



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF INFORMATION TECHNOLOGY
AND ENGINEERING**

WINTER SEMESTER 2021-22

SUBJECT: PYTHON PROGRAMMING

DIGITAL ASSIGNMENT-1

Course Code :ITA6013

REGISTER NUMBER: 21MCA0017

NAME: SANJAY NAIK

ASSESSMENT – 1

Name: SANJAY NAIK

Reg No: 21MCA0017

Exercise 1

Q.1 Currency converter

X, a currency conversion agent, serves his customers in exchanging currencies from Indian rupee to US Dollar. Design an algorithm and write a Python program to automate the process of finding the equivalent amount in US Dollar.

Input Format:

Indian rupee equivalent to one US Dollar

Amount to be converted in Indian rupees

Output Format:

Dollar equivalent for Indian money

Code:

```
rupees = float(input("Enter the amount in Indian Rupees: "))  
print(f"{rupees} Indian Rupees = {'%.2f'%(rupees/80)} US Dollars.")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe  
"E:/Python 2022/PRG1.py"
```

Enter the amount in Indian Rupees: 160.0

160 Indian Rupees = 1.67 US Dollars.

Q.2 Relay Race

In a ultra race, four runners are there and each of them cover equal distance. Given the total distance to be covered by the runners, design an algorithm and write a Python code to determine the Km and meter of distance to be covered by each runner? For example, if the total distance to be covered is 2 Km and 500 m then the distance to be covered by each runner is 0Km and 625 m.

Input Format:

Read Km of total distance to be covered

Read meter of total distance to be covered

Output Format:

Print Km of distance to be covered by each runner

Print meter of distance to be covered by each runner

Code:

```
total_dis_km = int(input("Enter the total distance covered KM:"))
total_dis_m = int(input("Enter the total distance covered M:"))
total_distance = (total_dis_km*1000) + total_dis_m
print(f"Distance covered by each runner: {total_distance%4}KM and {total_distance/4}M")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/PRG2.py"

Enter the total distance covered KM: 2

Enter the total distance covered M: 500

Distance covered by each runner: 0KM and 625.0M

Process finished with exit code 0

Q.3 Length of diagonal in Rectangle

The perimeter of a rectangle is 46cm. If the length of the rectangle is 15cm, then what is the length of the diagonal of the rectangle? (Hint: **use Pythagorus theorem to solve**).

Input Format

Perimeter of rectangle

length of rectangle

Output format

length of diagonal

Code:

```
import math
perimeter = int(input("Enter the perimeter of the rectangle: "))
length = int(input("Enter the length of the rectangle: "))
breadth = (perimeter / 2) - length
diagonal = math.sqrt(length**2 + breadth**2)
print(f"Length of Diagonal= {diagonal}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/PRG3.py"

Enter the perimeter of the rectangle: 46

Enter the length of the rectangle: 15

Length of Diagonal= 17.0

Process finished with exit code 0

Q.4 BMI Calculator

A person from USA wants to know his Body Mass Index (BMI). He knows his weight in pounds and height in inches. The evaluator knows the formula for calculating BMI using the formula,

$$\text{BMI} = (\text{weight in kilograms}) / (\text{height in m} * \text{height in m})$$

Help the person in finding his BMI by writing a program for him. (Use the conversion formulae: 1 pound = 0.4536 kilograms, 1 inch = 2.54 cms)

Input Format:

Weight of person in pounds.

Height of the person in inches.

Output Format:

BMI of the person calculated using the formula,

$$(\text{weight in kilograms}) / (\text{height in m} * \text{height in m})$$

Code:

```
weigh_in_pounds = float(input("Enter Weight of person in pounds: "))
height_in_inches = float(input("Enter Height of the person in inches: "))
weight_in_kg = weigh_in_pounds/2.205
height_in_m = height_in_inches/39.37
BMI = weight_in_kg/height_in_m**2
print(f"BMI of the person calculated using the formula,(weight in kilograms)/(height in m * height in m): {'%.2f'%BMI}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/PRG4.py"

Enter Weight of person in pounds: 139

Enter Height of the person in inches: 78

BMI of the person calculated using the formula,(weight in kilograms)/(height in m * height in m): 16.06

Process finished with exit code 0

Q.5 A company wants its employees to work for 'X' hours on average per day in a week (Monday to Friday). Given the number of hours worked by an employee on each day of a week, design a flowchart and write a Python code to compute the average number of hours worked by the employee. Number of hours worked can be floating point values. For example, 7 hours 30 minutes is entered as 7.5 hours

Input Format:

Number of hours worked on first day

Number of hours worked on second day

Number of hours worked on third day

Number of hours worked on fourth day

Number of hours worked on fifth day

Output Format:

Average hours worked in a week

Code:

```
work_hour_day1 = float(input("Enter number of hours worked on monday:
"))
work_hour_day2 = float(input("Enter number of hours worked on tuesday:
"))
work_hour_day3 = float(input("Enter number of hours worked on wednesday:
"))
work_hour_day4 = float(input("Enter number of hours worked on thursday:
"))
work_hour_day5 = float(input("Enter number of hours worked on friday:
"))

avg_work_hour =
(work_hour_day5+work_hour_day4+work_hour_day3+work_hour_day2+work_hour_d
ay1)/5
print(f"Average hours worked in a week: {avg_work_hour}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe

"E:/Python 2022/PRG5.py"

Enter number of hours worked on monday: 6

Enter number of hours worked on tuesday: 4.5

Enter number of hours worked on wednesday: 7

Enter number of hours worked on thursday: 7.5

Enter number of hours worked on friday: 6

Average hours worked in a week: 6.2

Process finished with exit code 0

Exercise 1B

Calculation of number of plants in a farm

A person 'X' has a farm which is rectangular in shape. 'X' wants to plant coconut trees in the farm and he has heard that every tree has to be separated by 2 feet. He needs to calculate the number of plants to be purchased. Write a program to calculate the number of plants when provided with the length and breadth of the farm (in feet). The program should display the number of rows and columns along with the total number of plants required for the farm. For example, if the length and breadth of the farm is 9 feet * 4 feet then the farmer can plant trees in positions 0, 2, 4, 6, 8 along the length and 0,2,4 along the breadth therefore number of trees to be purchased is $5 \times 3 = 15$.

Input Format:

The first input value read is the length of the farm (in feet).

The second input value read is the breadth of the farm(in feet).

Output Format:

Print number of rows

Print number of columns

Print number of plants required

```
length = int(input("Enter the length in feet: "))
breadth = int(input("Enter the breadth in feet: "))
row = (length // 2) + 1
column = (breadth // 2) + 1
total = row * column
print("Number of rows: ", row)
print("Number of columns: ", column)
print("Number of plant required: ", total)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe"E:/Python2022/EX1BP1.py"

Enter the length in feet: 5

Enter the breadth in feet: 4

Number of rows: 3

Number of columns: 3

Number of plant required:

9Process finished with exit code 0

Interest rate calculator

Raman, a finance consultant decides to create an application for the public who wants to decide on the type of investment they can make. He needs an application, which when given with the initial amount, no of years of investment 'n' and the expected amount after 'n' years, should tell the rate of interest based on which the customers will choose their investments. Assume that the application works with only simple interest calculation. Can you code for Raman? Use

$$\text{Rate of interest} = ((\text{expected amount} - \text{initial amount}) / (\text{initial amount} * \text{no of years})) * 100$$

Input Format:

The first input value read is the initial amount the customer has.

The second input value read is the no of years he want to invest.

The third input value is the expected amount after 'n' years.

Output Format:

Print rate of interest

```
initial = int(input("Enter the initial amount: "))
year = int(input("Enter the years: "))
expAmo = int(input("Enter the expected amount: "))
ROI = ((expAmo - initial) / (initial * year)) * 100
print("The Rate of interest will be: ", ROI)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX1BP2.py"

Enter the initial amount: 1000

Enter the years: 5

Enter the expected amount: 25000

The Rate of interest will be: 480.0

Process finished with exit code 0

Bill generator

XYZ is working as a cashier in a Department store. He uses a bill book, in which he has to enter the item code, no of quantities purchased and then the price of each item. He then calculates the total amount to be given by the customer. Assume that a customer can purchase only three types of items at a time). Write a program to automate the process.

Input Format:

The first input value read is the quantity of first item.

The second input value read is the price per first item.

The third input value read the quantity of second item.

The fourth input value read is the price per second item.

The fifth input value read the quantity of third item.

The sixth input value read is the price per third item.

Output Format:

Total amount to be paid by the customer

```
item1_quantity = int(input("Enter quantity of first item: "))
item1_price = int(input("Enter price of first item: "))
item2_quantity = int(input("Enter quantity of second item: "))
item2_price = int(input("Enter price of second item: "))
item3_quantity = int(input("Enter quantity third item: "))
item3_price = int(input("Enter price of third item: "))
total = (item1_quantity * item1_price) + (item2_quantity * item2_price) +
(item3_quantity * item3_price)
print("Total amount to be paid by the customer: ", total)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX1BP3.py"

Enter quantity of first item: 25

Enter price of first item: 400

Enter quantity of second item: 12

Enter price of second item: 456

Enter quantity third item: 25

Enter price of third item: 148

Total amount to be paid by the customer: 9797

Process finished with exit code 0

Thirsty Crow

Consider the thirsty crow story where a thirsty crow identifies a jug with little water. It puts pebbles into the water to raise the level of water and drinks it. Assume that the initial reading of the jug is 'm1' ml and the crow can drink water if the level of water has come to 'm2' ml. There are two categories of pebbles small and big in the field. Small pebble can raise the level of water by 'x' ml and big pebble can raise the level of water by 'y' ml. There are 'n' small pebbles. Crow prefers to place small pebbles in jug and then only takes big pebbles. Write an algorithm and the Python code to determine the number of pebbles required to raise the water to 'm2' level. **For example, if value of 'm1', 'm2','x','y' and 'n' are 54, 300, 10, 20, 10 respectively then the number of pebbles required is 13.**

Input Format:

Read the initial level of water in jug (in ml)
Read the level of water in jug required for drinking (in ml)
Read the height which small pebble will increase (in ml)
Read the height which big pebble will increase (in ml)
Read the number of small pebbles

Output Format:

Number of big pebbles required

```
initial_level = 54
required_level = 300
height_increased_by_small_pebbles = 10
height_increased_by_big_pebbles = 20
number_of_small_pebbles = 10

raise_level = required_level - initial_level
print(f"Raise Level: {raise_level}")
raised_by_small_pebbles = raise_level/number_of_small_pebbles *
height_increased_by_small_pebbles
print(f"Raised by small pebbles: {raised_by_small_pebbles}")
number_of_big_pebbles = (raise_level -
raised_by_small_pebbles)/height_increased_by_big_pebbles

print(number_of_big_pebbles)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX1BP4.py"

Raise Level: 246

Raised by small pebbles: 246.0

0.0

Process finished with exit code 0

Work and Men

A man engaged 'n' labourers to make 't' toys in 'd' days. Assume that all men work with same speed and efficiency. After 'd1' days, he found that only 't1' toys were made. Design an algorithm and write a Python code to determine the number of additional men to be employed to complete the task in time. For example, if n is 10, t is 320, d is 5, d1 is 3, and t1 is 120 then the number of additional men to be employed is 12. Assume that the speed of making toys is uniform for all men.

Input Format:

Read the number of labourers engaged in work

Read the total number of toys to be made (t)

Read the total number of days allotted for completion (d)

Read the number of days work had been done (d1)

Read the number of toys made in d1 days (t1)

Output Format:

Number of more men required for completing the job in allotted period

```
n = int(input("number of labourers engaged in work: "))
t = int(input("total number of toys to be made : "))
d = int(input("total number of days allotted for completion : "))
d1 = int(input("the number of days work had been done: "))
t1 = int(input("number of toys made in d1 days: "))
x = ((n * d1) * (t - t1)) / ((d - d1) * t1) - n
print("Number of men required for completing the job: ", x)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX1BP5.py"

"E:/Python

number of labourers engaged in work: 25

total number of toys to be made : 320

total number of days allotted for completion : 5

the number of days work had been done: 3

number of toys made in d1 days: 120

Number of men required for completing the job: 37.5

Process finished with exit code 0

Exercise 2

Q.1 Students in a class are appreciated based on the following factors

Number of 'S' grade ≥ 3

Attendance ≥ 90

Participation in sports activity in a semester ≥ 2

Appreciation is given as follows:

(i) 'Excellent' if all three conditions are met

(ii) 'Very Good' if conditions (i) and (ii) are met

(iii) 'Good' if conditions (i) and (iii) are met

Given the Number of 'S' grades, Attendance and Participation in sports activity in a semester, write the python code to output the appreciation for the student. Check boundary conditions and print 'Invalid input' for wrong input.

Boundary Condition:

All values of input ≥ 0

Code:

```
n = int(input("number of S grade: "))
t = int(input("total attendance: "))
p = int(input("Participation in sports activity in a semester: "))
if n <= 0 or t <= 0 or p <= 0:
    print("Invalid Input")
elif n >= 3 and t >= 90 and p >= 2:
    print("EXCELLENT")
elif n >= 3 and t >= 90:
    print("VERY GOOD")
elif n >= 3 and p >= 2:
    print("GOOD")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX2P1.py"

number of S grade: 3

total attendance: 5

Participation in sports activity in a semester: 2

GOOD

Process finished with exit code 0

Q.2 Design an algorithm and write the Python code to convert time in 24-hour format to 12-hour format. For example, if the input time is 16:45:12 then the output should be 4:45:12 PM. Check boundary conditions and print 'Invalid input' for wrong input.

Boundary Condition:

All input ≥ 0

Hours < 24

Minutes < 59

Seconds < 59

Code:

```
hr = int(input("Enter Hour: "))
mn = int(input("Enter minute: "))
sec = int(input("Enter seconds: "))
if 0 < hr < 24 and 0 < mn < 59 and 0 < sec < 59:
    print(f"The time in 24 hour format {hr}:{mn}:{sec}")

if hr > 12:
    hr = hr - 12
    print(f"The time in 12 hour format {hr}:{mn}:{sec}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX2P2.py"

"E:/Python

Enter Hour: 13

Enter minute: 54

Enter seconds: 25

The time in 24 hour format 13:54:25

The time in 12 hour format 1:54:25

Process finished with exit code 0

Q.3 Write a python program to display the grades according to the student marks in five different subjects. Check boundary conditions on your own intuition.

```
print("enter the marks of five subjects: ")
marks = []
for i in range(5):
    sub = int(input())
    marks.append(sub)

for j in range(5):
    if marks[j] >= 90:
        print(str(marks[j]) + "-->the grade is 'S'")
    elif 80 <= marks[j] < 90:
        print(str(marks[j]) + "-->the grade is 'A'")
    elif 70 <= marks[j] < 80:
        print(str(marks[j]) + "-->the grade is 'B'")
    elif 60 <= marks[j] < 70:
        print(str(marks[j]) + "-->the grade is 'C'")
    elif 50 <= marks[j] < 60:
        print(str(marks[j]) + "-->the grade is 'D'")
    elif 40 <= marks[j] < 50:
        print(str(marks[j]) + "-->the grade is 'E'")
    else:
        print(str(marks[j]) + "-->the grade is 'F'")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python
2022/EX2P3.py"

enter the marks of five subjects:

95

65

25

45

65

95-->the grade is 'S'

65-->the grade is 'C'

25-->the grade is 'F'

45-->the grade is 'E'

65-->the grade is 'C'

Process finished with exit code 0

Q.5 Write a python code to find the sum of the following series. Get N from the user.

$1+2+3+4+5+\dots+N$

$2+4+6+8+10+\dots+N$

Code:

```
n = int(input("Enter the number of terms: "))
print(f"Sum of Series1: {n*(n+1)/2}")
print(f"Sum of Series2: {n*(n+1)}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python
2022/EX2P5.py"
Enter the number of terms: 5
Sum of Series1: 15.0
Sum of Series2: 30
```

Process finished with exit code 0

Q.6 A palindromic number (also known as a numeral palindrome or a numeric palindrome) is a number (such as 16461) that remains the same when its digits are reversed. Write an algorithm/pseudo code, draw a flow chart and write the python code to check whether the given number is palindrome or not. Check boundary conditions and print 'Invalid input' for wrong input.

Boundary Condition:

Number>0

```
"""Q.6 A palindromic number (also known as a numeral palindrome or a numeric
palindrome) is a number (such as 16461)
that remains the same when its digits are reversed. Write an algorithm/pseudocode, draw
a flow chart and write the
python code to check whether the given number is palindrome or not. Check boundary
conditions and print 'Invalid
input' for wrong input. Boundary Condition: Number>0
"""

num = int(input("Enter any number: "))
if num <= 0:
    print("Enter a number greater than 0.")
else:
    temp = num
    sum = 0
    while num > 0:
        digit = num % 10
        sum = (sum*10) + digit
        num = num//10
    if temp == sum:
        print(f"{temp} is a Palindrome Number.")
    else:
        print(f"{temp} isn't a Palindrome Number.")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX2P6.py"

"E:/Python

Enter any number: 121

121 is a Palindrome Number.

Process finished with exit code 0

Q.7

Draw a flow chart and write a python pseudo-code, program to print the following pattern. Get input as a number of rows from the user. Display "Invalid input" if the boundary condition fails.

1 2 3 4 5

2 2 3 4 5

3 3 3 4 5

4 4 4 4 5

5 5 5 5 5

```
number_of_rows = int(input("Enter the number of rows: "))  
  
for i in range(1, number_of_rows+1):  
    for j in range(i):  
        print(i, end="")  
    for k in range(i+1, number_of_rows+1):  
        print(k, end="")  
    print()
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python
2022/EX2P7.py"

Enter the number of rows: 5

12345

22345

33345

44445

55555

Process finished with exit code 0

Q.8 Develop an algorithm and write the Python code to print the following pattern. Check boundary conditions and print 'Invalid input' for wrong output.

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

Boundary Condition:

$n > 0$

```
number_of_rows = int(input("Enter the number of rows: "))  
if number_of_rows <= 0:  
    print("Invalid Input")  
else:  
    for i in range(number_of_rows):  
        for j in range(2 * i + 1):  
            print("*", end="")  
        print()
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python
2022/EX2P8.py"

Enter the number of rows: 4

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

Process finished with exit code 0

Q.9 Write a python code to find the factorial of the given number.

```
num = int(input("enter the number: "))
factorial = 1
for i in range(1, num + 1):
    factorial = factorial * i
print("the factorial of ", num, "is", factorial)
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX2P9.py"
enter the number: 6
the factorial of 6 is 720
```

Process finished with exit code 0

Q.10 Write a python code to check whether a given number of odd or even.

```
num = int(input("Enter the number: "))
if num % 2 == 0:
    print("number is even")
else:
    print("number is odd")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX2P10.py"
```

```
Enter the number: 69
```

```
number is odd
```

Process finished with exit code 0

Q.11 Write a python program to segregate student based on their CGPA as outstanding, excellent, good, average, better and poor.

```
num = int(input("Enter the CGPA: "))
if 9 <= num <= 10:
    print("outstanding")
elif 8 <= num <= 8.99:
    print("Excellent")
elif 7 <= num <= 7.99:
    print("good")
elif 6 <= num <= 6.99:
    print("Average")
elif 5 <= num <= 5.99:
    print("better")
elif 0 <= num <= 4.99:
    print("Poor")
print(num)
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX2P11.py"
```

```
Enter the CGPA: 6.9
```

```
Average
```

```
6.9
```

Process finished with exit code 0

Q.12 Write a python script that accepts 5 numbers form the user. It then prints one of the two things: if any of the values entered are duplicates, print DUPLICATES otherwise, it prints ALL UNIQUE.

```
n = int(input("Enter the number of terms: "))
a = []
for i in range(n):
    a.append(int(input("Enter the value: ")))

flag = False
for x in a:
    if a.count(x) > 1:
        flag = True

if flag:
    print("Duplicate")
else:
    print("Unique")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX2P12.py"

Enter the number of terms: 5

Enter the value: 10

Enter the value: 20

Enter the value: 30

Enter the value: 40

Enter the value: 10

Duplicate

Process finished with exit code 0

Exercise 3

1. Every day morning and evening milk is brought from 'n' farms to a milk booth for sales. Given the amount of milk from 'n' farms, write the python code to compute total quantity of milk in the booth. For example, if milk comes from 3 farms in quantities 2 litres 300ml, 3 litres 700ml and 4 litres 600ml then the total quantity of milk in booth is 10litre 600ml.

```
number_of_farms = int(input("Enter the number of farms: "))
farms = list()
for i in range(number_of_farms):
    print(f"enter the amount of milk collected from farm{i+1}")
    l = int(input("Liter: :"))
    ml = int(input("Ml: "))
    farms.append(l*1000 +ml)

total = sum(farms)
print(f"Total milk collected: {total//1000}Ltr {total%1000}Ml")
```

Output

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX3_P1.py"

Enter the number of farms: 3

enter the amount of milk collected from farm1

Liter: :2

Ml: 300

enter the amount of milk collected from farm2

Liter: :3

Ml: 700

enter the amount of milk collected from farm3

Liter: :4

Ml: 600

Total milk collected: 10Ltr 600Ml

Process finished with exit code 0

2. Write the python code to determine if two numbers are amicable. Two numbers 'm' and 'n' are said to be amicable if sum of proper factors (factors other than itself) of one number is equal to the other. For example the pair (220, 284) is called amicable pair. Since sum of proper factors of 220 - 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110 is equal to 284 and sum of proper factors of 284 - 1, 2, 4, 71 and 142 is equal to 220.

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
sum1 = 0
sum2 = 0

for i in range(1, num1):
    if num1 % i == 0:
        sum1 += i

for i in range(1, num2):
    if num2 % i == 0:
        sum2 += i

if sum1 == num2 and sum2 == num1:
    print(f"({num1}, {num2}) are amicable pair.")
else:
    print(f"({num1}, {num2}) are not amicable pair.")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX3_P2.py"

Enter the first number: 220

Enter the second number: 284

((220, 284)) are amicable pair.

Process finished with exit code 0

3. In a fun game, MXM grid is given with full of coins. The player has to give a number 'N' of his choice. If N is lesser than M then he doesn't gain any points. Otherwise he has to place all coins of MxM grid in the N X N grid and he gains points equal to the number of free cells in the N X N grid.

Input format:

Dimension of grid - M

Choice by the player - N

Output format:

number of points gained

```
m = int(input("Enter the value of M: "))
n = int(input("Enter the value of N: "))

if n < m:
    print("You don't get any point")
else:
    free_cells = n * n - m * m
    print(f"You've gained: {free_cells} points.")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX3_P3.py"

Enter the value of M: 4

Enter the value of N: 3

You don't get any point

Process finished with exit code 0

Exercise 4

Q.1 Sanjay has lot of Indian rupees coins in his pocket, He is interested in finding the number of unique coins in his pocket. Given the total number of coins and amount of each coin as an input write a python code to find the unique coins he is carrying in his pocket.

```
n = int(input("Enter he number of total coins: "))
l = list()
for i in range(n):
    l.append(input("Enter the amount of coin: "))

print(f"Wallet: {l}")
print(f"Unique Coins: {set(l)}")
print(f"Number of unique coins: {len(set(l))}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P1.py"

Enter he number of total coins: 5

Enter the amount of coin: 1

Enter the amount of coin: 2

Enter the amount of coin: 3

Enter the amount of coin: 4

Enter the amount of coin: 1

Wallet: ['1', '2', '3', '4', '1']

Unique Coins: {'3', '2', '1', '4'}

Number of unique coins: 4

Process finished with exit code 0

Q.2 A little girl is found of toys. she grouped the toys according to their similarities. For example she grouped them and arranged them in a list after assigning a count for each group. Write a python program to count even and odd numbers in a list.

```
l = []
even_count=0
odd_count=0
n = int(input("Enter the number of items in a list: "))
for i in range(n):
    l.append((int(input("Enter the value: "))))

for i in l:
    if i%2==0:
        even_count+=1
    else:
        odd_count+=1

print(f"Number of Even Numbers in the list: {even_count}")
print(f"Number of Odd Numbers in the list: {odd_count}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P2.py"

Enter the number of items in a list: 3

Enter the value: 12

Enter the value: 45

Enter the value: 78

Number of Even Numbers in the list: 2

Number of Odd Numbers in the list: 1

Process finished with exit code 0

Q.3 Given a Matrix, the task is to write python program to reverse every Kth row. Where, K is some span value. Input : test_list = [[5, 3, 2], [8, 6, 3], [3, 5, 2], [3, 6], [3, 7, 4], [2, 9]], K = 4 Output : [[5, 3, 2], [8, 6, 3], [3, 5, 2], [6, 3], [3, 7, 4], [2, 9]] Explanation : Every 4th row is reversed.

```
test_list = [[5, 3, 2], [8, 6, 3], [3, 5, 2], [3, 6], [3, 7, 4], [2, 9]]
print(f"test_list: {test_list}")
k = int(input("Enter the value of 'k': "))
for i in range(len(test_list)):
    if i == k-1:
        test_list[i].reverse()
print(f"Output: {test_list}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P3.py"

test_list: [[5, 3, 2], [8, 6, 3], [3, 5, 2], [3, 6], [3, 7, 4], [2, 9]]

Enter the value of 'k': 3

Output: [[5, 3, 2], [8, 6, 3], [2, 5, 3], [3, 6], [3, 7, 4], [2, 9]]

Process finished with exit code 0

Q.4 Given a list, the task is to write a python program that can construct n*m matrix. Input : test_list = [6, 3, 7, 2, 6, 8, 4, 3, 9, 2, 1, 3], n, m = 3, 5 Output : "Matrix Not Possible" Explanation : List has 12 elements and 3*5 is 15, hence Matrix not possible. Input : test_list = [6, 3, 7, 2, 6, 8], n, m = 2, 3 Output : [[6, 3, 7], [2, 6, 8]] Explanation : List converted to 2*3 matrix.

```
test_list = [6, 3, 7, 2, 6, 8, 4, 3, 9, 2, 1, 3]
print("The original list is : " + str(test_list))

n = int(input("Enter the value of 'n': "))
m = int(input("Enter the value of 'm': "))

k = 0
res = []
if n * m != len(test_list):
    res = "Matrix Not Possible"
else:
    for i in range(0, n):
        sub = []
        for j in range(0, m):
            sub.append(test_list[k])
            k += 1
        res.append(sub)

print("Constructed Matrix : " + str(res))
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P4.py"

The original list is : [6, 3, 7, 2, 6, 8, 4, 3, 9, 2, 1, 3]

Enter the value of 'n': 3

Enter the value of 'm': 4

Constructed Matrix : [[6, 3, 7, 2], [6, 8, 4, 3], [9, 2, 1, 3]]

Process finished with exit code 0

Q.5 Given a List, extract all elements whose frequency is greater than K. Input : test_list = [4, 6, 4, 3, 3, 4, 3, 4, 3, 8], K = 3 Output : [4, 3] Explanation : Both elements occur 4 times. Input : test_list = [4, 6, 4, 3, 3, 4, 3, 4, 6, 6], K = 2 Output : [4, 3, 6] Explanation : Occur 4, 3, and 3 times respectively.

```
test_list = [4, 6, 4, 3, 3, 4, 3, 4, 3, 8]
k = int(input("Enter the value of 'k': "))
op_list = list()
for i in test_list:
    if test_list.count(i) >= k:
        op_list.append(i)
print(f"{list(set(op_list))}")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P5.py"

Enter the value of 'k': 3

[3, 4]

Process finished with exit code 0

Q.6 Given a list from user with number of elements, write a python code to perform swapping of first and last numbers in a list.

```
a = []
n = int(input("Enter the number of elements in list:"))
for x in range(0, n):
    element = int(input("Enter element" + str(x + 1) + ":"))
    a.append(element)
temp = a[0]
a[0] = a[n - 1]
a[n - 1] = temp
print("New list is:")
print(a)
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX4_P6.py"

Enter the number of elements in list:3

Enter element1:10

Enter element2:20

Enter element3:30

New list is:

[30, 20, 10]

Process finished with exit code 0

Exercise 5

Q1. Given a position of coins of player1 and player2 in a 3X3 Tic Tac Toc board, write a program to determine if the board position is a solution and if so identify the winner of the game. In a Tic Tac Toc problem, if the coins in a row or column or along a diagonal is of the same player then he has won the game. Assume that player1 uses '1' as his coin and player2 uses '2' as his coin. '0' in the board represent empty cell.

```
def getResult(board):
    game_result = 0
    # checking diagonal1
    for i in range(len(board)):
        if board[i][i] == 1:
            return 1
        elif board[i][i] == 2:
            return 2

    # checking diagonal2
    for i in range(len(board)):
        if board[i][len(board) - 1 - i] == 1:
            return 1
        elif board[i][len(board) - 1 - i] == 2:
            return 2

    # checking row wise
    for i in range(len(board)):
        for j in range(len(board)):
            if board[i][j] == 1:
                return 1
            elif board[i][j] == 2:
                return 2

    # checking column wise
    for i in range(len(board)):
        for j in range(len(board)):
            if board[j][i] == 1:
                return 1
            elif board[j][i] == 2:
                return 2

    return game_result

n = 3
board = list()
for i in range(n):
    row_list = list()
    for j in range(n):
        x = int(input("Enter the value: "))
        row_list.append(x)
    board.append(row_list)

result = getResult(board)
if result == 1:
    print("Player1 Wins!!!!")
elif result == 2:
    print("Player2 Wins!!!!")
else:
    print("Draw Match")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX5P1.py"

"E:/Python

Enter the value: 1

Enter the value: 2

Enter the value: 1

Enter the value: 2

Enter the value: 1

Enter the value: 0

Enter the value: 1

Enter the value: 0

Enter the value: 2

Player1 Wins!!!!

Process finished with exit code 0

Q2. Given a size of square matrix and number of elements, find if it an identity matrix or not?

```
n = int(input("enter the number of rows and columns: "))
mat = list()

print("Enter the elements of the matrix row wise:")
for i in range(n):
    row_list = list()
    for j in range(n):
        x = int(input("Enter the value: "))
        row_list.append(x)
    mat.append(row_list)

print("The Matrix")
for i in range(n):
    print(mat[i])
    print()

count_1 = 0
count_0 = 0
for i in range(n):
    for j in range(n):
        if i == j and mat[i][j] == 1:
            count_1 = 1
        if i != j and mat[i][j] == 0:
            count_0 = 1

if count_1 == 1 and count_0 == 1:
    print("Identity Matrix")
else:
    print("Not an Identity Matrix")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX5P2.py"

"E:/Python

enter the number of rows and columns: 3
Enter the elements of the matrix row wise:
Enter the value: 1
Enter the value: 0
Enter the value: 0
Enter the value: 0
Enter the value: 1
Enter the value: 0
Enter the value: 0
Enter the value: 0
Enter the value: 1

The Matrix

[1, 0, 0]

[0, 1, 0]

[0, 0, 1]

Identity Matrix

Process finished with exit code 0

Q.3 Given the rows and columns of a two-dimensional matrix and number of elements, find the sum of its diagonals and the absolute difference between the two diagonals.

```
"""
Q.3 Given the rows and columns of a two-dimensional matrix and number of elements, find
the sum of
its diagonals and the absolute difference between the two diagonals.
"""

n = int(input("enter the number of rows and columns: "))
mat = list()

print("Enter the elements of the matrix row wise:")
for i in range(n):
    row_list = list()
    for j in range(n):
        x = int(input("Enter the value: "))
        row_list.append(x)
    mat.append(row_list)

print("The Matrix")
for i in range(n):
    print(mat[i])
    print()

d1_sum = 0
d2_sum = 0

for i in range(n):
    for j in range(n):
        if i + j == n - 1:
            d2_sum += mat[i][j]
        if i == j:
            d1_sum += mat[i][j]

print(f"Sum of diagonal1: {d1_sum}.")
print(f"Sum of diagonal2: {d2_sum}.")
print(f"Absolute difference between the two diagonals: {abs(d1_sum - d2_sum)}.")
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX5P3.py"

"E:/Python

enter the number of rows and columns: 3

Enter the elements of the matrix row wise:

Enter the value: 1

Enter the value: 2

Enter the value: 3

Enter the value: 4

Enter the value: 5

Enter the value: 6

Enter the value: 7

Enter the value: 8

Enter the value: 9

The Matrix

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

Sum of diagonal1: 15.

Sum of diagonal2: 15.

Absolute difference between the two diagonals: 0.

Process finished with exit code 0

Q.4 Write a python code to perform matrix multiplication.

```
"""
Q.4 Write a python code to perform matrix multiplication.
"""
import numpy as np

mat1 = list()
mat2 = list()
product_mat = list()

r1 = int(input("Enter the number of rows of first matrix: "))
c1 = int(input("Enter the number of columns of first matrix: "))
r2 = int(input("Enter the number of rows of second matrix: "))
c2 = int(input("Enter the number of columns of second matrix: "))

if r1 != c2:
    print("Matrix Multiplication isn't possible.")
else:
    print("Enter the elements of the 1st matrix row wise:")
    for i in range(r1):
        row_list = list()
        for j in range(c1):
            x = int(input("Enter the value: "))
            row_list.append(x)
        mat1.append(row_list)

    print("First Matrix")
    for i in range(r1):
        print(mat1[i])
        print()

    print("Enter the elements of the 2nd matrix row wise:")
    for i in range(r2):
        row_list = list()
        for j in range(c2):
            x = int(input("Enter the value: "))
            row_list.append(x)
        mat2.append(row_list)

    print("Second Matrix")
    for i in range(r2):
        print(mat2[i])
        print()

    product_mat = np.dot(mat1, mat2)

    print("Product Matrix")
    for i in range(r1):
```

```
print(product_mat[i])  
print()
```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX5P4.py"

"E:/Python

Enter the number of rows of first matrix: 2

Enter the number of columns of first matrix: 3

Enter the number of rows of second matrix: 3

Enter the number of columns of second matrix: 2

Enter the elements of the 1st matrix row wise:

Enter the value: 1

Enter the value: 2

Enter the value: 3

Enter the value: 4

Enter the value: 5

Enter the value: 6

First Matrix

[1, 2, 3]

[4, 5, 6]

Enter the elements of the 2nd matrix row wise:

Enter the value: 10

Enter the value: 11

Enter the value: 20

Enter the value: 21

Enter the value: 30

Enter the value: 31

Second Matrix

[10, 11]

[20, 21]

[30, 31]

Product Matrix

[140 146]

[320 335]

Process finished with exit code 0

Q.5 In a University final examination, students got marks in various subjects and based on the marks their grades are determined. The following rules are applied to receive the grades.

Marks (In between)	Grades
90 to 100	S
80 to 89	A
70 to 79	B
60 to 69	C
50 to 59	D
Less than 50	F

For the given number of subjects, get the marks of the student in each subject and count the number of grades. If student received at least two 'S' grades and two 'A' grades then print 'Eligible for scholarship' otherwise print 'Not eligible for scholarship'. Check boundary conditions and print 'Invalid input' for wrong input.

Input Format:

Number of subjects

Marks for each subject

Output Format:

Display the count of each grade

'Eligible for scholarship' OR 'Not eligible for scholarship'

Boundary Conditions:

Number of subjects ≥ 5

$101 > \text{Marks} > 0$

```

number_of_subjects = int(input("Enter the number of subjects: "))
marks = []

for i in range(number_of_subjects):
    x = int(input(f"Enter the marks for subject{i + 1}: "))
    marks.append(x)

s_count = 0
a_count = 0
b_count = 0
c_count = 0
d_count = 0
f_count = 0

for x in marks:
    if x > 100 or x < 0:
        print("Invalid Input")
    elif 90 <= x <= 100:
        s_count += 1
    elif 80 <= x < 90:
        a_count += 1
    elif 70 <= x < 80:
        b_count += 1
    elif 60 <= x < 70:
        c_count += 1
    elif 50 <= x < 60:
        d_count += 1
    else:
        f_count += 1

if s_count >= 2 and a_count >= 2:
    print("Eligible for Scholarship.")
else:
    print("Not Eligible for Scholarship")

```

Output:

C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
2022/EX5P5.py"

"E:/Python

Enter the number of subjects: 50
Enter the marks for subject1: 69
Enter the marks for subject2: 54
Enter the marks for subject3: 52
Enter the marks for subject4: 69
Enter the marks for subject5: 61
Not Eligible for Scholarship

Process finished with exit code 0