

```
In [5]: # 1. Task: Calculate Profit Percentage
# • Write a javascript program that takes input for the cost price and selling price of an item.
# • Hints
# ○ Prompt the user to input the cost price and selling price.
# ○ Determine whether the transaction resulted in a profit or loss.
# ○ If there is a profit calculate the profit percentage;
#   if there is a loss calculate the loss percentage.
# ○ Display the profit or loss and the respective percentage.

# pro=profit/cost_price*100
# print("Prfit :",profit)
# print("Profit Persent:",int(pro))
```

```
#-----
cost = int(input("enter your cost price"))
selling = int(input("enter your selling price"))
profit = selling - cost
loss = cost- selling
if profit>=0:
    print(f" your profit is :{profit}")
    pro=profit/cost*100
    print(f"your profit % is:{pro}")
else:
    print(f"your loss is :{loss}")
    pro=loss/cost*100
    print(f"your loss % is : {pro}")
```

```
your loss is :10
your loss % is : 10.0
```

```
In [4]: # 2. Task: Cricket Stats Analyzer
# • Objective: Write a script to analyze cricket stats for a team.
# • Hints:
# ○ Prompt the user to input the runs scored by each of the five players in a cricket match.
# ○ For each player (Player 1 to Player 5) ask the user to input the runs they scored.
# ○ Calculate the total runs scored by all players and the average runs.
# ○ Display the total runs and average runs to the user.
```

```
#-----

# p1=int(input("Enter 1st Player Name:"))
# p2=int(input("Enter 2nd Player Name:"))
# p3=int(input("Enter 3rd Player Name:"))
# p4=int(input("Enter 4th Player Name:"))
# p5=int(input("Enter 5th Player Name:"))

# total=p1+p2+p3+p4+p5
# print(f"total run of all player:{total}")
# total=p1+p2+p3+p4+p5/5
# print(f"Avrage run of all player:{total}")
```

```
total run of all player:150
Avrage run of all player:110.0
```

```
In [6]: # 3. Task: Retirement Age Calculator
# • Objective: Write a program that prompts the user for their age and tells the
# many years until they reach retirement age (65).
# • Hints:
# ○ Ask the user to input their age.
# ○ Calculate how many more years they have until they reach 65 years of
# age.
# ○ Display the number of years left until retirement or a message if the user has
# already reached retirement age.

age=int(input("Enter your age:"))
if age <65:
    x=65-age
    print(f"user have {x} years left to reach 65 ")
else:
    print("user has already reached retirement age")
```

user have 35 years left to reach 65

```
In [7]: # 4. Task: Calculate the Area of a Circle
# • Objective: Write a program to calculate the area of a circle.
# • Hints:
# ○ Ask the user to input the radius of the circle.
# ○ Calculate the area of the circle using the formula: Area =  $\pi$  *
# radius^2.
# ○ Display the calculated area.

radius=int(input("radius of circle"))
Area= (22/7)*radius**2
print(Area)
```

28.285714285714285

```
In [8]: # 5. Task: Salary Calculation
# • Objective: You have to calculate an employee's salary by computing the gross
# salary tax and net salary based on the given parameters.
# • Hints:
# ○ Base Salary = ₹50000
# ○ Bonus = ₹5000
# ○ Tax Rate = 10%
# ○ Other Charges = ₹2000
# Display the Gross Salary Tax and Net Salary.

base=50000
bonus=5000
tax_rate =10
o_c=2000

gross= base + bonus
tax=gross/100*10
net = gross-tax-o_c

print(gross)
print(net)
print(tax)
```

55000

47500.0

5500.0

```
In [7]: # 6. Task: Bank Loan Approval System
# • Objective: You have to create a javascript script that checks whether an use
```

```

# eligible for a bank loan based on various criteria.
# • Hints:
# ○ The applicant's age must be between 18 and 60 years.
# ○ The applicant's monthly income must be greater than or equal to ₹25000.
# ○ The applicant's credit score must be greater than or equal to 700.
# ○ The applicant must not have any outstanding debts greater than ₹10000
# 1. Output:
# ○ Display "Loan Approved" if the applicant meets all the conditions.
# ○ Otherwise display "Loan Rejected".

age=int(input("enter your age:"))
income=int(input("Enter your income:"))
cr=int(input("Your Credit score:"))
debts=int(input("Your any outstanding debts:"))
if age>=18 and age<=60:
    if income>=25000:
        if cr>=700:
            if debts<10000:
                print("Loan Approved")
            else:
                print("Rejected")
        else:
            print("rejected")
    else:
        print("rejected")
else:
    print("rejected")

```

Loan Approved

```

In [29]: # 7. Task: Students Interview Eligibility Checker
# • Objective:you have to design a javascript script that checks whether a stude
# eligible for an interview based on their academic score attendance percentage
# and extracurricular participation.
# • Input:
# ○ Academic Score (percentage): A floating-point number representing the
# student's academic score. Ex .78.88
# ○ Attendance Percentage: A floating-point number representing the
# student's attendance percentage. Ex.85.88
# ○ Extracurricular Participation: This indicates whether the student has
# participated in any extracurricular activities. Ex.Yes/no

# Conditions for Interview Eligibility:
# 1. The student's academic score must be 60 or above.
# 2. The student's attendance percentage must be 75 or above.
# 3. The student should have participated in at least one extracurricular activi
# Output:
# • If the student meets all three conditions print "Eligible for Interview".
# • If the student fails to meet any of the conditions print "Not Eligible for I

percentage=float(input("Your Acadmic Marks Percentage :"))
attendance=float(input("Your Acadmic Attendance Percentage :"))
par=input("participated in any extracurricular :")

if percentage>=60:

```

```

    if attendance >= 75:
        if par == "yes":
            print("Eligible for Interview")
        else:
            print("Not Eligible for Interview")
    else:
        print("Not Eligible for Interview")
else:
    print("Not Eligible for Interview")

```

Eligible for Interview

```

In [33]: # 8. Task: Validating Email Domain
# • Objective: You will implement a javascript program to validate the domain of
# user's email address. The program will check if the email contains a specific
# domain (e.g. "gmail.com").
# Problem Statement:
# You are building a registration system that only accepts email addresses from
# domain (e.g. "gmail.com"). Your task is to:

# 1. Prompt the user to enter their email address.
# 2. Check if the entered email address contains the domain "gmail.com".
# 3. Display whether the email is eligible for registration based on the domain
# check.

# 4. Print a message to inform the user if their email is eligible for registrat
# or not.
email = input("Enter your email:")
domain = "gmail.com"
if domain in email:
    print("email is eligible for registration")
else:
    print("email is not eligible for registration")

```

email is eligible for registration

```

In [35]: # Task 9: Employee Access Level System
# Objective:
# You are building a security system for a corporate office where employees are
# different access levels:
# • Admin Access
# • Staff Access
# • Visitor Access

# Scenario:
# The system must check whether two employees have been assigned exactly the sam
# access level object in the system. This is important to avoid mistakenly dupli
# access levels in memory or logic.
# Instructions:

# 1. ADMIN = "admin"
# 2. STAFF = "staff"
# 3. VISITOR = "visitor"
# • Assign access levels to two employees using these predefined constants.

# • Display:
# o "Both employees have the same access level object." if
# they point to the same object.

# o "Employees have different access level objects."

```

```
# otherwise.
ADMIN = "admin"
STAFF = "staff"
VISITOR = "visitor"

emp1=input("enter first emp level:")
emp2=input("enter second emp level:")
if emp1 == emp2:
    print("Both employees have the same access level object.")
else:
    print("Employees have different access level objects.")
```

Employees have different access level objects.

```
In [8]: # 10.Task : Student Grading System
# Create a javascript program to calculate a student's grade based on their mark
# Task:
# 1. Input: Prompt the user to enter their marks.
# 2. Criteria:
# o Grade A: 90-100
# o Grade B: 80-89
# o Grade C: 70-79
# o Grade D: 60-69
# o Grade E: 50-59
# o Grade F: 0-49
# o Invalid marks: Outside the range 0-100.
# 3. Output: Display the grade or an error message for invalid marks.
# Example Outputs:
# • Marks: 85 → Grade: B
# • Marks: 45 → Grade: F
# • Marks: 105 → Invalid marks.

marks=int(input("Enter your marks here and Check your Grade:"))

if marks>=90 and marks<=100:
    print("Your Grade is "A" ")
else:
    if marks>=80 and marks<=89:
        print("Your Grade is "B" ")
    else:
        if marks>=70 and marks<=79:
            print("Your Grade is "C" ")
        else:
            if marks>=60 and marks<=69:
                print("Your Grade is "D" ")
            else:
                if marks>=50 and marks<=59:
                    print("Your Grade is "E" ")
                else:
                    if marks>=0 and marks<=49:
                        print("Your Grade is "F" ")
                    else:
                        print("Invalid marks")

#-----Other Way -----
marks = int(input("Enter your marks here and Check your Grade: "))
```

```
# if marks >= 90 and marks <= 100:
#     print('Your Grade is "A"')
# elif marks >= 80 and marks <= 89:
#     print('Your Grade is "B"')
# elif marks >= 70 and marks <= 79:
#     print('Your Grade is "C"')
# elif marks >= 60 and marks <= 69:
#     print('Your Grade is "D"')
# elif marks >= 50 and marks <= 59:
#     print('Your Grade is "E"')
# elif marks >= 0 and marks <= 49:
#     print('Your Grade is "F"')
# else:
#     print("Invalid marks")
```

Your Grade is "E"

```
In [9]: # 11.Task : Authentication System.
# Write a javascript program that authenticates a user by checking their username
# password. The program should compare the entered credentials with predefined v
# credentials.
# • Predefined valid usernames and corresponding passwords
# username1 = "user1"
# username1_password1 = "pass@123"

# Instructions:
# 1. Input:
# o Prompt the user to input their username and password.
# 2. Processing:
# o Check if the username and password match
# 3. Output:
# o If both the username and password match the predefined valid credentials
# display "Authentication successful."
# o If either the username or the password does not match display
# "Authentication failed."

# username= str(input("Enter your user name :"))
# pasword= str(input("Enter your pasword :"))

# username1 = "user1"
# username1_password1 = "pass@123"

# if username==username1 and pasword==username1_password1:
#     print("Authentication successful.")
# else:
#     print("Authentication failed.")
```

Authentication successful.

```
In [17]: # 12.Employee Salary Based on Experience.
# You are building a system for a Human Resources (HR) department that calculate
# employee's salary based on their years of experience. The system assigns salar
# based on the number of years an employee has been working. It also offers bonu
# employees who have more than 15 years of experience.
# Scenario Breakdown:
# Context 1: Senior Employee
# • An employee with 10 or more years of experience is classified as a Senior
# Employee. The base salary for such an employee is 80000.
```

```

# • If the employee has more than 15 years of experience they receive a bonus of
# 5000 to their salary.

# Example:
# An employee with 18 years of experience:
# • They are classified as Senior Employees.
# • Their base salary is 80000.
# • Since they have more than 15 years of experience they receive an additional
# 5000 bonus.

# • The final salary is 85000.

# Context 2: Mid-Level Employee
# • Employees with 5 to 9 years of experience are classified as Mid-Level
# Employees.
# • Their base salary is 50000 with no bonus.

# Example:
# An employee with 7 years of experience:
# • They are classified as a Mid-Level Employee.
# • Their base salary is 50000.
# • Since they have fewer than 10 years of experience no bonus is added.
# • The final salary is 50000.

# Context 3: Junior Employee
# • Employees with less than 5 years of experience are classified as Junior
# Employees.
# • Their base salary is 30000 with no bonus.

# Example:
# An employee with 3 years of experience:
# • They are classified as Junior Employees.
# • Their base salary is 30000.
# • No bonus is added.
# • The final salary is 30000.

# Output Examples:
# Senior Employee with 18 years of experience:
# Enter years of experience: 18
# Senior employee
# Experience exceeds 15 years. Bonus added.

exp=int(input("Enter your exp:"))
if exp>=10 and exp<=15:
    print ("Senior Employee")
    print ("Your base salary is:80000")
elif exp>15:
    print (f"Senior Employee with {exp} years of experience:")
    print ("Your base salary is:80000 and also bonus is 5000")
    print (f"your final salary is {80000+5000}")

elif exp >=5 and exp <=9:
    print(f"Mid - Level Employee with {exp} years of experience:")
    print ("Your base salary is:50000")

elif exp <5 :
    print(f""junior Employee with {exp} years of experience:"")
    print ("Your base salary is:30000")
    print (f"your final salary is {30000}")
else:

```

```
print("fresher")
```

junior Employee with -1 years of experience:
 Your base salary is:30000
 your final salary is 30000

```
In [1]: # 13. Library Charge Calculation
# Problem Statement:
# Write a javascript program that calculates the library charge based on the number of
# book has been borrowed.
# Charge Criteria:
# • Up to 5 days: Rs. 2 per day
# • 6 to 10 days: Rs. 3 per day
# • 11 to 15 days: Rs. 4 per day
# • More than 15 days: Rs. 5 per day
# Instructions:
# 1. Input: Prompt the user to enter the number of days the book has been borrowed.
# 2. Processing: Calculate the charge based on the given criteria.
# 3. Output: Display the calculated charge.

days=int(input("Num of days :"))

if days>=0 and days<=5:
    print("your charge is Rs. 2 per day")
    print(f"your charge is {days*2}")
elif days>=6 and days<=10:
    print("your charge is Rs. 3 per day")
    print(f"your charge is {days*3}")
elif days>=11 and days<=15:
    print("your charge is Rs. 4 per day")
    print(f"your charge is {days*4}")
else:
    print("your charge is Rs. 5 per day")
    print(f"your charge is {days*5}")
```

your charge is Rs. 3 per day
 your charge is 30

```
In [17]: # 14. Arranging Three Numbers in Descending Order
# Task:
# Write a javascript program to arrange three numbers in descending order.
# Input:
# Prompt the user to enter three numbers.
# Processing:
# Sort the numbers in descending order.
# Example:
```



```

# • Enter first number: 3
# • Enter second number: 1
# • Enter third number: 2
# Output:
# • Numbers in Descending Order: 3, 2, 1

num1 = int(input("• Enter first num1:"))
num2 = int(input("• Enter second num2:"))
num3 = int(input("• Enter third num3:"))

if num1>=num2 and num1>=num3:
    if num2>=num3:
        print(num1,num2,num3)
    else:
        print(num1,num3,num2)
else:
    if num2>=num1 and num2>=num3:
        if num1>=num3:
            print(num2,num1,num3)
        else:
            print(num2,num3,num1)
    else:
        if num1>=num2:
            print(num3,num1,num2)
        else:
            print(num3,num2,num1)

```

98 7 2

In [25]: # Write a program to calculate the tax on a car purchase based on the car brand
 # 1. Mahindra: 5% tax for prices between 7L (7 Lakh) and 10L.
 # 2. Audi: 10% tax for prices between 10L and 15L.
 # 3. Jaguar: 25% tax for prices between 15L and 20L.
 # 4. Mercedes: 30% tax for prices between 20L and 25L.
 # 5. Input: The car brand and price.
 # 6. Output: The calculated tax on the purchase.

```

brand=input("brand name ").lower()
price=int(input("car price"))

if brand=="mahindra" and price >=700000 and price <=1000000:
    tax=price*5/100
    print(f"tax on purchase {tax}")
else:
    if brand=="Audi" and price >=1000000 and price <=1500000:
        tax=price*10/100
        print(f"tax on purchase {tax}")
    else:
        if brand=="Jaguar" and price >=1500000 and price <=2000000:
            tax=price*25/100
            print(f"tax on purchase {tax}")
        else:
            if brand=="Mercedes" and price>=2000000 and price<=2500000:
                tax=price*30/100
                print(f"tax on purchase {tax}")
            else :
                print("give valid input")

```

give valid input

```
In [26]: # 16.Finding the Middle Number

# o Task: Write a program to determine the middle number among three inputs.
# o Input: Prompt the user to enter three numbers.
# o Processing: Identify the middle number, which is neither the largest nor the
# smallest.
# o Output: Display the middle number.
num1 = int(input("• Enter first num1:"))
num2 = int(input("• Enter second num2:"))
num3 = int(input("• Enter third num3:"))

if (num1 >= num2 and num1 <= num3) or (num1 <= num2 and num1 >= num3):
    print("middle number:", num1)
else:
    if (num2 >= num1 and num2 <= num3) or (num2 <= num1 and num2 >= num3):
        print("middle number:", num2)
    else:
        print("middle number:", num3)
```

middle number: 2

```
In [27]: # 17.Find the greatest number.

# o Task: Write a program to find greatest number from three number
# o Input: Prompt the user to enter three numbers.
# o Output: Display the greatest number.

num1 = int(input("• Enter first num1:"))
num2 = int(input("• Enter second num2:"))
num3 = int(input("• Enter third num3:"))

if num1>num2 and num1>num3:
    print ("greatest num:",num1)
else:
    if num2>num3 and num2>num1:
        print("greatest num:",num2)
    else:
        print("greatest num:",num3)
```

greatest num: 7

```
In [ ]: # 18.Authentication System

# o Task: Write a program to authenticate a user by validating their username and
# password.
# o Predefined Credentials:
# ■ Username: user1
# ■ Password: pass@123
# o Input: Prompt the user to input their username and password.
# o Output:
# ■ If the credentials match, display "Authentication successful."
# ■ If they do not match, display "Authentication failed."

username= str(input("Enter your user name :"))
password= str(input("Enter your password :"))

username1 = "user1"
username1_password1 = "pass@123"
```

```

if username==username1 and pasword==username1_password1:
    print("Authentication successful.")
else:
    print("Authentication failed.")

```

In [30]: # 19. Calculate Class Attendance Percentage

```

# o Task: Write a program to calculate the percentage of classes attended by a
# student and determine their eligibility to sit in the exam.
# o Conditions:
# ■ Attendance percentage < 75%: Not eligible to sit in the exam.
# ■ Attendance percentage ≥ 75%: Eligible to sit in the exam.
# o Output: Display the attendance percentage and eligibility status.

```

```

attain=int(input("Enter your attendance bro:"))

```

```

if attain>=75 and attain<=100:
    print(f"your Attendance is{attain}%: Eligible to sit in the exam ")
else:
    print(f"your Attendance is{attain}%: NOT Eligible to sit in the exam ")

```

your Attendance is10000%: NOT Eligible to sit in the exam

In []: # 20. Library Charge Calculation

```

# o Task: Write a program to calculate the library charges based on the number o
# days a book has been borrowed.
# o Charge Criteria:
# ■ Up to 5 days: ₹2/day.
# ■ 6 to 10 days: ₹3/day.
# ■ 11 to 15 days: ₹4/day.
# ■ More than 15 days: ₹5/day.
# o Output: Display the total charges.

```

```

days=int(input("Num of days :"))

```

```

if days>=0 and days<=5:
    print("your charg is Rs. 2 per day")
    print(f"your charge is {days*2}")
elif days>=6 and days<=10:
    print("your charg is Rs. 3 per day")
    print(f"your charge is {days*3}")
elif days>=11 and days<=15:
    print("your charg is Rs. 4 per day")
    print(f"your charge is {days*4}")
else:
    print("your charg is Rs. 5 per day")
    print(f"your charge is {days*5}")

```

In [42]: # 21. UPSC Selection Process

```

# o Task: Simulate the UPSC selection process with the following steps:
# 1. Eligibility Check
# ■ Criteria:
# ■ Age: 21-32 years.
# ■ Graduate status: Must be a graduate.
# ■ Nationality: Must be "Indian".

```

```

# ■ Output:
# ■ If eligible, proceed to Prelims.
# ■ If ineligible, display the reason for ineligibility.

# 2. Prelims Exam
# ■ Processing: Check if the candidate's score ≥ cut-off.
# ■ Output:
# ■ If passed, .
# ■ If failed, display "You failed the Prelims."

# 3. Mains Exam
# ■ Processing: Check if the candidate's score ≥ cut-off.
# ■ Output:
# ■ If passed, proceed to Interview.
# ■ If failed, display "You failed the Mains."

# 4. Interview
# ■ Processing: Check if the candidate's score ≥ cut-off.
# ■ Output:
# ■ If passed, display "Congratulations! You have cleared the
# UPSC."
# ■ If failed, display "You failed the Interview."

# ○ Final Output: Use nested conditional statements to simulate the entire process
age=int(input("Enter your age:"))
Grad_status=input("Enter your Graduate status: YES and NO").lower()
national = input("Enter your Nationality:").lower()
if age>=21 and age<=32 and Grad_status=="yes" and national=="indian":
    print("YOU ARE ELIGIBLE FOR PRILIMS")
    cut=int(input("Enter your prilims cutoff marks"))
    cutoff= 600
    if cut>=cutoff:
        print("YOU ARE ELIGIBLE FOR MAINS")
        mains=int(input("Enter your Mains exam marks :"))
        main_cutoff=800
        if mains>=main_cutoff:
            print("you are eligible for proceed to Interview")
            interview_marks = int(input("enter your interview marks:"))
            interview_cut_marks=500
            if interview_marks>=interview_cut_marks:
                print("Congratulations! You have cleared the UPSC")
            else:
                print("you are not eligible")
        else:
            print("you are not eligible")
    else:
        print("you are not eligible")
else:
    if age<21 and age>32:
        print("you are not eligible Because of age ")
    else:
        if Grad_status=="no":
            print("you are not eligible Because you are not graduated")
        else:
            print("your are not indian")

```

your are not indian

```
In [ ]: 22. Menu-Driven Login System
1. Create the Menu:
  o Display a menu with three choices for the user:
  ■ Login with Phone
  ■ Login with Email

  ■ Exit the system
2. Predefined Credentials:
  o Phone number: "1234567890"
  o OTP: "1234"
  o Email: "user@example.com"
  o Password: "password123"
3. Login Functionality:
  o Option 1 (Login with Phone):
  ■ Prompt the user to enter their phone number and OTP.
  ■ Compare the input with a predefined phone number and OTP.
  ■ Display success if both match or an error message if they don't.
  o Option 2 (Login with Email):
  ■ Prompt the user to enter their email and password.
  ■ Compare the input with predefined email and password.
  ■ Display success if both match or an error message if they don't.
  o Option 3 (Exit):
  ■ Display an exit message and terminate the program.
  o Invalid Input:
  ■ Handle invalid user choices and ask the user to select a valid option.
Output:
1. If the user enters a valid phone number and OTP, display: "Login successful with phone!"
2. If the user enters valid email and password, display: "Login successful with email!"
3. If the user selects the exit option, display: "Exiting the program. Have a nice day!"
4. If the user enters invalid credentials or an invalid choice, display appropriate messages.
```

```
In [ ]: 23. Create Your Own KBC Game
Design and implement a quiz game inspired by the popular Kaun Banega Crorepati (game show). The aim of this assignment is to test the user's knowledge through a multiple-choice questions, track their score, and display statistics at the end game also provides the flexibility to skip any question.
Instructions:

1. Game Structure:
  o The game will consist of 5 multiple-choice questions.
  o The user will be asked a question with 4 options (A, B, C, D).
  o The user can choose to skip any question they do not want to answer.
2. Scoring System:
  o Points will be awarded for correct answers as follows:
  ■ Question 1 → 1000 points
  ■ Question 2 → 2000 points
  ■ Question 3 → 3000 points
  ■ Question 4 → 5000 points
  ■ Question 5 → 10000 points
  o For incorrect answers, no points will be awarded.
  o For skipped questions, no points will be awarded, but the game will continue.
3. End of Game Statistics:
  o At the end of the game, the following statistics will be displayed:
  ■ Total score accumulated from correct answers.
```

- Number of correct answers provided by the user.
 - Number of skipped questions.
 - Number of wrong answers
4. User Experience:
- At the beginning of the game, ask the user whether they would like to start **or not** the game.
 - Provide the option **for** the user to skip any question at any point..
5. Game Ending:
- The game will end when all the questions have been answered **or** skipped. The user should receive their total score **and** a summary of their performance.