAP to find the area and perimeter of a circle and rectangle

In [28]:
```
def perimeter_per(x,y):
    p=2*x+2*y
    print(p)
perimeter_per(5,6)
```

```
22
```

In [29]:
```
def area_ar(x,y):
    a=x*y
    print(a)
```

In [30]:
```
area_ar(5,6)
```

```
30
```

In [32]:
```
import math
math.pi
```

Out[32]:
```
3.141592653589793
```

In [33]:
```
def perim_eter(r):
    p=2*math.pi*r
    print(p)
```

In [34]:
```
perim_eter(5)
```

```
31.41592653589793
```

In [35]:
```
def area_ar(r):
    a=math.pi*r*r
    print(a)
```

In [36]:
```
area_ar(5)
```

```
78.53981633974483
```

17.) WAP to convert temprature into celcius to fahreheit and vice

versa.

In [2]:
```
print("Press1, for converting celcius into fahrenheit: ")
print("Press 2, for converting fahrenheit into celcius: ")
choice=int(input("Enter your choice: "))
temp=int(input("Enter the temprature: "))
if choice==1:
    print((temp*1.8)+32,"F")
elif choice==2:
    print((temp-32)*555.6,"c")
else:
    print("invalid choice")
```

```
Press1, for converting celcius into fahrenheit:
Press 2, for converting fahrenheit into celcius:
Enter your choice: 1
Enter the temprature: 23
73.4 F
```

18.) Given a point(x,y). Write a program to find whether to find it lies in the first,second,third or fourth quadrant of x-y plane

In [3]:
```
def poin_ts(x,y):
    if x>0 and y>0:
        print('1st quadrant')
    elif x<0 and y>0:
        print('2nd quadrant')
    elif x<0 and y<0:
        print('3rd quadrant')
    elif x>0 and y<0:
        print('4th quadrant')
```

In [5]:
```
poin_ts(5,6)
```

```
1st quadrant
```

19.) WAP To check if items in the list are sorted in ascending order

and print suitable message accordingly.

In [6]:
```
lst=[1,2,3,4,5,6,7,8]
sort1=lst[:]
sort1.sort()
if sort1==lst:
    print("Given list is already sorted")
else:
    print("Given list is unsorted")
```

```
Given list is already sorted
```

20.) WAP to check prime number upto user defined range

In [7]:
```
def prime(n,m):
    if (n==m+1):
        return
    if (n==2 or n%2!=0):
        print(n)
    prime(n+1,m)
```

In [8]:
```
prime(2,12)
```

```
2
3
5
7
9
11
```

21.) WAP to count total number of vowels,consonants and blanks in a string.

In [9]:
```
 def str_ing(str):
        v=0
        b=0
        c=0
```

```
        for i in str:
            if i=='a' or i=='e' or i=='i'or i=='o' or i=='u':
                v=v+1
            elif i==' ':
                b=b+1
            else:
                c=c+1
        print('number of vowels',v)
        print('number of blanks',b)
        print('number of consonants',c)
```

In [10]:
```
str_ing("bella caio")
```

```
number of vowels 5
number of blanks 1
number of consonants 4
```

22.) WAP to find total number of digits,uppercase,lowercaseletter in a sentence

In [11]:
```
def string_sr(str):
    u=0
    l=0
    d=0
    for i in str:
        if i.isupper():
            u=u+1
        elif i.islower():
            l=l+1
        elif i.isdigit():
            d=d+1
    print('number of uppercase',u)
    print('number of lowercase',l)
    print('number of digits',d)
```

In [13]:
```
string_sr("Bella caio 143")
```

```
number of uppercase 1
number of lowercase 8
number of digits 3
```

22.) WAP to print prime numbers between 2 and 100

In [15]:
```
for i in range(2,100):
    for j in range(2,i+1):
        if i%j==0:
            break
    if j==i:
        print(i)
```

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

23.) WAP to find factorial of a given number

In [3]:
```
fact=int(input("enter the number: "))
fact_1=1
i=1
```

```
while i<=fact:
    fact_1*=i
    i+=1
print("Factorial of given number is: ", fact_1)
```

```
enter the number: 4
Factorial of given number is:  24
```

24.) WAP to find the largest number among three numbers.

In [4]:
```
n1=int(input('enter first number'))
n2=int(input('enter second number'))
n3=int(input('enter third number'))
if n1>n2 and n1>n3:
    print('largest number is',n1)
elif n2>n1 and n2>n3:
    print('largest number is',n2)
else:
    print('largest number is',n3)
```

```
enter first number8
enter second number3
enter third number6
largest number is 8
```

Functions

In [7]:
```
def sum_1(a,b,c):
    s=a+b+c
    m=a*b*c
    d1=a/b
    d2=b/c
    print("sum=",s,"multiplication=",m,"div1=",d1,"div2=",d2)
```

In [8]:
```
x=1
y=2
z=3
sum_1(x,y,z)
```

```
sum= 6 multiplication= 6 div1= 0.5 div2= 0.6666666666666666
```

In [9]:
```
def sum_1 (a,b,c):
    s=a+b+c
    m=a*b*c
    d1=a/b
    d2=b/c
    print("sum=",s,"multiplication=",m,"div1=",d1,"div2-",d2)
    return(s,m,d1,d2)
```

In [10]:
```
[s1,s2,s3,s4]=sum_1(2,3,4)
```

```
sum= 9 multiplication= 24 div1= 0.6666666666666666 div2- 0.75
```

In [11]:
```
s1,s2,s3,s4
```

Out[11]:
```
(9, 24, 0.6666666666666666, 0.75)
```

In [14]:
```
def funct_1(a,b,*c):
    print(a)
    print(b)
    print(c)
```

In [15]:
```
funct_1(2,3,4)
```

```
2
3
(4,)
```

In [16]:
```
funct_1(2,3,4,5,6,7)
```

```
2
3
(4, 5, 6, 7)
```

In [ ]:
```
```