

CLASS XI- PYTHON PROGRAMMING

PRACTICAL JOURNAL: 2023-2024

STUDENT NAME : EESHA PRADEEP

CLASS & SEC : 11 A

ROLL NO. : S9422

INDEX

SNO	TOPIC	PAGE NO.
1	Simple Programs	3
2	If, elif, else	10
3	Loops	23
4	String Manipulation	34
5	List Manipulation	40
6	Tuple	52
7	Dictionary	57

SIMPLE PROGRAMS

1. WAP to assign your name and class to the variables *name* and *grade* respectively. Display the result in the given format

Name :<your name>

Class :<your class>

(make use of “\t”)

1 - name and class.py ×

```
1 name = input("Enter your name: ")
2 grade = input("Enter your class: ")
3 print("Name:",name," \t Class:",grade)
```

Shell ×

```
Enter your name: Eesha
Enter your class: 11A
Name: Eesha      Class: 11A
```

2. WAP to assign the value 10 to the variable price, double the value of the variable price , decrease the value by 5 and print it.

1 - name and class.py × 2 - 10x2-5.py ×

```
1 price = 10
2 print(price*2-5)
```

Shell ×

```
>>> %Run '2 - 10x2-5.py'
15
```

3. WAP to accept a username and print “Welcome” + *username*

1 - name and class.py × 2 - 10x2-5.py × 3 - username.py ×

```
1 name = input("Enter your username: ")
2 print("Welcome "+name)
```

Shell ×

```
>>> %Run '3 - username.py'
Enter your username: Eesha_Pradeep
Welcome Eesha_Pradeep
```

4. WAP to accept a number and print its square and cube.

1 - name and class.py x 2 - 10x2-5.py x 3 - username.py x 4 - square and cube.py x

```
1 n = int(input("Enter a number: "))
2 print(n**2)
3 print(n**3)
```

Shell x

```
>>> %Run '4 - square and cube.py'
```

Enter a number: 2

4

8

5. WAP to accept the radius of a circle and display the diameter, area and perimeter. [$2r$, πr^2 and $2\pi r$]

5 - circle.py x

```
1 n = int(input("Enter the radius: "))
2 print("diameter = ", n*2)
3 print("area = ", 3.14*n**2)
4 print("perimeter = ", n*2*3.14)
```

Shell x

```
>>> %Run '5 - circle.py'
```

Enter the radius: 49

diameter = 98

area = 7539.14

perimeter = 307.72

6. WAP to find the area of a triangle. [$\frac{1}{2} (l*b)$]

6 - area of triangle.py x 7 - swapping variables.py x 8 - total and average.py x 9

```
1 h = int(input("Enter the height: "))
2 b = int(input("Enter the base: "))
3 print("area = ", h*b/2)
```

Shell x

```
>>> %Run '6 - area of triangle.py'
```

Enter the height: 5

Enter the base: 7

area = 17.5

7. WAP to swap the values of two numbers.

```
7 - swapping variables.py 8 - total and average.py 9 - time input minutes.py 10
1 n1 = input("Enter the first number: ")
2 n2 = input("Enter the second number: ")
3 print("Actual order: ",n1,n2)
4 n1, n2 = n2, n1
5 print("Reverse order: ",n1,n2)
```

Shell

```
>>> %Run '7 - swapping variables.py'
Enter the first number: 4
Enter the second number: 8
Actual order: 4 8
Reverse order: 8 4
```

8. Input 3 numbers. Calculate total and Average.

```
8 - total and average.py 9 - time input minutes.py 10 - inches to cm.py 11 - sphere.py
1 n1 = int(input("Enter the first number: "))
2 n2 = int(input("Enter the second number: "))
3 n3 = int(input("Enter the third number: "))
4 total = n1+n2+n3
5 avg = total/3
6 print("total = ",total)
7 print("Average = ",avg)
```

Shell

```
>>> %Run '8 - total and average.py'
Enter the first number: 5
Enter the second number: 10
Enter the third number: 20
total = 35
Average = 11.666666666666666
```

9. WAP to convert the given time in minutes into hours and minutes.(E.g. : 90

min = 1 Hr 30 min)

```
9 - time input minutes.py 10 - inches to cm.py 11 - sphere.py 12 - gallon to cubic feet.py 14 - money input pair
1 imin = int(input("Enter the number of minutes: "))
2 h = imin//60
3 m = imin%60
4 print("total number of minutes = ", imin)
5 print(imin, "minutes = ", h, "hours and ", m, "minutes")
```

Shell

```
>>> %Run '9 - time input minutes.py'
Enter the number of minutes: 500
total number of minutes = 500
500 minutes = 8 hours and 20 minutes
```

10. WAP to accept the length in inches and convert it to centimeters. (one inch = 2.54 cm)

```
10 - inches to cm.py ✘ 11 - sphere.py ✘ 12 - gallon to cubic feet.py ✘ 14 - money input paise.py ✘ 15 - lengt
1 inches = int(input("Enter the length in inches: "))
2 cm = inches*2.54
3 print(inches, "inches= ",cm, " cm")
```

Shell ✘

```
>>> %Run '10 - inches to cm.py'
Enter the length in inches: 45
45 inches= 114.3 cm
```

11. WAP to input the radius and calculate area and volume of a sphere.

```
11 - sphere.py ✘ 12 - gallon to cubic feet.py ✘ 14 - money input paise.py ✘ 15 -
1 n = int(input("Enter the radius: "))
2 print("area = ", 4*3.14*n**2)
3 print("volume = ", n**3*3.14*4/3)
```

Shell ✘

```
>>> %Run '11 - sphere.py'
Enter the radius: 7
area = 615.44
volume = 1436.0266666666666
```

12. Assume 1 cubic foot = 7.481 gallons. WAP to ask the user to enter the number of gallons and display it in cubic feet.

```
12 - gallon to cubic feet.py ✘ 14 - money input paise.py ✘ 15 - length input metres.py ✘
1 gallons = float(input("Enter the volume in gallons: "))
2 cubicfeet = gallons/7.481
3 print(gallons, "gallons= ",cubicfeet, " cubic feet")
```

Shell ✘

```
>>> %Run '12 - gallon to cubic feet.py'
Enter the volume in gallons: 110
110.0 gallons= 14.703916588691351 cubic feet
```

- 13. Input basic salary. Calculate allowance as 12% of basic salary and deduction as 10% of basic salary. Calculate net salary by using the formula net salary=basic +allowance – deduction**

```
13 - salary.py x
1 bs = int(input("Enter the basic salary: "))
2 a = bs*12/100
3 d = bs/10
4 s = bs+a-d
5 print("Salary =",s)
```

```
Shell x
>>> %Run -c $EDITOR_CONTENT
Enter the basic salary: 30000
Salary = 30600.0
```

- 14. Input amount in paise. Convert to equivalent Rupees and paise.**

```
14 - money input paise.py x 15 - length input metres.py x
1 ip = int(input("Enter the amount in paise: "))
2 r = ip//100
3 p = ip%100
4 print("total amount in paise = ", ip)
```

```
Shell x
>>> %Run '14 - money input paise.py'
Enter the amount in paise: 540
total amount in paise =  540
540 paise = 5 rupees and 40 paise
```

- 15. Input value in metres. Convert to equivalent metres and kilometres.**

```
15 - length input metres.py x
1 im = int(input("Enter the amount in metres: "))
2 k = im//1000
3 m = im%1000
4 print("total distance in metres = ", im)
5 print(im, "m = ", k, "km and ", m, "m")
```

```
Shell x
>>> %Run '15 - length input metres.py'
Enter the amount in metres: 7869
total distance in metres =  7869
7869 m = 7 km and 869 m
```

16. Input value in seconds. Calculate equivalent hours , minutes and seconds

```
15 - length input metres.py 16 - time input seconds.py 17 - sum of digits of number.py 19 - switching three numbers.py 18 - celcius to farenheit.py
1 isec = int(input("Enter the number of seconds: "))
2 h = isec//3600
3 rs = isec%3600
4 m = rs//60
5 rs2 = rs%60
6 print("total number of seconds = ", isec)
7 print(isec, "seconds = ", h, "hours and ", m, "minutes and ", rs2,"seconds")
```

Shell >

```
>>> %Run '16 - time input seconds.py'
Enter the number of seconds: 697800
total number of seconds = 697800
697800 seconds = 193 hours and 50 minutes and 0 seconds
```

17. WAP to accept a 2 digit number and display the sum of its digits. (e.g. for 57 the display should be 12)

```
17 - sum of digits of number.py 19 - switching three numbers.py 18 - celcius to farenheit.py
1 n = int(input("Enter a two digit number: "))
2 n1 = n//10
3 n2 = n%10
4 s = n1+n2
5 print("Sum =", s)
```

Shell >

```
>>> %Run '17 - sum of digits of number.py'
Enter a two digit number: 34
Sum = 7
```

18. WAP to accept temperature in celsius and convert it to fahrenheit

```
19 - switching three numbers.py 18 - celcius to farenheit.py
1 f = int(input("Enter the temperature in celcius: "))
2 c = (f-32)*5/9
3 print("Temperature in farenheit = ", c)
```

Shell >

```
>>> %Run '18 - celcius to farenheit.py'
Enter the temperature in celcius: 56
Temperature in farenheit = 13.333333333333334
```

19. WAP to input 3 numbers and swap them as 1st number becomes 2nd, 2nd the 3rd and 3rd as the 1st.

```
19 - switching three numbers.py x
1 n1 = int(input("Enter the first number: "))
2 n2 = int(input("Enter the second number: "))
3 n3 = int(input("Enter the third number: "))
4 n1, n2, n3 = n2, n3, n1
5 print("The first number after reversing =", n1)
6 print("The second number after reversing =", n2)
7 print("The third number after reversing =", n3)

Shell x
>>> %Run '19 - switching three numbers.py'
Enter the first number: 1
Enter the second number: 2
Enter the third number: 3
The first number after reversing = 2
The second number after reversing = 3
The third number after reversing = 1
```

IF, ELIF, ELSE

Class work:

1. Write a program to check if a number entered is positive or negative.

```
(cw) 1 (cw) (cw) 3 - (cw) 4 - (HOTS) (HOT (HOTS (HOTS) (HOTS (hw) 1 (hw
1 n = int(input("Enter a number: "))
2 if n!=0:
3     if n>0:
4         print("Positive")
5     else:
6         print("Negative")
7
```

```
Shell ×
Enter a number: 5
Positive
```

2. Write a program to check if the number entered is even or odd.

```
(cw) (cw) 3 - (cw) 4 - (HOTS) (HOT (HOTS (HOTS) (HOTS (hw) 1 (hw 3
1 n = int(input("Enter a number: "))
2 if n%2==0:
3     print("Even")
4 else:
5     print("Odd")
```

```
Shell ×
>>> %Run '(cw) 2 - even or odd.py'
Enter a number: 52
Even
```

3. Write a program to print the square if the number entered is even, otherwise print its cube.

```
(cw) 3 - (cw) 4 - (HOTS) (HOT (HOTS (HOTS) (HOTS) (hw) 1 (hw) 3 (hw
1 n = int(input("Enter a number: "))
2 if n%2==0:
3     print(n**2)
4 else:
5     print(n**3)
```

```
Shell ×
>>> %Run '(cw) 3 - even square and odd cube.py'
Enter a number: 1009
1027243729
```

4. Write a program to accept a number and display if it is negative, positive or a zero.

```
1 n = int(input("Enter a number: "))
2 if n!=0:
3     if n>0:
4         print("Positive")
5     else:
6         print("Negative")
7 else:
8     print("zero")
```

Shell ×
=> %Run '(cw) 4 - positive, negative o
Enter a number: 0
zero

Home Work:

- 1. Accept a day(Sunday, Monday..)If the entered day is Friday or Saturday
display “Holiday” otherwise “Working day”**

```
(HOTS) 1 (HOTS (HOTS) (HOTS) 4 (HOTS) (hw) 1 - (hw) 3 - (hw) 2 (pr
1 d = input("Enter a day: ")
2 if d=="Friday" or d=="Saturday":
3     print("Holiday")
4 else:
5     print("Working day")
```

Shell ×
=> %Run '(hw) 1 - working day or holi
Enter a day: Sunday
Working day

- 2. WAP accepts 3 integers representing date (14, 6, 19) and prints the date in the
format 14th June 2019.**

```
(practical) 8 - date month year.py (practical) 9 - quadratic roots.py (practical) 10 - goemetrical area menu.py (hw) 2 - date month
1 d = int(input("Enter the date: "))
2 m = int(input("Enter the month: "))
3 y = int(input("Enter the year: "))
4 year = 2000+y
5 if d%10==1 and d!=11:
6     day=str(d)+"st"
7 elif d%10==2 and d!=12:
8     day=str(d)+"nd"
9 elif d%10==3:
10    day=str(d)+"rd"
11 else:
12    day=str(d)+"th"
13 month_list = [ "", "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December" ]
14 for i in month_list:
15     if m==month_list.index(i):
16         month=i
17 print(day,month,year)
18
```

Shell ×
=> %Run '(hw) 2 - date month year.py'
Enter the date: 23
Enter the month: 3
Enter the year: 23
23rd March 2023

3. WAP to print whether a given input is an uppercase, or a lowercase character or a digit or any other character.

```
(hw) 3 - (practi (practical) (practice (practical (practical) (practical) 7 (p
1 n = input("Enter any character: ")
2 if n>="a" and n<="z":
3     print("It is a lowercase letter")
4 elif n>="A" and n<="Z":
5     print("It is a uppercase letter")
6 else:
7     print("It is a number or special character")
```

Shell ×

```
>>> %Run '(hw) 3 - upper and lowercase.py'
Enter any character: D
It is a uppercase letter
```

Practical:

1. Write a program to accept a 4 digit number and display if it is a leap year.

```
(practic (practical) 2 (practical (practical) (practical) 6 (practical) 7
1 year = int(input("Enter the year: "))
2 if year%100==0:
3     if year%400==0:
4         print(year,"is a leap year")
5     else:
6         print(year,"is not a leap year")
7 else:
8     if year%4==0:
9         print(year,"is a leap year")
10    else:
11        print(year,"is not a leap year")
```

Shell ×

```
*** runfile('C:\Users\practical\Documents\leap year.py'
Enter the year: 1800
1800 is not a leap year
```

2. Write a program to accept the 2 sides of a rectangle and display if its area is greater than its perimeter.

```
(practical) 2 - (practical) (practical) 5 (practical) 6 - (practical) 7 - | (practical) (prac
```

```
1 l = int(input("Enter the length of the rectangle: "))
2 b = int(input("Enter the breadth of the rectangle: "))
3 a = l*b
4 p = 2*(l+b)
5 if a>b:
6     print("Area is greater than perimeter.")
7 else:
8     print("Area is not greater than perimeter.")
```

Shell ×

```
>>> %Run '(practical) 2 - area or perimeter greater.py'
Enter the length of the rectangle: 20
Enter the breadth of the rectangle: 18
Area is greater than perimeter.
```

3. Write a program to print the largest of three numbers.

```
(practical) 3 (practical) 5 - (practical) 6 - d (practical) 7 - m (practical) 8 (practica
```

```
1 n1 = int(input("Enter the first number: "))
2 n2 = int(input("Enter the second number(different): "))
3 n3 = int(input("Enter the third number(different): "))
4 if n1>n2:
5     if n1>n3:
6         print(n1,"is the greatest")
7     else:
8         print(n3,"is the greatest")
9 else:
10    if n2>n3:
11        print(n2,"is the greatest")
12    else:
13        print(n3,"is the greatest")
```

Shell ×

```
>>> %run '(practical) 3 - greatest number.py'
Enter the first number: 87
Enter the second number(different): 92
Enter the third number(different): -3
92 is the greatest
```

4. Write a program to accept the sales amount and display the net amount according to the following discount table.

Sales > 50000	40% discount
Sales between 40001 to 50000	30% discount
Sales between 30001 to 40000	20% discount
Sales between 20001 to 30000	15% discount
Sales <= 20000	05% discount

$$\text{Net} = \text{Sales} - \text{Discount}$$

```
(practical) 5 - (practical) 6 - d (practical) 7 - m@ (practical) 8 - (practical)
```

```

1 s = int(input("Enter the sales amount: "))
2 if s>50000:
3     d = 40
4 elif s>40000:
5     d = 30
6 elif s>30000:
7     d = 20
8 elif s>20000:
9     d = 15
10 else:
11     d = 5
12 print("Discount = ",d,"%")
13 print("Net Amount = ",s-(d*s/100))

```

Shell ×

```
>>> %Run '(practical) 4 - net sale amount.py'
Enter the sales amount: 4000
Discount = 5 %
Net Amount = 3800.0
```

5. Write a program to accept two numbers and an operator and display the answer according to the specified operator (+, -, *, /, %).

```
(practical) 5 - in (practical) 6 - day (practical) 7 - marki (practical) 8 - (practical)
```

```

1 n1 = int(input("Enter the first number: "))
2 n2 = int(input("Enter the second number: "))
3 o = input("Enter the operator to be used: ")
4 if o=="+":
5     print("Sum =", n1+n2)
6 elif o=="-":
7     print("Difference =",n1-n2)
8 elif o=="*":
9     print("Product =",n1*n2)
10 elif o=="/":
11     print("Quotient =",n1/n2)
12 elif o=="%":
13     print("Remainder =",n1%n2)
14 else:
15     print("Invalid operator input")

```

Shell ×

```
>>> %Run '(practical) 5 - input user operation.py'
Enter the first number: 2
Enter the second number: 9
Enter the operator to be used: %
Remainder = 2
```

6. Write a program to accept the day number from the user and display the entrance fee for a theme park. Accept the number of tickets and display the total amount to be paid.

Saturday	Dhs 5
Sunday	Dhs 4
Monday	Dhs 3
Tuesday	Dhs 2
Wednesday	Dhs 2
Thursday	Dhs 5
Friday	Dhs 4

```
(practical) 6 - day num: (practical) 7 - marking scl (practical) 8 - date (practical) 9 - qua (practical) 10 - go
1 d = int(input("Enter the day number (saturday is 1 and friday is 7): "))
2 n = int(input("Enter the number of tickets needed: "))
3 if d==1 or d==6:
4     print("The entry fee is",n*5," Dirhams")
5 elif d==2 or d==7:
6     print("The entry fee is",n*4," Dirhams")
7 elif d==3:
8     print("The entry fee is",n*3," Dirhams")
9 elif d==4 or d==5:
10    print("The entry fee is",n*2," Dirhams")
11 else:
12     print("Invalid entry")
```

Shell ×

```
>>> %Run '(practical) 6 - day number and ticket fee.py'
Enter the day number (saturday is 1 and friday is 7): 5
Enter the number of tickets needed: 3
The entry fee is 6 Dirhams
```

7. Write a program to accept the Physics, Chemistry and Mathematics marks of a student and display the grade accordingly.

Average	Grade
> 90	A
81 to 90	B
71 to 80	C
<=70	D

(practical) 7 - marking scheme a (practical) 8 - date mon (practical) 9 - quadratic (practical)

```

1 p = int(input("Enter the mark for physics out of 100: "))
2 c = int(input("Enter the mark for chemistry out of 100: "))
3 m = int(input("Enter the mark for mathematics out of 100: "))
4 a = (p+c+m)/3
5 print("Average=",a)
6 if a>90:
7     print("Grade - A")
8 elif a>81:
9     print("Grade - B")
10 elif a>71:
11     print("Grade - C")
12 else:
13     print("Grade - D")

```

Shell x

```

>>> %Run '(practical) 7 - marking scheme and grades.py'
Enter the mark for physics out of 100: 79
Enter the mark for chemistry out of 100: 56
Enter the mark for mathematics out of 100: 98
Average= 77.66666666666667
Grade - C

```

8. WAP that accepts 3 integers representing date (14, 6, 19) and prints the date in the format 14th June 2019.

```
(practical) 8 - date month year.py × (practical) 9 - quadratic roots.py × (practical) 10 - goemetrical area menu.py ×
1 d = int(input("Enter the date: "))
2 m = int(input("Enter the month: "))
3 y = int(input("Enter the year: "))
4 year = 2000+y
5 if d%10==1 and d!=11:
6     day=str(d)+"st"
7 elif d%10==2 and d!=12:
8     day=str(d)+"nd"
9 elif d%10==3:
10    day=str(d)+"rd"
11 else:
12    day=str(d)+"th"
13 month_list = [ "", "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December" ]
14 for i in month_list:
15     if m==month_list.index(i):
16         month=i
17 print(day,month,year)

Shell ×
>>> %Run '(practical) 8 - date month year.py'
Enter the date: 11
Enter the month: 5
Enter the year: 21
11th May 2021
```

9. Write a program to calculate and print roots of a quadratic equation $ax^2+bx+c=0$. (hint :the roots depend on value d which is equal to b^2-4ac , if $d>0$, the roots are $r_1=-b+\sqrt{d}/2a$ and $r_2=-b-\sqrt{d}/2a$, if $d=0$ then the $r_1=r_2=-b/2a$ otherwise if $d<0$ then imaginary roots)

```
(practical) 9 - quadratic roots.py × (practical) 10 - goemetrical area menu.py ×
1 print("The format of a quadratic equation is ax^2 + bx + c = 0")
2 a = int(input("Enter the value of a: "))
3 b = int(input("Enter the value of b: "))
4 c = int(input("Enter the value of c: "))
5 d = b**2-(4*a*c)
6 if a==0:
7     print("Not a quadratic equation.")
8 elif d<0:
9     print("The roots are not real")
10 elif d==0:
11     print("The roots are real and equal")
12     print("The root is", -b/(2*a))
13 else:
14     print("The roots are real and distinct")
15     print("The roots are", (-b+d**0.5)/(2*a), "and", (-b-d**0.5)/(2*a))

Shell ×
>>> %Run '(practical) 9 - quadratic roots.py'
The format of a quadratic equation is ax^2 + bx + c = 0
Enter the value of a: 4
Enter the value of b: -12
Enter the value of c: 8
The roots are real and distinct
The roots are 2.0 and 1.0
```

10.WAP to create a menu as shown below and perform the various operations based on users response

MAIN MENU

Area of a Triangle
Area of a Square
Area of a Circle
Exit

```
(practical) 10 - goemetrical area menu.py x
1 print(''  MAIN MENU
2 -----
3 1. Area of a triangle
4 2. Area of a square
5 3. Area of a circle
6 4. Exit''')
7 menu = int(input("Enter the shape whose area you want to find from the menu (options - 1,2,3 or 4): "))
8 if menu==1:
9     tb = int(input("Enter the base of the triangle: "))
10    th = int(input("Enter the height of the triangle: "))
11    print("Area of the triangle =", tb*th/2)
12 elif menu==2:
13     ss = int(input("Enter the side of the square: "))
14     print("Area of the square =", ss**2)
15 elif menu==3:
16     r = int(input("Enter the radius of the circle: "))
17     print("Area of the circle =", 3.14*r**2)
18 elif menu==4:
19     print("No area for you :<< ")
20 else:
21     print("Invalid entry")
<
Shell x
MAIN MENU
-----
1. Area of a triangle
2. Area of a square
3. Area of a circle
4. Exit
Enter the shape whose area you want to find from the menu (options - 1,2,3 or 4): 1
Enter the base of the triangle: 4
Enter the height of the triangle: 3
Area of the triangle = 6.0
```

HOTS Challenging Questions

1. WAP to print whether a given input is an uppercase, or a lowercase character or a digit or any other character.

(HOTS) 1 - character identification (HOTS) 3 - non duplicate sum.py

```
1 n = input("Enter a character: ")
2 if n>="A" and n<="Z":
3     print("Uppercase")
4 elif n>="a" and n<="z":
5     print("Lowercase")
6 elif n>="0" and n<="9":
7     print("Digit")
8 else:
9     print("Special Character")
```

Shell ×

```
>>> %Run '(HOTS) 1 - character identification'
Enter a character: j
Lowercase
```

2. Write a program to get date in three variables dd, mm, yy to accept the date, month and the year in the century format. Write code to check the validity of the date.

(HOTS) 3 - non duplicate sum.py × (HOTS) 5 - India or UAE holiday.py × (HOTS) 2 - validity of date.py × (HOTS) 4 - largest and second largest.py ×

```
1 dd = int(input("Enter the date in digits: "))
2 mm = input("Enter the month in its full form in lowercase: ")
3 yy = int(input("Enter the year in its full form in digits: "))
4 leap = True
5 if yy%100==0:
6     if yy%400!=0:
7         leap = False
8 else:
9     if yy%4!=0:
10        leap = False
11 if mm not in ["january", "february", "march", "april", "may", "june", "july", "august", "september", "october", "november", "december"]:
12     print("Invalid date (month input incorrect)")
13 elif dd<1 or dd>31:
14     print("Invalid date (date input incorrect)")
15 elif leap==False and mm=="february" and dd==29:
16     print("Invalid date")
17 elif mm=="february" and dd>29:
18     print("Invalid date (date/month input incorrect)")
19 elif mm in ["april", "june", "september", "november"] and dd>30:
20     print("Invalid date (month/date input incorrect)")
21 else:
22     print("The date is valid! :) ")
```

Shell ×

```
>>> %Run '(HOTS) 2 - validity of date.py'
Enter the date in digits: 29
Enter the month in its full form in lowercase: february
Enter the year in its full form in digits: 2019
Invalid date
```

3. Accept three numbers. Find the sum and the sum of non duplicate numbers(Eg: sum of 3, 3, 6=12;

Sum of Non duplicate numbers of 3, 3, 6=6)

(HOTS) 3 - non duplicate sum.py × (HOTS) 5 - India or l

```
1 n1 = int(input("Enter the first number: "))
2 n2 = int(input("Enter the second number: "))
3 n3 = int(input("Enter the third number: "))
4 s = n1+n2+n3
5 if n1!=n2 and n1!=n3:
6     if n2!=n3:
7         ts = n1+n2+n3
8     else:
9         ts = n1+n2
10 elif n1!=n3:
11     ts = n1+n3
12 else:
13     ts = n1+n2
14 print("sum = ",s)
```

Shell ×

```
>>> %Run '(HOTS) 3 - non duplicate sum.py'
Enter the first number: 3
Enter the second number: 4
Enter the third number: 3
sum = 10
sum of non duplicate numbers = 7
```

4. Write the code to find the largest and second largest from accepted three numbers.

(HOTS) 5 - India or UAE holiday.py × (HOTS) 4 - largest and second largest.py ×

```
1 n = int(input("Enter the number of numbers you want to check: "))
2 a = int(input("Enter the first number: "))
3 b = int(input("Enter another number: "))
4 if a>b:
5     l = a
6     s = b
7 else:
8     l = b
9     s = a
10 for i in range(n-2):
11     c = int(input("Enter another number: "))
```

```

12     if c>l:
13         s = l
14         l = c
15     elif c>s:
16         s = c
17 print("largest =" .l)

```

Shell ×

```

>>> %Run '(HOTS) 4 - largest and second largest.py'
Enter the number of numbers you want to check: 3
Enter the first number: 3
Enter another number: -2
Enter another number: 7
largest = 7
second largest = 3

```

5. Accept the country name(India or UAE).Accept a day also. Display Holiday Sunday and Saturday for India Friday and Saturday for UAE

(HOTS) 5 - India or UAE holiday.py ×

```

1 c = input("Enter India or UAE: ")
2 d = input("Enter a day: ")
3 if c == "India":
4     if d=="Sunday" or d=="Saturday":
5         print("Holiday")
6     else:
7         print("Working day")
8 elif c == "UAE":
9     if d=="Friday" or d=="Saturday":
10        print("Holiday")
11    else:
12        print("Working day")
13 else:
14     print("Invalid Country input")
15

```

Shell ×

```

>>> %Run '(HOTS) 5 - India or UAE holiday.py
Enter India or UAE: India
Enter a day: Sunday
Holiday

```

EXTRA QUESTIONS:

1. To find the HCF of two given nos.

```

1 n1 = int(input("Enter the first no.: "))
2 n2 = int(input("Enter the second no.(different): "))
3 if n1>n2:
4     smaller = n2
5 else:
6     smaller = n1
7 for i in range(1, smaller+1):
8     if ((n1%i==0) and (n2%i==0)):
9         hcf = i

```

Shell ×

```

Enter the first no.: 54
Enter the second no.(different): 24
The HCF is 6

```

2. To find the LCM of two given nos.

```

1 n1 = int(input("Enter the first no.: "))
2 n2 = int(input("Enter the second no.(different): "))
3 if n1>n2:
4     greater=n1
5 else:
6     greater=n2
7 while True:
8     if (greater%n1==0) and (greater%n2==0):
9         lcm = greater
10        break
11    greater+=1
12 print("LCM =",lcm)

```

Shell ×

```

>>> %Run -c $EDITOR_CONTENT
Enter the first no.: 54
Enter the second no.(different): 24
216

```

3. To find the smallest and second smallest from the n nos inputted.

```

1 x = int(input("Enter the number of numbers: "))
2 n = int(input("Enter a number: "))
3 nn = int(input("Enter another number: "))
4 if n>nn:
5     l = nn
6     s1 = n
7 elif n==nn:
8     l = nn
9     s1 = 1
10 else:
11     l = n
12     s1 = nn
13 for i in range(x-2):
14     n = int(input("Enter another number: "))
15     if n<l:
16         sl = 1
17         l = n
18     elif n<s1:
19         sl = n
20 print("Lowest =",l)
21 print("Second Lowest =",sl)

```

Shell ×

```

>>> %Run 'lowest and second lowest.py'
Enter the number of numbers: 10
Enter a number: 1
Enter another number: 0
Enter another number: 123
Enter another number: -1
Enter another number: -4
Enter another number: -65
Enter another number: 265
Enter another number: -90
Enter another number: -12
Enter another number: 34
Lowest = -90
Second Lowest = -65

```

LOOPS

FOR LOOP:

1. Write a program to print “Computer Science” 10 times.

1 - computer s 2 - first ten natura 3 - odd

```
1 for i in range(10):  
2     print("Computer Science")
```

Shell ×

```
Computer Science  
Computer Science
```

2. WAP to print first N natural numbers.

2 - first ten natural n 3 - odd

```
1 for i in range(1,11):  
2     print(i)
```

Shell ×

```
>>> %Run '2 - first ten na
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

3. WAP to print all odd and even numbers between 1 and n. Accept n from user

3 - odd even 5 - sum of first n num 6 - sum o

```
1 n = int(input("Enter a number: "))
2 for i in range(1,n,2):
3     print(i,"- odd")
4     print(i+1,"- even")
```

Shell ×

```
>>> %Run '3 - odd even.py'
```

```
Enter a number: 4
1 - odd
2 - even
3 - odd
4 - even
```

4. WAP to print the even numbers in reverse between 100 and n.

```
1 n = int(input("Enter a number: "))
2 for i in range(100,n,-2):
3     print(i)
```

Shell ×

```
>>> %Run '4 - reverse even numbers between 100 and n.py'
```

```
Enter a number: 85
100
98
96
94
92
90
88
86
```

5. WAP to print the sum of 1st N natural numbers. Accept N from user

5 - sum of first n numbe 6 - sum of 10 numb

```
1 n = int(input("Enter a number: "))
2 s = 0
3 for i in range(1,n+1):
4     s+=i
5 print(s)
```

Shell ×

```
>>> %Run '5 - sum of first n numbers.py'
```

```
Enter a number: 9
45
```

6. WAP to accept any 10 numbers and display their sum.

6 - sum of 10 numbers.py 7 - 1 to n step value 3.

```
1 s = 0
2 for i in range(10):
3     n = int(input("Enter a number: "))
4     s+=n
5 print(s)
```

Shell ×

```
>>> %Run '6 - sum of 10 numbers.py'
```

```
Enter a number: 5
Enter a number: 4
Enter a number: 3
Enter a number: 2
Enter a number: 1
Enter a number: 6
Enter a number: 7
Enter a number: 8
Enter a number: 9
Enter a number: 10
55
```

7. WAP to display the series 1 4 7 10 13 16 n.

7 - 1 to n step value 3.py × 8 - fibonacci.py ×

```
1 n = int(input("Enter a number: "))
2 for i in range(1,n+1,3):
3     print(i)
```

Shell ×

```
>>> %Run '7 - 1 to n step value 3.py'
```

```
Enter a number: 15
1
4
7
10
13
```

8. WAP to print the Fibonacci series accepting the number of terms from the user. (first two terms of the series should be 0 and 1)

```
1 n = int(input("Enter the number of terms: "))
2 f = 0
3 l = 1
4 print(f, l, sep=", ", end=", ")
5 for i in range(n-2):
6     m = f
7     f = l
8     l+=m
```

Shell ×

```
>>> %Run '8 - fibonacci.py'
```

```
Enter the number of terms: 7
0, 1, 1, 2, 3, 5, 8,
```

9. WAP to print the series:

$$1 + X^2 + X^4 + \dots \dots \dots X^n$$

9 - sum of powers of even.py 10 - menu driven program.py 11

```
1 s = 1
2 x = int(input("Enter the base number: "))
3 n = int(input("Enter a number for last even power: "))
4 print("1",end=" ")
5 for i in range(2,n+1,2):
6     print("+ ",x**i, sep="", end=" ")
7     s+=x**i
```

Shell

```
>>> %Run '9 - sum of powers of even.py'
```

```
Enter the base number: 2
Enter a number for last even power: 6
1 + 4 + 16 + 64 = 85
```

10. Write a menu driven program to do the following operations:

- **To find factorial of a number**
- **To print multiplication table of a number**
- **To print divisors of a number**
- **To check if a number is prime or not**

```
1 v = int(input("""Choose 1,2 3 or 4:  
2 - To find factorial of a number  
3 - To print multiplication table of a number  
4 - To print divisors of a number  
5 - To check if a number is prime or not  
6 : """))  
7 if v == 1:  
8     f = 1  
9     n = int(input("Enter a number: "))  
10    for i in range(1,n+1):  
11        f*=i  
12        print(f)  
13 elif v == 2:  
14     n = int(input("Enter a number: "))  
15     for i in range(1,11):  
16         print(n*i)  
17 elif v == 3:  
18     n = int(input("Enter a number: "))  
19     for i in range(1,n+1):  
20         if n%i==0:  
21             print(i, end=", ")  
22 elif v == 4:  
23     n = int(input("Enter a number: "))  
24     for i in range(2,n):  
25         if n%i==0:  
26             print("not prime")  
27             break  
28     else:  
29         print("prime")  
30 else:  
31     print("invalid input")
```

Shell ×

```
>>> %Run '10 - menu driven program.py'  
Choose 1,2 3 or 4:  
- To find factorial of a number  
- To print multiplication table of a number  
- To print divisors of a number  
- To check if a number is prime or not  
: 1  
Enter a number: 7  
5040
```

11. Accept n1 and n2 from user. Write a menu driven program to do the following operations :

- Print sum of all numbers between n1 and n2
- Print sum of all even numbers between n1 and n2
- Print sum of all odd numbers between n1 and n2

The screenshot shows a Jupyter Notebook environment. On the left, there is a code cell containing Python code. On the right, there is a shell terminal window showing the execution of the code and its output.

```
1 v = int(input("""Choose 1,2 or 3:  
2 - Print sum of all numbers between n1 and n2  
3 - Print sum of all even numbers between n1 and n2  
4 - Print sum of all odd numbers between n1 and n2  
5 : """))  
6 n1 = int(input("Enter the first number: "))  
7 n2 = int(input("Enter the second number: "))  
8 if v==1:  
9     s = 0  
10    for i in range(n1+1,n2):  
11        s+=i  
12    print(s)  
13 elif v==2:  
14     s = 0  
15     if n1%2==0:  
16         for i in range(n1+2,n2,2):  
17             s+=i  
18     else:  
19         for i in range(n1+1,n2,2):  
20             s+=i  
21    print(s)
```

```
22 elif v==3:  
23     s = 0  
24     if n1%2!=0:  
25         for i in range(n1+2,n2,2):  
26             s+=i  
27     else:  
28         for i in range(n1+1,n2,2):  
29             s+=i  
30     print(s)  
31 else:  
32     print("Invalid choice")  
33  
34 # Run the code cell  
35  
36 # Output from the shell terminal  
37 >>> %Run '11 - menu driven program n1 n2.py'  
38 Choose 1,2 or 3:  
39 - Print sum of all numbers between n1 and n2  
40 - Print sum of all even numbers between n1 and n2  
41 - Print sum of all odd numbers between n1 and n2  
42 : 3  
43 Enter the first number: 4  
44 Enter the second number: 9  
45 12
```

NESTED FOR LOOPS:

1. WAP to print all the prime numbers between m and n.

```
nf 1 - prime | nf 2 - sum of cor nf 3 - factori nf 4
```

```
1 m = int(input("Enter the first number: "))
2 n = int(input("Enter the second number: "))
3 if m<1:
4     m = 1
5 for i in range(m+1,n):
6     for j in range(2,i):
7         if i%j==0:
8             break
9     else:
```

Shell ×

```
>>> %Run 'nf 1 - prime numbers bw n and m.py'
Enter the first number: 0
Enter the second number: 18
2
3
5
7
11
13
17
```

2. WAP to print the sum of all composite numbers between n1 and n2. Accept n1 and n2 from user

```
nf 2 - sum of comp | nf 3 - factorial | nf 4 - st | nf 5 -
```

```
1 m = int(input("Enter the first number: "))
2 n = int(input("Enter the second number: "))
3 if m<1:
4     m = 1
5 s = 0
6 for i in range(m+1,n):
7     for j in range(2,i):
8         if i%j==0:
9             s+=i
10            break
11 print(s)
```

Shell ×

```
>>> %Run 'nf 2 - sum of composite numbers bw n and m.py'
Enter the first number: 0
Enter the second number: 17
94
```

3. WAP to print the factorial of numbers between 1 and n. Accept n from user

```
1 n = int(input("Enter the second number: "))
2 for i in range(2,n):
3     f = 1
4     for j in range(1,i+1):
5         f*=j
6     print(i, f, sep=" - ")
```

Shell ×

```
>>> %Run 'nf 3 - factorials of numbers 1 to n.py'
Enter the second number: 9
2 - 2
3 - 6
4 - 24
5 - 120
6 - 720
7 - 5040
8 - 40320
```

4. WAP to print the following:

```
*  
*   *  
*   *   *  
*   *   *   *
```

```
1 for i in range(1,5):  
2     print("*"*i)  
  
Shell <input>  
>>> %Run 'nf 4 - star patt'  
*  
**  
***  
****
```

5. WAP to print (up to n terms)

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

```
1 n = int(input("Enter a number: "))  
2 for i in range(1,n+1):  
3     for j in range(1,i+1):  
4         print(j, end=" ")  
5     print()  
  
Shell <input>  
>>> %Run 'nf 5 - number pyramid.py'  
Enter a number: 6  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6
```

6. WAP to print

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

```
1 n = int(input("Enter a number: "))  
2 for i in range(1,n+1):  
3     for j in range(i,0,-1):  
4         print(j, end=" ")  
5     print()  
  
Shell <input>  
>>> %Run 'nf 6 - reverse number pyramid'  
Enter a number: 5  
1  
2 1  
3 2 1  
4 3 2 1  
5 4 3 2 1
```

7. WAP to print the triangle:

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

```
1 n = int(input("Enter a number: "))  
2 p=0  
3 for i in range(n):  
4     for j in range(n,p,-1):  
5         print(j, end=" ")  
6     p+=1  
7     print()  
  
Shell <input>  
>>> %Run 'nf 7 - reverse upside down num'  
Enter a number: 5  
5 4 3 2 1  
5 4 3 2  
5 4 3  
5 4  
5
```

8. WAP to print

5

5 4

5 4 3

5 4 3 2

5 4 3 2 1

```

1 n = int(input("Enter a number: "))
2 p=n
3 for i in range(n+1):
4     for j in range(n,p,-1):
5         print(j, end=" ")
6     p-=1
7 print()

```

Shell ×

>>> %Run 'nf 8 - reverse right side up'

Enter a number: 5

```

5
5 4
5 4 3
5 4 3 2
5 4 3 2 1

```

9. WAP to print

A

A B

A B C

A B C D

A B C D E

```

1 alpha = ["A","B","C","D","E"]
2 p = 1
3 for i in range(5):
4     for j in range(0,p):
5         print(alpha[j], end=" ")
6     if p==5:
7         break
8     else:
9         p+=1
10    print()

```

Shell ×

>>> %Run 'nf 9 - alphabet pyramid.py'

A

A B

A B C

A B C D

A B C D E

10. WAP to display the sum of the series $1 + 1/1! + 2/2! + 3/3! \dots n/n!$.

```

1 s = 1
2 n = int(input("Enter the number of terms: "))
3 print("1")
4 for i in range(1,n+1):
5     m = 1
6     for j in range(1,i+1):
7         m*=j
8     s+=i/m
9     print("+" ,i,"/",i,"!")
10    print("=",s)

```

Shell ×

>>> %Run 'nf 10 - n by n factorial series.py'

Enter the number of terms: 5

```

1
+ 1 / 1 !
+ 2 / 2 !
+ 3 / 3 !
+ 4 / 4 !
+ 5 / 5 !
= 3.708333333333333

```

11. WAP to display the sum of the series $1 - 1/1! + 2/2! - 3/3! \dots n/n!$

nf 11 - alternate sign n by n factorial series.py

```
1 s = 1
2 n = int(input("Enter the number of terms: "))
3 print("1")
4 for i in range(1,n+1):
5     m = 1
6     l = i
7     for j in range(1,i+1):
8         m*=j
9     if i%2!=0:
10        i*=-1
11    s+=i/m
12    print("+ ",i,"/",l,"!", sep="")
13 print("=",s)
```

Shell

```
>>> %Run 'nf 11 - alternate sign n by n factorial.s
Enter the number of terms: 5
1
+ -1/1!
+ 2/2!
+ -3/3!
+ 4/4!
+ -5/5!
= 0.625
```

12. WAP to display the cosine series $1 - x^2/2! + x^4/4! - \dots x^n/n!$

```
1 s = 1
2 n = int(input("Enter the last term: "))
3 x = int(input("Enter the base number: "))
4 print("1")
5 for i in range(2,n+1,2):
6     m = 1
7     l = i
8     for j in range(1,i+1):
9         m*=j
10    if i%4!=0:
11       b = -x
12       sign = "-"
13    else:
14       sign = "+"
15    s+= (b**i)/m
16    print(sign+x^",i,"/",l,"!", sep="")
```

Shell

```
>>> %Run 'nf 12 - alternating powers of x by factorial.py'
Enter the last term: 8
Enter the base number: 2
1
-x^2/2!
+x^4/4!
-x^6/6!
+x^8/8!
= 3.761904761904762
```

13. Program Write a program to find sum of the following series:

(1)+(1+2)+(1+2+3)+(1+2+3+4)+...upto N terms

nf 13 - number of brackets.py

```
1 n = int(input("Enter the number of terms: "))
2 ts = 0
3 for i in range(1,n+1):
4     s = 0
5     print("(0",end="")
6     for j in range(1,i+1):
7         s+=j
8         print("+",j,sep="",end="")
9     print(")", end=" + ")
10    ts+=s
11 print("0 =",ts)
```

Shell

```
>>> %Run 'nf 13 - number of brackets.py'
Enter the number of terms: 4
(0+1) + (0+1+2) + (0+1+2+3) + (0+1+2+3+4) + 0 = 20
```

STRING MANIPULATION

1. WAP to count the number of characters in an input string.
2. WAP to count the number of vowels in an input string.
3. WAP to enter a word and check if the word comes before or after “CBSE” in alphabetic order.
4. WAP to input your name and print each character of your name in same line with a space between.
5. WAP to check if a user entered string is a palindrome or not.
6. WAP to replace all characters other than alphabets and numbers to space.
7. WAP to capitalize the first letter of the user entered string.
8. WAP that capitalizes every alternate letter of the user entered string.
9. WAP that enters a phone no with 10 characters and 2 dashes. Check if phone no is valid.

(Phone no format – 017-555-3462)

10. WAP to enter a string and check if there are digits in the string. If digits are present find sum of digits otherwise print the message “No digits”.
 11. WAP to enter a string and count the number of words, no of alpha numeric characters and special characters.
 12. WAP to enter a string and convert all upper case to lower case and vice versa.
 13. WAP to keep on entering strings till user says ‘y’ and convert all upper case to lower case and vice versa.
-

```
1 str1 = input("Enter a string value: ")
2 print("Number of characters in string =",len(str1))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: Hello World
Number of characters in string = 11
```

```
1 vowels = ["a","e","i","o","u"]
2 count = 0
3 str1 = input("Enter a string value: ")
4 for i in str1:
5     if i.lower() in vowels:
6         count+=1
7 print("Number of vowels in the string =",count)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: Hello world :)
Number of vowels in the string = 3
```

```
1 str1 = input("Enter a word: ")
2 if str1.upper()>"CBSE":
3     print("The word comes after CBSE in the dictionary")
4 else:
5     print("The word comes before CBSE in the dictionary")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a word: ICSE
The word comes after CBSE in the dictionary
```

```
1 name = input("Enter your name: ")
2 for i in name:
3     print(i, end=" ")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter your name: Eesha
E e s h a
```

```
1 str1 = input("Enter a word: ")
2 rev = str1[::-1]
3 if rev.lower()==str1.lower():
4     print("The word is a palindrome")
5 else:
6     print("The word is not a palindrome")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a word: racecar
The word is a palindrome
```

```
1 str1 = input("Enter a string value: ")
2 output = ""
3 for i in str1:
4     if not i.isalnum():
5         output+=" "
6     else:
7         output+=i
8 print(output)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a string value: I <3 Python!!
I 3 Python
```

```
1 str1 = input("Enter a string value: ")  
2 print(str1.capitalize())
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT  
Enter a string value: hello world  
Hello world
```

```
1 str1 = input("Enter a string value: ")  
2 for i in range(0,len(str1),2):  
3     if i==len(str1)-1:  
4         print(str1[i].lower())  
5     else:  
6         print(str1[i].lower()+str1.upper()[i+1], end="")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT  
Enter a string value: Hello World  
hElLo wOrLd
```

```
1 no = input("Enter your phone number: ")  
2 if len(no)!=12:  
3     print("Invalid entry")  
4 elif no[3]+no[7]!="--":  
5     print("Invalid entry")  
6 elif not (no[:3]+no[4:7]+no[8:]).isdigit():  
7     print("Invalid entry")  
8 else:  
9     print("Valid number :)")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT  
Enter your phone number: +91-456-4567  
Invalid entry
```

```
1 str1 = input("Enter a string value: ")
2 s = 0
3 for i in str1:
4     if i.isdigit():
5         s+=int(i)
6 if s==0:
7     print("No digits")
8 else:
9     print("sum of digits in",str1,"=",s)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: A1b2c3D4
sum of digits in A1b2c3D4 = 10
```

```
1 str1 = input("Enter a string value: ")
2 an = d = sc = 0
3 for i in str1:
4     if i.isalnum():
5         an+=1
6         if i.isdigit():
7             d+=1
8     else:
9         sc+=1
10 print("Number of digits:",d)
11 print("Number of alpha numeric characters:",an)
12 print("Number of special characters:",sc)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: Hello World! 1234 ^v^
Number of digits: 4
Number of alpha numeric characters: 15
Number of special characters: 6
```

```
1 str1 = input("Enter a string value: ")
2 output = ""
3 for i in str1:
4     if i.islower():
5         output+=i.upper()
6     elif i.isupper():
7         output+=i.lower()
8     else:
9         output+=i
10 print(output)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: Hello World!
hELLO WORLD!
```

```
1 cont = 1
2 while cont!="y":
3     str1 = input("Enter a string value: ")
4     output = ""
5     for i in str1:
6         if i.islower():
7             output+=i.upper()
8         elif i.isupper():
9             output+=i.lower()
10        else:
11            output+=i
12    print(output)
13    cont = input("Enter 'y' to stop the program: ")
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a string value: python
PYTHON
Enter 'y' to stop the program: no
Enter a string value: PROGRAMMING
programming
Enter 'y' to stop the program: no
Enter a string value: Computer Science
COMPUTER SCIENCE
Enter 'y' to stop the program: y
```

LIST MANIPULATION

1. WAPS to find the minimum element from a list of elements along with index.
 2. WAPS to find mean of a given list.
 3. WAPS to search for an element in the list.
 4. WAPS to count the frequency of given element in list of numbers.
 5. WAPS to reverse an array of integers.
 6. WAP to enter n values from the user. If the entered values are multiples of 9, then add them to the list.
 7. WAP that adds 5 to all odd values and 10 to all even values in the list.
 8. WAP to find the sum of all values which are ending with 3 from the list.
 9. WAP to enter +ve and -ve numbers from the user to a list. Create 2 lists to store these numbers separately.
 10. WAP to enter 2 lists. Create a new list which will have 1st element of the 1st list, then 1st element of the 2nd list and so on till the end.
 11. WAP that takes two lists of same size and add their values together to form a new list, whose elements are sums of the corresponding elements of list1 and list 2.
 12. WAP to accept a list and display:
 - a. Alternate elements
 - b. Sum and average of all the elements
 - c. Even elements and odd elements
 13. WAP to find the maximum and minimum element in the list.
 14. WAP to Search for a particular element and give its first occurrence.
-

- 15.WAP to count number of times an element occurs in List.
- 16.WAP to find frequencies of all elements of a list. Also, print the list of unique elements in the list and duplicate elements in the given list.
- 17.WAP to reverse a list.
- 18.WAP to exchange the values of the first half of the array with the second half of the array.
- a. For example, if the array contains:
 - i. 2, 4, 1, 6, 7, 9, 23, 10
 - b. The resultant array is:
 - i. 7, 9, 23, 10, 2, 4, 1, 6
- 19.WAP to count the number of elements within a given range.
- 20.WAP to transfer contents of a list to two different lists ODD and EVEN. The odd list should contain elements from odd positions and even list should have elements from even positions.
- 21.WAP to modify the contents of the list in such a way that the elements that are multiples of 10 are swapped with the value present next to it.
- 22.WAP to create a list and replace elements having odd values with thrice their values and elements having even values with twice their values.
- 23.WAP to shift elements of a list so that first element moves to second element and so on. Last element needs to be placed at first location.
- 24.Write a Python program to find the list of words that are longer than n from a given list of words.
- 25.WAP to accept 2 lists in ascending. Merge these lists into a third one in ascending order.
-

```
1 l = eval(input("Enter a list of numbers: "))
2 m = min(l)
3 print("The element with the lowest value is",m,"at index",l.index(m))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list of numbers: [1,2,3,4,-1,2,-9,0,9,8,-6]
The element with the lowest value is -9 at index 6
```

```
1 l = eval(input("Enter a list of numbers: "))
2 print("mean of numbers in the list =",sum(l)/len(l))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list of numbers: [11,22,33,44,55]
mean of numbers in the list = 33.0
```

```
1 l = eval(input("Enter a list: "))
2 find = input("Enter the element to find: ")
3 print(find, "is located at index", l.index(find), "in",l)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list: ["a", "b", "c", "d", "e"]
Enter the element to find: e
e is located at index 4 in ['a', 'b', 'c', 'd', 'e']
```

```
1 l = eval(input("Enter a list: "))
2 element = input("Enter the element: ")
3 print("frequency of",element,"in",l,"is",l.count(element))
```

Shell ×

```
%Run -c $EDITOR_CONTENT
```

```
Enter a list: ["a", "b", "c", "d", "a", "e", "a", "f"]
Enter the element: a
frequency of a in ['a', 'b', 'c', 'd', 'a', 'e', 'a', 'f'] is 3
```

```
1 l = eval(input("Enter a list of numbers: "))
2 print(l[::-1])
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list of numbers: [1,2,3,4,5]
[5, 4, 3, 2, 1]
```

```
1 l1 = eval(input("Enter list of values: "))
2 nines = []
3 for i in l1:
4     if i%9==0:
5         nines+=[i]
6 print(nines)
```

Shell ×

```
Enter list of values: [1,9,34,27]
[9, 27]
```

```
1 l1 = eval(input("Enter a list of numbers: "))
2 l2 = []
3 for i in l1:
4     if i%2==0:
5         l2+=[i+10]
6     else:
7         l2+=[i+5]
8 print(l2)
```

Shell ×

```
>>> %Run 'list worksheet 7.py'
Enter a list of numbers: [0,1,2,3,4,5]
[10, 6, 12, 8, 14, 10]
```

```
1 l1 = eval(input("Enter a list of numbers: "))
2 s = 0
3 for i in l1:
4     if i%10==3:
5         s+=i
6 print(s)
```

Shell ×

```
>>> %Run 'list worksheet 8.py'
Enter a list of numbers: [3,6,9,33,54,43]
79
```

```
1 l1 = eval(input("Enter a list of numbers: "))
2 positive = []
3 negative = []
4 for i in l1:
5     if i>0:
6         positive+=[i]
7     elif i<0:
8         negative+=[i]
9 print(f"the positive numbers from the list are {positive}")
10 print(f"the negative numbers from the list are {negative}")
```

Shell ×

```
>>> %Run 'list worksheet 9.py'
Enter a list of numbers: [0,1,-3,4,6,5,-9,-7,5,-6,3,5]
the positive numbers from the list are [1, 4, 6, 5, 5, 3, 5]
the negative numbers from the list are [-3, -9, -7, -6]
```

```
1 l1 = eval(input("Enter the first list: "))
2 l2 = eval(input("Enter the second list: "))
3 newl = []
4 for i in range(len(l1)):
5     newl+=[l1[i]]
6     newl+=[l2[i]]
7 print(newl)
```

Shell ×

```
>>> %Run 'list worksheet 10.py'
Enter the first list: [2,4,6,8,10]
Enter the second list: [1,3,5,7,9]
[2, 1, 4, 3, 6, 5, 8, 7, 10, 9]
```

```
1 l1 = eval(input("Enter the first list: "))
2 l2 = eval(input("Enter the second list: "))
3 newl = []
4 for i in range(len(l1)):
5     s = l1[i]+l2[i]
6     newl+=[s]
7 print(newl)
```

Shell ×

```
>>> %Run 'list worksheet 11.py'
Enter the first list: [1,2,3,4,5,6]
Enter the second list: [10,20,30,40,50,60]
[11, 22, 33, 44, 55, 66]
```

```
1 l1 = eval(input("Enter a list of numbers: "))
2 m = int(input(""))
3 Choose 1, 2 or 3:
4 1. alternate elements
5 2. sum and average of all the elements
6 3. even elements and odd elements
7 : """))
8 if m==1:
9     for i in range(0,len(l1),2):
10         print(l1[i], end=" ")
11 elif m==2:
12     s = 0
13     for i in l1:
14         s+=i
15     a = s/len(l1)
16     print(f"sum = {s}")
17     print(f"average ={a}")
18 elif m==3:
19     even = []
20     odd = []
21     for i in l1:
22         if i%2==0:
23             even.append(i)
24         else:
25             odd.append(i)
26     print("Even numbers in list: ",even)
27     print("Odd numbers in list: ",odd)
28 else:
29     print("Invalid Input")
```

Shell ×

```
>>> %Run 'list worksheet 12.py'

Enter a list of numbers: [1,2,3,4,5,6,7,8,9,10,-1,-2,-3,0]

Choose 1, 2 or 3:
1. alternate elements
2. sum and average of all the elements
3. even elements and odd elements
: 2
sum = 49
average =3.5
```

```
1 l1 = eval(input("Enter a list of numbers: "))
2 print("Maximum value:",max(l1))
3 print("Minimum value:",min(l1))
```

Shell ×

```
>>> %Run 'list worksheet 13.py'
Enter a list of numbers: [1,2,3,4,5,0,-3,-1,-2,-4]
Maximum value: 5
Minimum value: -4
```

```
1 l = eval(input("Enter a list: "))
2 find = input("Enter the element to find: ")
3 print(find, "is first located at index", l.index(find), "in",l)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list: ["a", "b", "c", "a", "b", "d"]
Enter the element to find: b
b is first located at index 1 in ['a', 'b', 'c', 'a', 'b', 'd']
```

```
1 l = eval(input("Enter a list: "))
2 element = input("Enter the element: ")
3 print("frequency of",element,"in",l,"is",l.count(element))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list: ["a", "b", "c", "a", "b", "d"]
Enter the element: a
frequency of a in ['a', 'b', 'c', 'a', 'b', 'd'] is 2
```

```
2 unique = []
3 duplicate = []
4 for i in l:
5     if i not in duplicate:
6         if l.count(i)>1:
7             duplicate.append(i)
8     else:
9         unique.append(i)
10 print("Unique elements:",unique, "all with frequency = 1")
11 print("Duplicate elements:",duplicate, "with frequencies listed below")
12 for i in duplicate:
13     print(i,":",l.count(i))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list: [1,2,3,4,5,6,5,4,3,2,1,0,-9,-4,-3,7,5,3]
Unique elements: [6, 0, -9, -4, -3, 7] all with frequency = 1
Duplicate elements: [1, 2, 3, 4, 5] with frequencies listed below
1 : 2
2 : 2
3 : 3
4 : 2
5 : 3
```

```
1 l = eval(input("Enter a list of numbers: "))
2 print(l[::-1])
3
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list of numbers: [1,2,3,4,5]
[5, 4, 3, 2, 1]
```

```
1 l = eval(input("Enter a list: "))
2 m = len(l)//2
3 l[m:], l[:m] = l[:m], l[m:]
4 print(l)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list: [1,2,3,4,5,6]
[4, 5, 6, 1, 2, 3]
```

```
1 l = eval(input("Enter a list: "))
2 start = int(input("Enter the starting range: "))
3 end = int(input("Enter the ending range: "))
4 print("Number of elements in the given range =",len(l[start:end+1]))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list: [1,2,3,4,5,6]
Enter the starting range: 2
Enter the ending range: 5
Number of elements in the given range = 4
```

```
1 l = eval(input("Enter a list: "))
2 ODD = l[::2]
3 EVEN = l[1::2]
4 print("Elements in odd positions (not odd indices): ",ODD)
5 print("Elements in even positions (not even indices): ",EVEN)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list: [1,2,3,4,5,6]
Elements in odd positions (not odd indices): [1, 3, 5]
Elements in even positions (not even indices): [2, 4, 6]
```

```
1 l = eval(input("Enter a list: "))
2 i = 0
3 while i<len(l)-1:
4     if l[i]%10==0:
5         l[i], l[i+1] = l[i+1], l[i]
6         i+=2
7     else:
8         i+=1
9 print(l)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a list: [10,15,20,25,30,35]
[15, 10, 25, 20, 35, 30]
```

```
1 l = eval(input("Enter a list: "))
2 output = []
3 for i in l:
4     if i%2==0:
5         output.append(2*i)
6     else:
7         output.append(3*i)
8 print(output)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list: [1,2,3,4,5,6]
[3, 4, 9, 8, 15, 12]
```

```
1 l = eval(input("Enter a list: "))
2 for i in range(1,len(l)):
3     l[0], l[i] = l[i], l[0]
4 print(l)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list: [1,2,3,4,5,6]
[6, 1, 2, 3, 4, 5]
```

```
1 l = eval(input("Enter a list of words: "))
2 word = input("Enter the word: ")
3 longer_words = []
4 for i in l:
5     if len(i)>len(word):
6         longer_words.append(i)
7 print("List of words longer than",word,"-->",longer_words)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a list of words: ["Hello", "World", "Welcome", "to", "Python", "Programming"]
Enter the word: "Computer"
List of words longer than "Computer" --> ['Programming']
```

```
1 l1 = eval(input("Enter the first list: "))
2 l2 = eval(input("Enter the second list: "))
3 print("Combined list -->",l1+l2)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter the first list: [1,2,3,4]
Enter the second list: [5,6,7,8]
Combined list --> [1, 2, 3, 4, 5, 6, 7, 8]
```

TUPLES

1. Create a tuple namely cars storing car names as elements. Write a program to print names of the cars along with the number.

E.g.

Cars= ('Toyota', 'Volvo', 'GM', 'BMW')

- 1 Toyota
- 2 Volvo.....

2. Create a tuple with Squares of numbers from 1-50
3. Create a tuple ('a', 'bb', 'ccc',) last element as 26 copies of z
4. Write a program that creates a tuple storing first 10 terms of Fibonacci series.
5. WAP to print the length of shortest string in a tuple of strings
6. Create two tuples a, b and print true if every element in a is also in b else prints False.
7. WAP that takes a Fibonacci term and returns a number telling which term it is.
E.g. 3 fifth term 8 return 7th term
8. Given tuple pairs ((2,5), (4,6), (9,8)) count the no. of pairs of (a, b) such that both a and b are even.
9. Create a tuple that stores mark in three subjects for five students.
Calculate total and average of each student.
10. Create a nested tuple. Display the mean of individual tuple also compute the mean of all computed means.
 $t=((1,2),(1,2,3),(1,3),(2,3,4,5))$

```
1 cars = ("Toyota", "Volvo", "GM", "BMW")
2 for i in range(len(cars)):
3     print(i+1, cars[i])
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
1 Toyota
2 Volvo
3 GM
4 BMW
```

```
1 t1 = tuple([i*i for i in range(1,51)])
2 print(t1)
```

Shell ×

```
(1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144,
169, 196, 225, 256, 289, 324, 361, 400, 441, 484
, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1
024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1
600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2
304, 2401, 2500)
```

```
1 t = tuple([i*chr(i+96) for i in range(1,27)])
2 print(t)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
('a', 'bb', 'ccc', 'dddd', 'eeeeee', 'ffffff', 'gggggggg', 'hhhhhhhh', 'iiiiiiii
', 'jjjjjjjjjjj', 'kkkkkkkkkkk', 'lllllllllllll', 'mmmmmmmmmmmmmm', 'nnnnnnnnnnnnn
n', 'ooooooooooooooo', 'pppppppppppppppp', 'qqqqqqqqqqqqqqqq', 'rrrrrrrrrrrrrr
rrrr', 'ssssssssssssssssss', 'ttttttttttttttttt', 'uuuuuuuuuuuuuuuuuuu
', 'vvvvvvvvvvvvvvvvvvv', 'wwwwwwwwwwwwwwwwww', 'xxxxxxxxxxxxxxxxxxxxxx
', 'yyyyyyyyyyyyyyyyyyyyyyyyyy', 'zzzzzzzzzzzzzzzzzzzzz')
```

```
1 t1 = (0,1)
2 o = 0
3 t = 1
4 for i in range(8):
5     n = o+t
6     t1+=(n,)
7     o, t = t, n
8 print(t1)
```

Shell ×

```
>>> %Run 4.py
(0, 1, 1, 2, 3, 5, 8, 13, 21, 34)
```

```
1 t = eval(input("Enter the tuple of strings: "))
2 t_lens = []
3 for i in t:
4     t_lens.append(len(i))
5 print("Length of the shorted string:",min(t_lens))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter the tuple of strings: ("aaaa", "bb", "ccc")
Length of the shorted string: 2
```

```
1 a = eval(input("Enter a tuple: "))
2 b = eval(input("Enter another tuple: "))
3 for i in a:
4     if i not in b:
5         print(False)
6         break
7 else:
8     print(True)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
Enter a tuple: (1,2,3,4,5,6)
Enter another tuple: (1,2,3,5)
False
```

```
1 term = int(input("Enter the term: "))
2 t1 = (0,1)
3 o = 0
4 t = 1
5 for i in range(30):
6     n = o+t
7     t1+=(n,)
8     o, t = t, n
9 if term<t1[-1] and term not in t1:
10    print("Not a fibonacci term")
11    exit()
12 elif term in t1:
13    print(term, " -->", t1.index(term))
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter the term: 233
233 --> 13
```

```
1 count = 0
2 t1 = eval(input("Enter a tuple of number pairs: "))
3 for i in t1:
4     if i[0]%2==0 and i[1]%2==0:
5         count+=1
6 print(count)
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
```

```
Enter a tuple of number pairs: ((1,2), (3,4), (5,7), (6,8), (8,5), (4,2), (10,12))
3
```

```

1 t = ()
2 for i in range(1,6):
3     print("student",i)
4     math = int(input("Enter the math marks: "))
5     eng = int(input("Enter the english marks: "))
6     science = int(input("Enter the science marks: "))
7     t+=((math,eng,science),)
8 print(t)
9 total = [sum(i) for i in t]
10 avg = [i/3 for i in total]
11 print("Total of each student:",total)
12 print("Avergae of each student:",avg)

```

Shell ×

```

>>> %Run -c $EDITOR_CONTENT
student 1
Enter the math marks: 99
Enter the english marks: 89
Enter the science marks: 98
student 2
Enter the math marks: 56
Enter the english marks: 63
Enter the science marks: 45
student 3
Enter the math marks: 78
Enter the english marks: 75
Enter the science marks: 68
student 4
Enter the math marks: 64
Enter the english marks: 24
Enter the science marks: 64
student 5
Enter the math marks: 46
Enter the english marks: 97
Enter the science marks: 46
((99, 89, 98), (56, 63, 45), (78, 75, 68), (64, 24, 64), (46, 97, 46))
Total of each student: [286, 164, 221, 152, 189]
Avergae of each student: [95.33333333333333, 54.666666666666664, 73.666666666666667, 50.666666666666664, 63.0]

```

```

1 t = eval(input("Enter a tuple of tuples: "))
2 means = ()
3 for i in t:
4     means+=sum(i)/len(i),
5 print("Average values of each tuple:",means)
6 print("Mean of all means:",sum(means)/len(means))

```

Shell ×

```

>>> %Run -c $EDITOR_CONTENT
Enter a tuple of tuples: ((1,2), (1,2,3), (1,3), (2,3,4,5))
Average values of each tuple: (1.5, 2.0, 2.0, 3.5)
Mean of all means: 2.25

```

DICTIONARY

1. WAP to create a dictionary country to store 4 country names and their capital. Display the capital in the given format.

Capital of <country> is <capital>

Capital of Austria is Vienna

2. WAP to create a phone book by accepting name, contact no and birthday of 5 of your friends.
3. Modify the above program to search for the details of the name taken as input.
4. WAP to repeatedly ask the user to enter product name and prices. Store all these in a dictionary whose keys are product names and whose values are the prices.

After entering products and prices, accept a product name and perform the below given functions displayed as a menu.

1. Add products.
2. Display price
3. Update price
4. Remove product.
5. Exit

5. Create a menu driven to do the following:

1. Add elements
2. Display keys
3. Display values
4. Display items
5. Display details of books where the price is more than 100.
6. Write a program to accept and store Employee name as key and his salary, allowance, and deductions as value in a dictionary Dict for 10 employees. Display the following menu:
 - a) Display Details
 - b) Display Total allowance and deductions.
 - c) Search for an employee.
 - d) Exit

When user selects option a, display a report having Name, Salary, Allowance, Deduction, Gross Salary (Salary + allowance) and Net Salary (Gross Salary-Deductions)

When the user selects option b, find total allowance and deductions for all the 10 employees.

When the user selects option c, accept the name of the employee and search in the dictionary and display the details.

(without using dictionary methods or functions)

```
1 country = {'azerbaijan': 'baku', 'india': 'new delhi',
2             'uae': 'abu dhabi', 'australia': 'canberra',
3             'turkey': 'ankara'}
4 for i in country:
5     print("Capital of",i,"is",country[i])
```

Shell ×

```
Capital of azerbaijan is baku
Capital of india is new delhi
Capital of uae is abu dhabi
Capital of australia is canberra
Capital of turkey is ankara
```

```
1 pb = {}
2 while True:
3     name = input("Enter name: ")
4     if name not in pb:
5         no = input("Enter phone number: ")
6         bday = input("Enter birthday: ")
7         pb[name] = (no, bday)
8     else:
9         print(name,"already exists in phonebook")
10    a = int(input("Do you want to add another person? (enter 1 for yes and 0 for no): "))
11    if a==0:
12        break
13 print(pb)
```

Shell ×

```
>>> %Run '2 - phonebook.py'
Enter name: Eesha
Enter phone number: 55555
Enter birthday: 25th may
Do you want to add another person? (enter 1 for yes and 0 for no): 1
Enter name: Bob
Enter phone number: 12345
Enter birthday: 25th March
Do you want to add another person? (enter 1 for yes and 0 for no): 0
{'Eesha': ('55555', '25th may'), 'Bob': ('12345', '25th March')}
```

```
1 pb = {}
2 while True:
3     name = input("Enter name: ")
4     if name not in pb:
5         no = input("Enter phone number: ")
6         bday = input("Enter birthday: ")
7         pb[name] = (no, bday)
8     else:
9         print(name,"already exists in phonebook")
10    a = int(input("Do you want to add another person? (enter 0 for no): "))
11    if a==0:
12        break
13 print("Phonebook: ",pb)
14 while True:
15     name=input("Enter name: ")
16     if name in pb:
17         print(pb[name])
18     else:
19         print(name,"not in phonebook")
20     a = int(input("Would you like anyone else's details? (enter 0 for no): "))
21     if a==0:
22         break
```

Shell <

```
>>> %Run '3 - phonebook search.py'

Enter name: Eesha
Enter phone number: 5555
Enter birthday: 25th May
Do you want to add another person? (enter 0 for no): 1
Enter name: Bob
Enter phone number: 12345
Enter birthday: 25th March
Do you want to add another person? (enter 0 for no): 0
Phonebook: {'Eesha': ('5555', '25th May'), 'Bob': ('12345', '25th March')}
Enter name: Bob
('12345', '25th March')
Would you like anyone else's details? (enter 0 for no): 0
```

```
pp = {}
while True:
    f = int(input(""))

Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: """))
    if f not in [0,1,2,3,4]:
        print("Invalid Input.")
    elif f==1:
        product=input("Enter product name: ")
        if product not in pp:
            price=input("Enter product price: ")
            pp[product] = price
        else:
            print(product,"already exists in book")
    elif f==2:
        product=input("Enter product name: ")
        if product in pp:
            print(pp[product])
        else:
            print(product,"does not exist in book")
    elif f==3:
        product=input("Enter product name: ")
        if product in pp:
            price=input("Enter new product price: ")
            pp[product] = price
        else:
            print(product,"does not exist in book")
    elif f==4:
        product=input("Enter product name: ")
        if product in pp:
            pp.pop(product)
            print(product,"has been removed from the book")
        else:
            print(product,"does not exist in book")
    elif f==0:
        break
```

```
Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: 1
Enter product name: Juice
Enter product price: 5
```

```
Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: 1
Enter product name: Chips
Enter product price: 7
```

```
Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: 2
Enter product name: Chips
7
```

```
Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: 3
Enter product name: Juice
Enter new product price: 2
```

```
Choose one and type the number of the function-
1) Add Products
2) Display price
3) Update price
4) Remove product
0) Exit: 4
Enter product name: Chips
Chips has been removed from the book
```

```
print("Welcome to the library!!")
lib = {}
while True:
    f = int(input(""))

Choose one and type the number of the function-
1) Add book
2) Display id of books in library
3) Display book details of books in library
4) Display all books and their details in library
5) Display details of books that cost over 100 rupees
6) Display author name
7) Display price
8) Display book name
9) Check if book exists in library
0) Exit: """))

    if f not in [0,1,2,3,4,5,6,7,8]:
        print("Invalid Input.")
    elif f==1:
        book=input("Enter book id: ")
        if book not in lib:
            bn=input("Enter book name: ")
            au=input("Enter book author: ")
            price=int(input("Enter book price: "))
            lib[book] = (bn, au, price)
        else:
            print(book,"already exists in lib")
    elif f==2:
        print(lib.keys())
    elif f==3:
        print(lib.values())
    elif f==4:
        print(lib.items())
    elif f==5:
        for i in lib:
            if lib[i][2]>100:
                print(lib[i])
    elif f==6:
        book=input("Enter book id: ")
        if book in lib:
            print(lib[book][1])
        else:
            print(book,"does not exist in library")
    elif f==0:
        break
```

```
Welcome to the library!!
```

```
Choose one and type the number of the function-
```

- 1) Add book
 - 2) Display id of books in library
 - 3) Display book details of books in library
 - 4) Display all books and their details in library
 - 5) Display details of books that cost over 100 rupees
 - 6) Display author name
 - 7) Display price
 - 8) Display book name
 - 9) Check if book exists in library
 - 0) Exit: 1
- ```
Enter book id: 12345
Enter book name: Bob
Enter book author: Eesha
Enter book price: 250
```

```
Choose one and type the number of the function-
```

- 1) Add book
  - 2) Display id of books in library
  - 3) Display book details of books in library
  - 4) Display all books and their details in library
  - 5) Display details of books that cost over 100 rupees
  - 6) Display author name
  - 7) Display price
  - 8) Display book name
  - 9) Check if book exists in library
  - 0) Exit: 4
- 
- ```
dict_items([('12345', ('Bob', 'Eesha', 250))])
```

```

emp_diary = {}
for i in range(10):
    name=input("Enter employee name: ")
    if name not in emp_diary:
        s=int(input("Enter salary: "))
        a=int(input("Enter allowance: "))
        d=int(input("Enter deduction: "))
        emp_diary[name] = (s, a, d)
    else:
        print(name,"already exists in diary")
while True:
    f = input("Choose one and type the number of the function-
a) Display details
b) Display total allowance and deductions
c) Search for an employee
d) Exit
: ")
    if f not in ["a", "b", "c", "d"]:
        print("Invalid Input.")
    elif f=="a":
        name = input("Enter employee name: ")
        if name not in emp_diary:
            print("Employee not registered")
        else:
            print("Employee:",name)
            print("Salary:",emp_diary[name][0])
            print("Allowance:",emp_diary[name][1])
            print("Deduction:",emp_diary[name][2])
            print("Gross salary:",emp_diary[name][0]+emp_diary[name][1])
            print("Net salary:",emp_diary[name][0]-emp_diary[name][2])
    elif f=="b":
        for i in emp_diary:
            a = d = 0
            a+=emp_diary[i][1]
            d+=emp_diary[i][2]
        print("Total allowance of all employees:",a)
        print("Total deduction of all employees:",d)
    elif f=="c":
        name = input("Enter employee name: ")
        if name not in emp_diary:
            print("Employee not registered")
        else:
            print(emp_diary[name])
    else:
        break

```

```
Enter employee name: a
Enter salary: 12
Enter allowance: 23
Enter deduction: 34
Enter employee name: b
Enter salary: 345
Enter allowance: 343
Enter deduction: 4534
Enter employee name: c
Enter salary: 435
Enter allowance: 56456
Enter deduction: 32432
Enter employee name: d
Enter salary: 543345
Enter allowance: 34234
Enter deduction: 45456
Enter employee name: e
Enter salary: 345345
Enter allowance: 34234
Enter deduction: 34234
Enter employee name: f
Enter salary: 67567
Enter allowance: 567567
Enter deduction: 567567
Enter employee name: g
Enter salary: 45345345
Enter allowance: 456456
Enter deduction: 345345
Enter employee name: h
Enter salary: 453454
Enter allowance: 345345
Enter deduction: 45345
Enter employee name: i
Enter salary: 4534534
Enter allowance: 456456
Enter deduction: 345345
Enter employee name: j
Enter salary: 34534534
Enter allowance: 34534534
Enter deduction: 453454
```

Choose one and type the number of the function-

- a) Display details
- b) Display total allowance and deductions
- c) Search for an employee
- d) Exit

: a

Enter employee name: e

Employee: e

Salary: 345345

Allowance: 34234

Deduction: 34234

Gross salary: 379579

Net salary: 311111

Choose one and type the number of the function-

- a) Display details
- b) Display total allowance and deductions
- c) Search for an employee
- d) Exit

: b

Total allowance of all employees: 34534534

Total deduction of all employees: 453454

Choose one and type the number of the function-

- a) Display details
- b) Display total allowance and deductions
- c) Search for an employee
- d) Exit

: c

Enter employee name: d

(543345, 34234, 45456)

Choose one and type the number of the function-

- a) Display details
- b) Display total allowance and deductions
- c) Search for an employee
- d) Exit

: e

Invalid Input.

Choose one and type the number of the function-

- a) Display details
- b) Display total allowance and deductions
- c) Search for an employee
- d) Exit

: d