



VIDYAVAHINI GROUP OF INSTITUTIONS

VIDYAVAHINI FIRST GRADE & PG COLLEGE

— AFFILIATED TO TUMKUR UNIVERSITY —

DEPARTMENT OF bca

VI-semester

LAB MANUAL

for

PYTHON PROGRAMMING LAB

1. Program to convert the given temperature from Fahrenheit to Celcius and vice versa depending upon user's choice.

while True:

```
print("1: Celcius to Fahrenheit")
print("2: Fahrenheit to Celcius ")
ch=int(input("Enter your choice"))
if ch==1:
    tmp=float(input("Enter temperature in Celcius"))
    F=tmp * 9/5 + 32
    print("%f Celcius = %f Fahrenheit"%(tmp, F))
if ch==2:
    tmp=float(input("Enter temperature in Fahrenheit"))
    c=(tmp - 32) * 5/9
    print("%f Fahrenheit = %f Celcius"%(tmp, c))
if ch>=3:
    break;
```

OUTPUT:

```
1: Celcius to Fahrenheit
2: Fahrenheit to Celcius
3: Exit
Enter your choice 1
Enter temperature in Celcius 25
25.000000 Celcius = 77.000000 Fahrenheit
1: Celcius to Fahrenheit
2: Fahrenheit to Celcius
3: Exit
Enter your choice 3
Process finished with exit code 0
```

- 2. Program to calculate total marks, percentage and grade of student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:**

Grade A: Percentage ≥ 80

Grade B: Percentage ≥ 70 or ≥ 60

Grade C: Percentage ≥ 60 or ≥ 40

Grade D: Percentage ≥ 40

Grade E: Percentage < 40

```
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
omarks=sub1+sub2+sub3;
mmarks=300
print("secured marks is:",omarks)
per=(omarks/mmmarks)*100
print("percentage os student is:",per)
if(per>=80):
    print("Grade: A")
elif(per>=70 or per==60):
    print("Grade: B")
elif(per>=60 or per==40):
    print("Grade: C")
elif(per>=40):
    print("Grade: D")
else:
    print("Grade: E")
```

OUTPUT:

```
Enter marks of the first subject: 40
Enter marks of the second subject: 50
Enter marks of the third subject: 60
secured marks is: 150
percentage os student is: 50.0
Grade: D
```

```
Process finished with exit code 0
```

3. Program, using user defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.

```
def arect(l,b):
    area=l*b
    print("area of rectangle is:",area)
    return

def acircle(r):
    area=3.142*r*r
    print("area of circle is:",area)
    return

def asquare(s):
    area=s*s
    print("area of square is:",area)
    return

def atri(b,h):
    area=0.5*b*h
    print("area of traingle is:",area)
    return

l=int(input("enter length"))
b=int(input("enter breadth"))
r=int(input("enter radius"))
s=int(input("enter side"))
h=int(input("enter height"))
arect(l,b)
acircle(r)
asquare(s)
atri(b,h)
```

OUTPUT:

enter length 3

enter breadth 2

enter radius 2

enter side 4

enter height 6

area of rectangle is: 6

area of circle is: 12.568

area of square is: 16

area of traingle is: 6.0

Process finished with exit code 0

4. Program to generate Pythagorean Triplets.

Solution: $a^2 + b^2 = c^2$

We can see that $a^2 + b^2 = c^2$, so instead of iterating for a, b and c we can iterate for m and n and can generate these triplets.

$$\begin{aligned}a &= m^2 - n^2 \\b &= 2 * m * n \\c &= m^2 + n^2\end{aligned}$$

```
limit=int(input("enter the upperlimit"))
c=0;
m=2;
while(c<limit):
    for n in range(1,m+1):
        a=m*m-n*n
        b=2*m*n
        c=m*m+n*n
        if(c>limit):
            break
        if(a==0 or b==0 or c==0):
            break
        print(a,b,c)
    m=m+1
```

OUTPUT:

enter the upperlimit 20

3 4 5

8 6 10

5 12 13

15 8 17

12 16 20

Process finished with exit code 0

5. Program to display the first n terms of Fibonacci series.

```
N=int(input("Enter the number of terms:"))
n1=0
n2=1
i=2
if N<=0:
    print("Enter a positive integer..")
elif N==1:
    print("Fibonacci series up to",N,":")
    print(n1)
else:
    print("Fibonacci series up to",N,":")
    print(n1)
    print(n2)
while i<N:
    n3=n1+n2
    print(n3)
    n1=n2
    n2=n3
    i=i+1
```

OUTPUT

Enter the number of terms: 5

Fibonacci series up to 5 :

0

1

1

2

3

Process finished with exit code 0

6. Program to check whether a given number is Armstrong number or not.

```
num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

OUTPUT:

1.

Enter a number: 123

123 is not an Armstrong number

Process finished with exit code 0

2.

Enter a number: 153

153 is an Armstrong number

Process finished with exit code 0

7. Program to find sum of the following series for n terms: $1 - 2/2! + 3/3! - 4/4! + \dots + n/n!$

```
t=[]
total=0
t=[]
def fact(num):
    f=1.0
    while(num>0):
        f=f*num
        num=num-1
    return(f)
n=int(input("enter no of terms"))
for x in range(1,n+1):
    terms=x/(fact(x))
    if(x%2==0):
        terms=-terms
    t.append(terms)
    total=total+terms
print(t)
print("the sum of series is:",total)
```

OUTPUT:

```
enter no of terms 4
[1.0, -1.0, 0.5, -0.16666666666666666]
the sum of series is: 0.33333333333333337
```

Process finished with exit code 0

8. Program to calculate the sum of two compatible matrices.

```
rows = int(input("Enter the Number of rows : " ))
column = int(input("Enter the Number of Columns: "))

print("Enter the elements of First Matrix:")
matrix_a= [[int(input()) for i in range(column)] for i in range(rows)]
print("First Matrix is: ")
for n in matrix_a:
    print(n)

print("Enter the elements of Second Matrix:")
matrix_b= [[int(input()) for i in range(column)] for i in range(rows)]
for n in matrix_b:
    print(n)

result=[[0 for i in range(column)] for i in range(rows)]

for i in range(rows):
    for j in range(column):
        result[i][j] = matrix_a[i][j]+matrix_b[i][j]

print("The Sum of Above two Matrices is : ")
for r in result:
    print(r)
```

OUTPUT:

Enter the Number of rows : 2

Enter the Number of Columns: 2

Enter the elements of First Matrix:

2

3

4

5

First Matrix is:

[2, 3]

[4, 5]

Enter the elements of Second Matrix:

2

4

5

6

[2, 4]

[5, 6]

The Sum of Above two Matrices is :

[4, 7]

[9, 11]

Process finished with exit code 0

9. Program to calculate the product of two compatible matrices.

```
r1=int(input("enter the number of rows for first matrix"))
c1=int(input("enter the number of cols for first matrix"))
r2=int(input("enter the number of rows for second matrix"))
c2=int(input("enter the number of cols for second matrix"))
if(c1!=r2):
    print("order mismatch, multiplication is not possible")
    exit()
matA=[[0 for j in range(0,c1)] for i in range(0,r1)]
matB=[[0 for j in range(0,c2)] for i in range(0,r2)]
result=[[0 for j in range(0,c2)] for i in range(0,r1)]
print("enter the elements of first matrix")
for i in range(0,r1):
    for j in range(0,c1):
        matA[i][j]=int(input("enter elements"))
print("enter the elements of second matrix")
for i in range(0,r2):
    for j in range(0,c2):
        matB[i][j]=int(input("enter elements"))
for i in range(len(matA)):
    for j in range(len(matB[0])):
        for k in range(len(matB)):
            result[i][j]=result[i][j]+matA[i][k]*matB[k][j]
print("resultant matrix after multiplication")
for i in range(len(result)):
    for j in range(len(result[0])):
        print("\t",result[i][j],end=" ")
    print()
```

OUTPUT:

```
enter the number of rows for first matrix 2
enter the number of cols for first matrix 2
enter the number of rows for second matrix 2
enter the number of cols for second matrix 2
enter the elements of first matrix
enter elements 2
enter elements 3
enter elements 3
enter elements 4
enter the elements of second matrix
enter elements 2
enter elements 3
enter elements 4
enter elements 6
resultant matrix after multiplication
    16    24
    22    33
```

Process finished with exit code 0

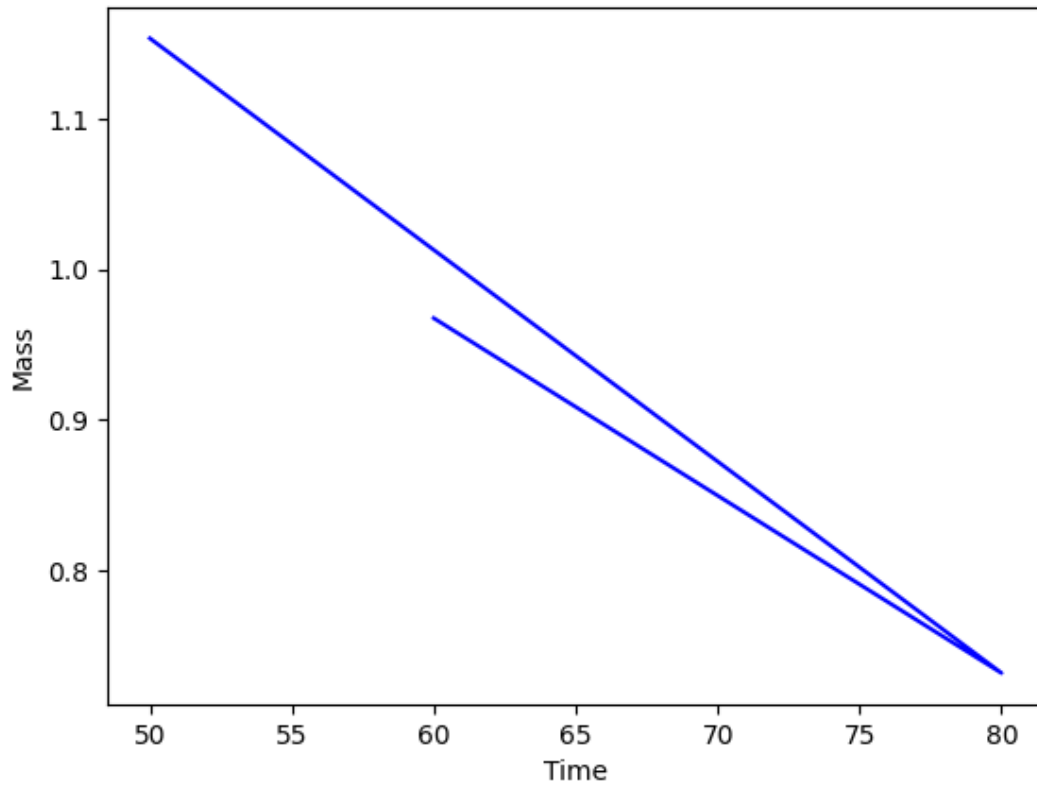
- 10. Program to calculate the mass "m" in a chemical reaction. The mass m (in gms) varies according to the formula $m=60/(t+2)$, where "t" is the time in hours. Sketch a graph for t vs. m where $t \geq 0$**

```
import matplotlib.pyplot as plt
tlist=[]
mlist=[]
interval=int(input("Enter interval"))
for i in range(1,interval+1):
    t=int(input("Enter tvalue"))
    tlist.append(t)
    m=60/(t+2)
    mlist.append(m)
print ("corresponding mass value:",m)
plt.plot(tlist,mlist,'blue')
plt.ylabel('Mass')
plt.xlabel('Time')
plt.show()
```

OUTPUT:

```
Enter interval 3
Enter tvalue 50
Enter tvalue 80
Enter tvalue 60
corresponding mass value: 0.967741935483871
```

Figure 1



11. Input initial velocity and acceleration and plot the following graphs.

I. Velocity With Respect To Time ($v=u+at$)

II. Distance With Respect To Time ($s=u*t + 0.5*a*t*t$)

III. Distance With Respect To Velocity ($s=(v*v-u*u)/(2*a)$)

```
import matplotlib.pyplot as plt

t=0
vf=[]
d=[]

def velocity(invel,accel):
    t=0
    if(invel>=0 and accel>=0):
        for t in range(0,10):
            v=u+a*t
            t+=10
            print(v)
            vf.append(v)
        print(vf)
        plt.plot(vf)
        plt.ylabel('some numbers')
        plt.show()
    else:
        print("enter positive values")

def disttime(invel,accel):
    if(invel>=0 and accel>=0):
        for t in range(0,10):
            s=(u*t+0.5*a*t*t)
            t+=10
            print(s)
            d.append(s)
        print(d)
```

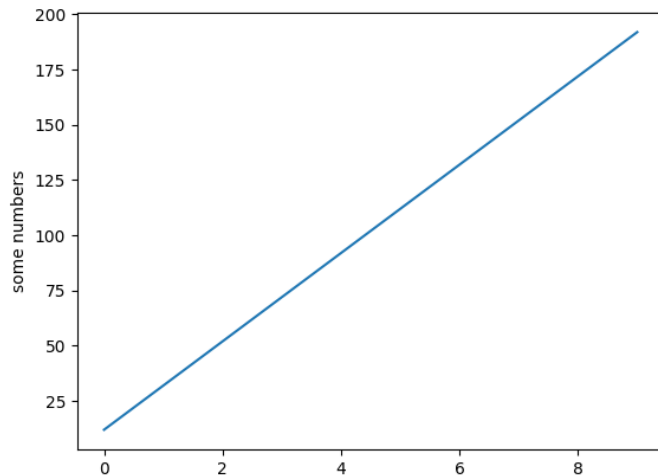


```
plt.plot(d)
plt.ylabel('some numbers')
plt.show()
else:
    print("enter positive values")
def distvel(invel,accel):
    if(invel>=0 and accel>=0):
        for v in range(0,10):
            s=(v*v-u*u)/(2*a)
            v+=10
            print(s)
            d.append(s)
        print(d)
        plt.plot(d)
        plt.ylabel('some numbers')
        plt.show()
    else:
        print("enter positive values")
u=int(input("enter initial velocity"))
a=int(input("enter the accelaration"))
while True:
    print ("1. velocity wrt to time")
    print ("2. distance wrt to time")
    print ("3. distance wrt to velocity")
    print ("4. exit")
    ch=int(input("enter ur choice"))
    if ch==1:
        velocity(u,a)
    elif ch==2:
        disttime(u,a)
```

```
elif ch==3:  
    distvel(u,a)  
elif ch==4:  
    exit(0)
```

OUTPUT:

```
enter initial velocity 12  
enter the accelaration 20  
1. velocity wrt to time  
2. distance wrt to time  
3. distance wrt to velocity  
4. exit  
enter ur choice 1  
12  
32  
52  
72  
92  
112  
132  
152  
172  
192  
[12, 32, 52, 72, 92, 112, 132, 152, 172, 192]
```



1. velocity wrt to time
2. distance wrt to time
3. distance wrt to velocity
4. exit

enter ur choice2

0.0

22.0

64.0

126.0

208.0

310.0

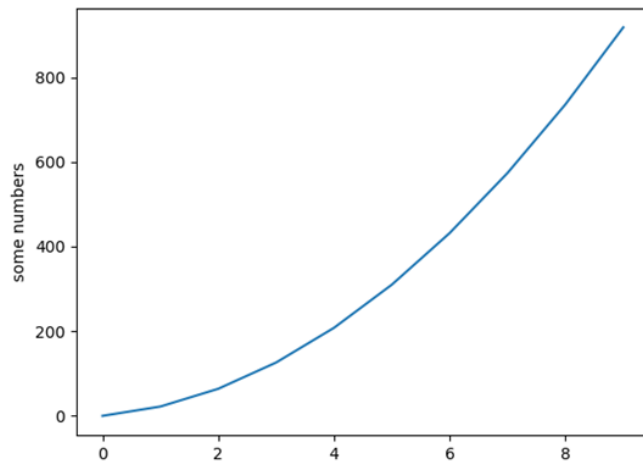
432.0

574.0

736.0

918.0

[0.0, 22.0, 64.0, 126.0, 208.0, 310.0, 432.0, 574.0, 736.0, 918.0]



1. velocity wrt to time
2. distance wrt to time
3. distance wrt to velocity
4. exit

enter ur choice3

-3.6

-3.575

-3.5

-3.375

-3.2

-2.975

-2.7

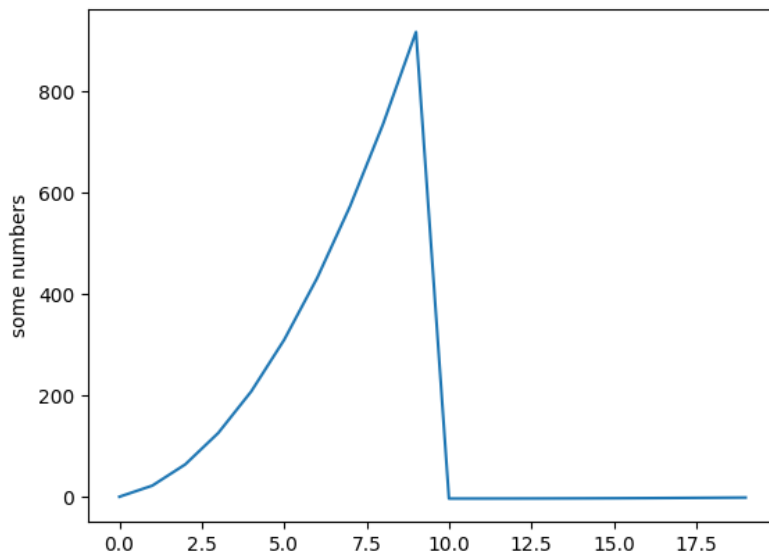
-2.375

-2.0

-1.575

[0.0, 22.0, 64.0, 126.0, 208.0, 310.0, 432.0, 574.0, 736.0, 918.0, -3.6, -3.575, -3.5, -3.375, -3.2, -2.975, -2.7, -2.375, -2.0, -1.575]

Figure 1



1. velocity wrt to time
2. distance wrt to time
3. distance wrt to velocity
4. exit

enter ur choice 4

Process finished with exit code 0

- 12. A population of 1000 bacteria is introduced into a nutrient medium. The population "p" grows as follows:**

$P(t) = (15000(1+t))/(15+e)$ where the time "t" is measured in hours. Write a Program to determine the size of the population at given time "t" and plot a graph for "p" vs "t" for the specified time interval.

```
import matplotlib.pyplot as plt

tlist=[]

plist=[]

e=1000

inter=int(input("enter number of intervals"))

size=int(input("enter size of each interval"))

for i in range (0,inter):

    t=int(i)*size

    print(t)

    tlist.append(t)

    p=(15000*(1+t))/(15+e)

    plist.append(p)

plt.plot(tlist,plist)

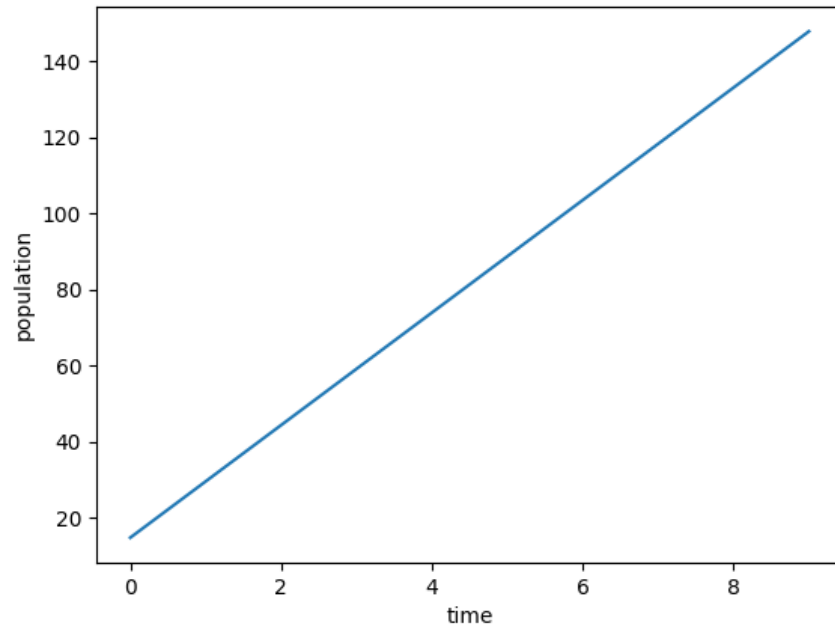
plt.xlabel("time")

plt.ylabel("population")

plt.show()
```

OUTPUT:

```
enter number of intervals 4
enter size of each interval 3
0
3
6
9
```



Process finished with exit code 0
