

```
#functions and inputs
#function create - call as many time u wanted
def greet():
    print("Hello,welcome to python")
greet()

def greeting(name):
    print("Hello",name)

greeting("sowmya")
greeting("johndoe")
```

```
→ Hello,welcome to python
Hello sowmya
Hello johndoe
```

```
def check_age(age):
    if age >= 18:
        return "your adult"
    else:
        return "your not adult"

result = check_age(20)
print(result)
```

```
→ your adult
```

```
name = input("enter your name")
print("hello",name)
age = int(input("enter your age"))
print("your age is",age)
height = float(input("enter your height"))
print("your height is",height)
```

```
→ enter your namesow
hello sow
enter your age56
your age is 56
enter your height78
your height is 78.0
```

```
def calculate_grade(marks):
    if marks >= 90:
        return "A"
    elif marks >= 80:
        return "B"
    elif marks >= 70:
        return "C"
    elif marks >= 60:
        return "D"
    else:
        return "F"

print("Student Grade Calculator")
name = input("enter your name")
print("hello",name)
marks = int(input("enter your marks"))
grade = calculate_grade(marks)
print(f"\n{name},your grade is {marks}")
print("your grade is",grade)
```

```
→ Student Grade Calculator
enter your namesowmya
hello sowmya
enter your marks67

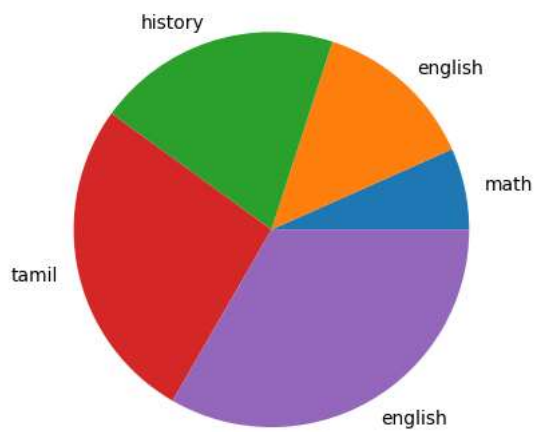
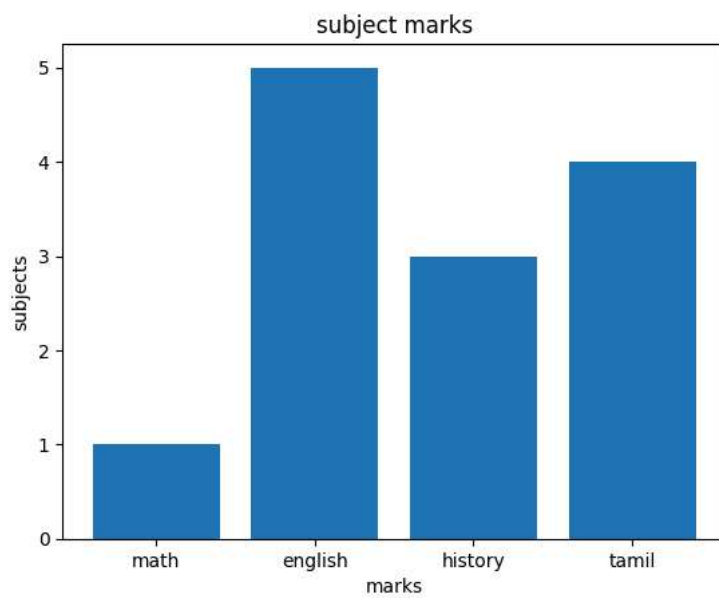
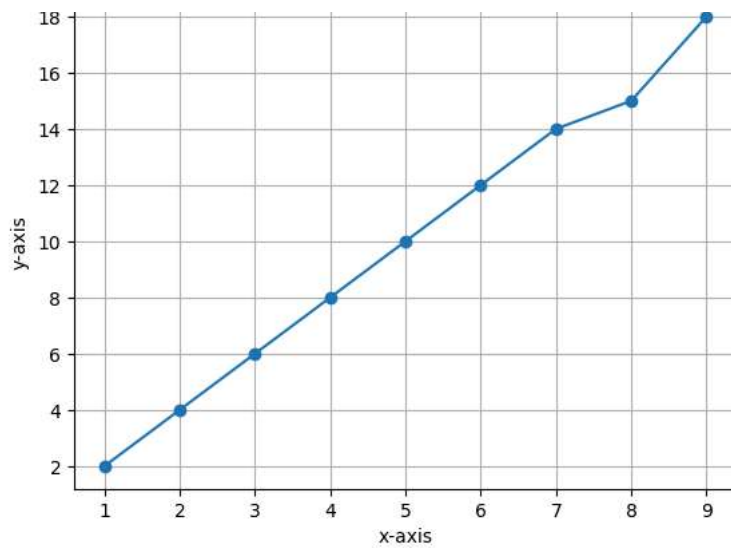
sowmya,your grade is 67
your grade is D
```

```
#visusalisation
import matplotlib.pyplot as plt

#data
x = [1,2,3,4,5,6,7,8,9]
y = [2,4,6,8,10,12,14,15,18]

#plot
plt.plot(x,y,marker='o')
plt.title("simple line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.grid(True)
plt.show()

#bar
subjects = ["math", "english", "history", "tamil", "english"]
marks = [1,2,3,4,5]
plt.bar(subjects,marks)
plt.title("subject marks")
plt.xlabel("marks")
plt.ylabel("subjects")
plt.show()
plt.pie(marks,labels=subjects)
plt.show()
```



```

import ipywidgets as widgets
from IPython.display import display

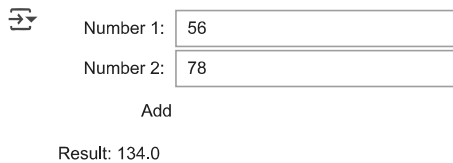
# Create input fields
num1 = widgets.FloatText(description="Number 1:")
num2 = widgets.FloatText(description="Number 2:")
result = widgets.Label(value="Result will appear here")

# Function to add numbers
def add_numbers(b):
    try:
        res = num1.value + num2.value
        result.value = f"Result: {res}"
    except Exception as e:
        result.value = f"Error: {e}"

# Button
button = widgets.Button(description="Add")
button.on_click(add_numbers)

# Display UI
display(num1, num2, button, result)

```



Number 1:

Number 2:

Add

Result: 134.0

Double-click (or enter) to edit

```

#neural network
# !pip install tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
#NEURAL NETWORK HAS 3 LAYERS Input output hidden layer
# first layer 8 neurons
# each neuron gets 2 inputs
# output layer 1 neuron
#building neural network
model = Sequential() #BUILD A MODEL LAYER BY LAYER
#DENSE -> FULLY CONNECTED LAYER
model.add(Dense(units=4,activation='relu',input_dim=2)) #STARTING WITH EMPTY LINE
model.add(Dense(units=1,activation='sigmoid'))
#compile model
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
#binary_crossentropy binary classification tasks
#optimizing optimizing the ip op
#metrics = accuracy

X = np.array([[0,0],[0,1],[1,0],[1,1]])
#xor
y = np.array([0,1,1,0])
model.fit(X,y,epochs=100,verbose=0)

model.fit(X,y,epochs=100,batch_size=1)
test_data = np.array([[0,0],[0,1],[1,0],[1,1]])
predictions = model.predict(test_data)
print((predictions > 0.5).astype(int))

```

