**DAY-3**

**PROGRAMS AND OUTPUTS**

**1.** **Write a program to print the following pattern**

**Sample Input: Enter the number to be printed: 121**

**Max Number of time printed: 3**

**121**

**121 121**

**121 121 121**

**121 121**

**121**

num = input("Enter the number to be printed: ")

max\_count = int(input("Enter the number of times to be printed: "))

for i in range(1, max\_count + 1):

    print((num + ' ') \* i)

for i in range(max\_count - 1, 0, -1):

    print((num + ' ') \* i)

**2. Write a program to print square star and rectangle dollar pattern? Get the number of rows and columns from the user.**

n=int(input("enter the number of rows and columns for a square:"))

for i in range(n):

    print("\*"\*n)

rows = int(input("enter the number of rows:"))

cols = int(input("enter the number of columns"))

for i in range(rows):

    print("$"\*cols)

**3. Write a program to print hollow square symbol pattern? Get the symbol and Square size as input from the user**

symbol=(input("enter the symbol that want to print:"))

size=int(input("enter the size of the square:"))

for i in range(size):

    if i==0 or i==size-1:

        print(symbol\*size)

4**. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is SecondIf aggregate is 60>= and = and Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is**

**Fail.**

**Sample Input & Output:**

**Enter the marks in python: 90**

**Enter the marks in c programming: 91**

**Enter the marks in Mathematics: 92**

**Enter the marks in Physics: 93**

**Total= 366**

**Aggregate = 91.5**

**DISTINCTION**

**Test cases:**

**1. 18, 76,93,65**

**2. 73,78,79,75**

**3. 98,106,120,95**

**4. 96,73, -85,95**

**5. 78,59.8,76,79**

    else:

        print(symbol+" "\*(size-2)+symbol)

p\_marks=float(input("enter the marks obtained in python:"))

c\_marks=float(input("enter the marks obtained in c-programming:"))

m\_marks=float(input("enter the marks obtained in maths:"))

ph\_marks=float(input("enter the marks obtained in physics:"))

if p\_marks<0 or c\_marks<0 or m\_marks<0 or ph\_marks<0 :

    print("invalid input")

else:

    total\_marks=p\_marks+c\_marks+m\_marks+ph\_marks

    aggregate=total\_marks/4

    print("total marks:",total\_marks)

    print("aggregate =",aggregate)

    if aggregate>75:

        print("DISTINCTION")

    elif 60<=aggregate<75:

        print("FIRST DIVISION")

    elif 50<=aggregate<60:

        print("SECOND DIVISION")

    elif 40<=aggregate<50:

        print("third division")

    else:

        print("fail")

**5. Write a program that would sort a list of names in alphabetical order Ascending or**

**Descending, choice get from the user?**

**Sample Input:**

**Banana**

**Carrot**

**Radish**

**Apple**

**Jack**

**Order(A/D) : A**

**Sample Output:**

**Apple**

**Banana**

**Carrot**

**Jack**

**Radish**

def sort\_names(names,order):

    if order.upper()=='A':

        return sorted(names)

    elif order.upper()=='D':

        return sorted(names,reverse=True)

    else:

        return "Invalid order"

names=input("enter a list of names spearated by spaces:").split()

order=input("enter the order(A/D):")

sorted\_names=sort\_names(names,order)

print(''.join(sorted\_names))

**6. Write a program to calculate tax given the following conditions:**

**a. If income is less than or equal to 1,50,000 then no tax[1,50,000 –Basic Slab]**

**b. If taxable income is 1,50,001 – 3,00,000 then charge 5% tax for basic slab then charge 10% tax for the balance amount**

**c. If taxable income is 3,00,001 – 5,00,000 then charge 5% tax for basic slab then charge 20% tax for the balance amount**

**d. If taxable income is above 5,00,001 then charge5% tax for basic slab then charge 30% tax for the balance amount**

**Sample Input:**

**Enter the income:200000**

**Sample Output:**

**Tax= 12500**

**Test cases: 1. 400700**

**2. 2789239**

**3. 150000**

**4. 00000**

**5. -125486**

def calculate\_tax(income):

    if income<150000:

        return 0

    elif income<=300000:

        return 0.05\*150000+0.1\*(income-150000)

    elif income<=500000:

        return 0.05\*150000+0.2\*(income-300000)

    else:

        return 0.05\*150000+0.3\*(income-500000)

income=int(input("enter the income:"))

tax=calculate\_tax(income)

print("tax:",tax)

**7. Write a program for matrix addition?**

**Sample Input:**

**Mat1 = 1 2**

**5 3**

**Mat2 = 2 3**

**4 1**

**Sample Output:**

**Mat Sum = 3 5**

**9 4**

Mat1 = [[1, 2], [5, 3]]

Mat2 = [[2, 3], [4, 1]]

result = [[0, 0], [0, 0]]

for i in range(len(Mat1)):

    for j in range(len(Mat1[0])):

        result[i][j] = Mat1[i][j] + Mat2[i][j]

print("Resultant Matrix:")

for row in result:

    print(row)

**8.** **Write a program to print rectangle symbol pattern. Get the symbol and rectangle size as input from user, get the choice from the user for the pattern hollow rectangle or full rectangle**.

def print\_rectangle(symbol, rows, cols, hollow):

    for i in range(rows):

        for j in range(cols):

            if hollow and (i != 0 and i != rows - 1) and (j != 0 and j != cols - 1):

                print(" ", end="")

            else:

                print(symbol, end="")

        print()

symbol = input("Enter the symbol: ")

rows = int(input("Enter the number of rows: "))

cols = int(input("Enter the number of columns: "))

choice = input("Enter your choice (H for Hollow, F for Full): ")

if choice.upper() == 'H':

    hollow = True

elif choice.upper() == 'F':

    hollow = False

else:

    print("Invalid input. Please enter H for Hollow or F for Full.")

    exit()

print\_rectangle(symbol, rows, cols, hollow)

**9** **. A Pythagorean triplet is a set of three integers a, b and c such that a2 + b2 = c 2 . Given a limit, generate all Pythagorean Triples with values smaller than given limit?**

def generate\_pythagoras\_triplets(limit):

    triplets=[]

    for a in range(1,limit):

        for b in range(a,limit):

            c=(a\*\*2+b\*\*2)\*\*0.5

            if c<limit and c==int(c):

                triplets.append((a,b,int(c)))

                return triplets

limit=int(input("enter the limit:"))

triplets=generate\_pythagoras\_triplets(limit)

for triplet in triplets:

    print(triplet)

**10. Write a program to print the first n perfect numbers. (Hint Perfect number means a**

**positive integer that is equal to the sum of its proper divisors)**

**Sample Input:**

**N = 3**

**Sample Output:**

**First 3 perfect numbers are: 6 , 28 , 496**

**Test Cases:**

**6. N = 0**

**7. N = 5**

**8. N = -2**

**9. N = -5**

**10. N = 0.2**

def is\_perfect(num):

    sum = 0

    for i in range(1, num):

        if num % i == 0:

            sum += i

    return sum == num

def print\_perfect\_numbers(n):

    if n <= 0:

        print("Input should be a positive integer.")

        return

    count = 0

    num = 1

    while count < n:

        if is\_perfect(num):

            print(num, end=" ")

            count += 1

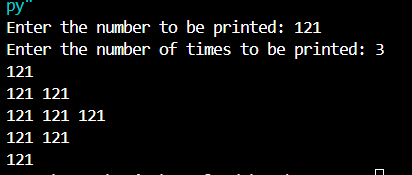
        num += 1

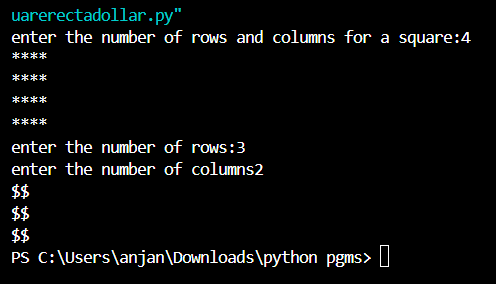
n = int(input("Enter the value of N: "))

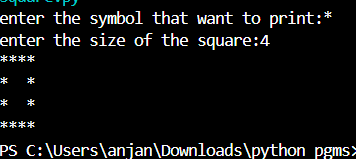
print("First", n, "perfect numbers are:", end=" ")

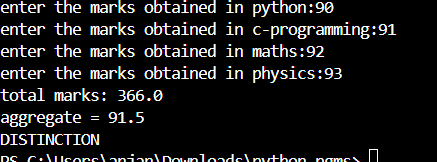
print\_perfect\_numbers(n)

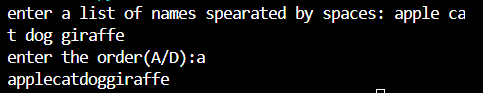
**OUTPUTS:**





****

****

****

