T1_class Program_Vishal Acharya

October 13, 2024

1 Write a python program that determines whether a student is eligible for PG course or not.To be eligible, the student must have obtained more than 80% in X and XII examination,and 70% plus marks in Graduation.If the student changes his stream (Science, Commerce, or Arts), then deduct 5% from his Graduation score.

[1]:

```
print("Welcome to PG eligibility checker")
a = int(input("Enter the percentage obtained in class X(Please enter without␣
 ↪'%' symbol): "))
b = int(input("Enter the percentage obtained in class XII(Please enter without␣
 ↪'%' symbol): "))
c = int(input("Enter the percentage obtained in class Graduation(Please enter␣
 ↪without '%' symbol): "))
d = input("Stream selected for Graduation: ")
e = input("Stream selecting for Post Graduation: ")
if a > 80 and b > 80:
    if  d != e:
        c = c-5
    if c > 70:
        print("Congratulation you are eligible to apply for PG in",e,".")
    else:
        print("Sorry you are not eligible to apply for PG.")
else:
    print("Sorry you are not eligible to apply for PG.")
```

```
Welcome to PG eligibility checker
Enter the percentage obtained in class X(Please enter without '%' symbol): 85
Enter the percentage obtained in class XII(Please enter without '%' symbol): 87
Enter the percentage obtained in class Graduation(Please enter without '%'
symbol): 59
Stream selected for Graduation: Arts
Stream selecting for Post Graduation: Arts
Sorry you are not eligible to apply for PG.
```

2 write a python program to calculate electricity bill based on following instructions

- 0 to 150 Units - Rs. 3 per unit
- 151-300 units - Rs.100 plus 3.75 per unit exceeding 150 units
- 301 to 450 units - Rs. 250 plus 4 per unit exceeding 300 units
- 451-600 units - Rs. 300 plus 4.25 per unit exceeding 450 units
- Above 600 - Rs.400 plus Rs. 5 per unit exceeding 600 units” “ ”

[6]:

```
unit=int(input("Enter units: "))

if unit<0:
    print("Enter correct value")
elif unit<=150:
    bill=unit*3
elif unit>150 and unit<=300:
    bill=(unit*3)+(100+(unit-150)*3.75)
elif unit>300 and unit<=450:
    bill=(unit*3)+(100+(unit-150)*3.75)+(250+(unit-300)*4)
elif unit>450 and unit<=600:
    bill=(unit*3)+(100+(unit-150)*3.75)+(250+(unit-300)*4)+(300+(unit-450)*4.25)
else:
    bill=(unit*3)+(100+(unit-150)*3.75)+(250+(unit-300)*4)+(300+(unit-450)*4.
  ↪25)+(400+(unit-600)*5)
print(bill,"Rs.")
```

```
Enter units: 487
4280.0 Rs.
```

3 Write a program to calculate parking fees. take input of entry time in hours and minute and exit time in hours and minutes. Parking rate for 3 hours will be: for Truck/bus-20Rs.

[46]:

```
vehicle=int(input("Enter 1--for Truck/bus, 2--for car, 3--for cycle/motorcycle/
  ↪scooter: "))
enter_hr=int(input("enter hour in 12hr format: "))
enter_min=int(input("Enter min: "))
enter_am=input("Enter AM or PM: ")
exit_hr=int(input("exit hour in 12 hr format: "))
exit_min=int(input("Exit min: "))
exit_am=input("Enter AM or PM: ")
if enter_am!='AM' or enter_am!='PM' or exit_am!='AM' or exit_an!='PM':
    print("Enter AM or PM")
else:
    if enter_am=='PM' and enter_hr !=12:
        enter_hr +=12
```

```python
    if exit_am=='PM' and exit_hr !=12:
        exit_hr +=12
    if enter_am=='AM' and enter_hr ==12:
        enter_hr==0
    if exit_am=='AM' and exit_hr==12:
        exit_hr==0
    if exit_hr==12 and enter_hr==12 and (exit_am != enter_am):
        exit_hr+=12
    if exit_hr<enter_hr:
        exit_hr +=24
    parking_time_hr=(exit_hr-enter_hr)
    parking_time_min=(exit_min-enter_min)

    if parking_time_hr<3 or (parking_time_hr==3 and parking_time_min<=0):
        if vehicle==1:
            rate=20
        elif vehicle==2:
            rate=10
        elif vehicle==3:
            rate=5
    elif parking_time_hr>3 or (parking_time_hr==3 and parking_time_min>0):
        if vehicle==1:
            rate=30
        elif vehicle==2:
            rate=20
        elif vehicle==3:
            rate=10
    print(f"Your parking charge is {rate}Rs.")
```

```
Enter 1--for Truck/bus, 2--for car, 3--for cycle/motorcycle/scooter: 1
enter hour in 12hr format: 12
Enter min: 0
Enter AM or PM: AM
exit hour in 12 hr format: 12
Exit min: 0
Enter AM or PM: PM
Your parking charge is 30Rs.
```

4 Problem Definition: Parking Fee Calculation Based on Vehicle Type and Stay Duration

- You are tasked with creating a program that calculates parking fees based on the type of vehicle and the duration of time it stays parked. The parking charges vary depending on whether the vehicle is petrol (PV), diesel (DV), or electric (EV). The user will provide the entry and exit times in hours and minutes, specifying whether the time is AM or PM. The system should then calculate the total hours of stay and determine the parking fee accordingly.
- Problem Breakdown:

1. Input Details:

- The type of vehicle (PV, DV, or EV).
- Entry time (hours, minutes, AM/PM).
- Exit time (hours, minutes, AM/PM).
- Time Calculation: Convert the 12-hour format (AM/PM) to a continuous time format.

2. Calculate the total stay in hours.

- Rounding Logic: If the stay duration is a fractional value (e.g., 3.5 hours), round up to the next whole number for billing purposes.

3. Fee Calculation Based on Vehicle Type:

- Petrol Vehicle (PV): If the stay is less than or equal to 4 hours, the fee is 100 INR per hour. For more than 4 hours, the first 4 hours cost 400 INR, and each additional hour costs 150 INR.
- Diesel Vehicle (DV): If the stay is less than or equal to 4 hours, the fee is 150 INR per hour. For more than 4 hours, the first 4 hours cost 600 INR, and each additional hour costs 200 INR.
- Electric Vehicle (EV): If the stay is less than or equal to 4 hours, the fee is 20 INR per hour. For more than 4 hours, the first 4 hours cost 80 INR, and each additional hour costs 200 INR.

4. Output:

- The total stay duration.
- The net chargeable hours (rounded appropriately).
- The total parking fee based on the vehicle type and stay duration.

5. Key Constraints:

- Hours should be between 0 and 12.
- Minutes should be between 0 and 59.
- The program should handle incorrect inputs and prompt the user for valid values.

[13]:

```
vehicle = input("PV, DV, EV :\n ")

while True:

    inHrs = int(input('entry hrs:\n'))
    inMin = int(input('entry min:\n'))
    iam_pm = input("entry am or pm\n")
    oHrs = int(input('out hrs:\n'))
    oMin = int(input('out min:\n'))
    oam_pm = input("out am or pm:\n")
    if inHrs >12 or inHrs <0 or oHrs >12 or oHrs < 0 or inMin <0 or inMin >= 60␣
 ↪or oMin < 0 or oMin >= 60 :
        print()
        print("enter Hour values between 0 and 12:\n")
        print("enter Min values between 0 and 60:\n")
    else:
        break
```

```python
if iam_pm== 'am':
    itime = inHrs + (inMin/60)
elif iam_pm== 'pm':
    itime =  inHrs + (inMin/60)
if oam_pm == 'am':
    otime = 12+oHrs + (oMin/60)
elif oam_pm == 'pm':
    otime = 12 + (oHrs + (oMin/60))
citystay = otime - itime

print("stay in Manali:", citystay, 'Hrs')

if citystay >0 and citystay <12 and (type(citystay) == float):
    chargehrs = int(citystay) + 1
else:
    chargehrs = int(citystay)

print('Net Chargeble Hours: ', chargehrs)

if vehicle == 'PV': # Petrol vehicle
    if chargehrs <=4:
        pay = chargehrs*100
    elif chargehrs >4:
        pay = 400 + chargehrs*150
if vehicle == 'DV': # Diesel Vehicle
    if chargehrs <=4:
        pay = chargehrs*150
    elif chargehrs >4:
        pay = 600 + chargehrs*200
if vehicle == 'EV': # Electric Vehicle
    if chargehrs <=4:
        pay = chargehrs*20
    elif chargehrs >4:
        pay = 80 + chargehrs*200

print(f'The net payment for {citystay} hrs in Manali is INR {pay}')
```

```
PV, DV, EV :
 PV
entry hrs:
12
entry min:
21
entry am or pm
am
out hrs:
7
out min:
```

```
15
out am or pm:
pm
stay in Manali: 6.9 Hrs
Net Chargeble Hours:  7
The net payment for 6.9 hrs in Manali is INR 1450
```

5 Calculate Maximum Even and Minimum Odd Numbers from User Entered Numbers “Write a Python program that prompts the user to enter numbers and stops only when the user enters “done”. After this, print the average of the numbers and the minimum odd number and the maximum even number from among all the numbers entered by the user.

- Note: You are not allowed to use any built-in structures like lists, tuples, etc. or any built-in functions like max, min, etc.

[13]:

```
count=sum=0
count_e=count_o=0
while(True):
 n=input("Enter Number or 'done' to stop")
 if n=='done':
     break
 n=int(n)
 count+=1
 if n%2==0 and count_e==0:
     max_e=n
     count_e+=1
 if n%2!=0 and count_o==0:
     min_o=n
     count_o+=1
 sum += n
 if n%2==0 and n>max_e:
     max_e=n
 if n%2!=0 and n<min_o:
     min_o=n
print("Average:",sum/count)
print("Minimum Odd Number:",min_o)
print("Maximum Even Number",max_e)
```

```
Enter Number or 'done' to stop5
Enter Number or 'done' to stop-3
Enter Number or 'done' to stop9
Enter Number or 'done' to stop8
Enter Number or 'done' to stop-2
Enter Number or 'done' to stopdone
```

```
Average: 3.4
Minimum Odd Number: -3
Maximum Even Number 8
```

6 Find Sum of digits of a postitve integer until sum is in single digit

[14]:

```
n=int(input("Enter any positive integer: "))
sum=0
while (n!=0 or sum>9):
    sum=sum+n%10
    if n==0:
        n=sum
        sum=0
    else:
        n=n//10
print(sum)
```

```
Enter any positive integer: 15235
7
```

7 butterfly pattern

[15]:

```
for i in range(5):
    for j in range(i+1):
        print('*',end=" ")
    for l in range(5,i+1,-1):
        print(" "*3,end=" ")
    for k in range(i+1):
        print('*',end=" ")
    print()
for i in range(4,0,-1):
    for j in range(i):
        print('*',end=" ")
    for l in range(5,i,-1):
        print(" "*3,end=" ")
    for k in range(i):
        print('*',end=" ")
    print()
```

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

[16]:

```
# Number of rows
N = 5

# Variables to store number of spaces and stars
spaces = 2 * N - 1
stars = 0

# The outer loop will run for (2 * N - 1) times
for i in range(1, 2 * N):
        # Upper half of the butterfly
        if i <= N:
                spaces = spaces - 2
                stars += 1
        # Lower half of the butterfly
        else:
                spaces = spaces + 2
                stars -= 1

        # Print stars
        for j in range(1, stars + 1):
                print("*", end="")

        # Print spaces
        for j in range(1, spaces + 1):
                print(" ", end="")

        # Print stars
        for j in range(1, stars + 1):
                if j != N:
                        print("*", end="")

        print() # Move to the next line
```

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

[17]:

```
for i in range (5):
     for j in range(5,i,-1):
        print('*',end=" ")
    for l in range(i):
        print(" "*3,end=" ")
    for k in range(5,i,-1):
        print('*',end=" ")
    print()
for i in range (5):
    for j in range(i+1):
        print('*',end=" ")
    for l in range(5,i+1,-1):
        print(" "*3,end=" ")
    for k in range(i+1):
        print('*',end=" ")
    print()
```

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

[21]:

```
userInput = int(input("Please input side length of diamond: "))
if userInput > 0:
    for i in range(userInput):
        for s in range (userInput - i) :
            print(" ", end="")
        for j in range((i * 2) - 1):
            print("*", end="")
        print()
    for i in range(userInput, 0, -1):
        for s in range (userInput - i) :
            print(" ", end="")
        for j in range((i * 2) - 1):
            print("*", end="")
        print()
```

```
Please input side length of diamond: 8

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

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

8 Write a program that asks the user for an hour between 1 and 12, asks them to enter am or pm, and asks them how many hours into the future they want to go. Print out what the hour will be that many hours into the future, printing am or pm as approp

- An example is shown below.
- Enter hour: 8
- am (1) or pm (2)? 1
- How many hours ahead? 5
- New hour: 1 pm” “ ”

[8]:

```python
user = int(input('enter hour :'))
day = input('enter am or pm:')
hour = int(input('How many hours ahead ?:'))
if user<0 or user>12 or (day!='am' and day!='pm'):
     print("Enter correct time")
else:
     if day == 'pm' and user!=12:
         user += 12;
     elif day == 'am' and user == 12:
         user = 0;
     new_hour = user + hour;
     new_hour = new_hour % 24;
     if new_hour >= 0 and new_hour <12:
         if new_hour == 0:
             new_hour = 12;
         print(f'New hour : {new_hour}', end='am')
         print()
     else:
         new_hour = new_hour % 12;
         if new_hour == 0:
             new_hour = 12 ;
         print(f'New hour : {new_hour}', end='pm')
         print()
```

```
enter hour :12
enter am or pm:pm
How many hours ahead ?:84
New hour : 12am
```

9 Python Program to Find All Pythagorean Triplets in the Range pythagorean triplets are the three numbers which follows pythagoros therom, a^{2+b}2=cˆ2 –>then a,b,c is triplets.

[4]:

```
limit=int(input("Enter upper limit:"))
c=0
m=2
while(c<limit):
    for n in range(1,m+1):
        a=m*m-n*n
        b=2*m*n
        c=m*m+n*n
        if(c>limit):
            break
        if(a==0 or b==0 or c==0):
            break
        print(a,b,c)
    m=m+1
```

```
Enter upper limit:20
3 4 5
8 6 10
5 12 13
15 8 17
12 16 20
```

[6]:

```
n= int(input("enter row:"))
for i in range(1,n+1):
     for k in range(i):
        print(" ",end=" ")
     for j in range(i,n+1):
        print(" ",j,end=" ")
     print()
for i in range(2,n+1):
     for k in range(n+1,i,-1):
         print(" ",end=" ")
     for j in range(n,n-i,-1):
         print(" ",j,end=" ")
     print()
```

```
enter row:10
    1   2   3   4   5   6   7   8   9   10
```

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

10 find first max and second max

[7]:

```
count=0
f_h=-1
s_h=-1
while(True):
     n=input("Enter marks or done")

     if n=='done':
         break

     n=int(n)
     if n<0 or n>100:
         continue
     count+=1
     if n>f_h:
         s_h=f_h
         f_h=n
     elif n>s_h and n!=f_h:
         s_h=n

print("1st Maximum marks",f_h)
print("2nd Maximum marks",s_h)
```

```
Enter marks or done10
Enter marks or done8
Enter marks or done9
Enter marks or done25
Enter marks or done35
Enter marks or done45
Enter marks or done84
Enter marks or done75
Enter marks or done32
Enter marks or done47
Enter marks or done94
Enter marks or done12
Enter marks or donedone
1st Maximum marks 94
2nd Maximum marks 84
```

[8]:

```python
# Initialize variables to store the first and second lowest numbers
first_lowest = None
second_lowest = None
# Initialize a variable to keep track of the number of numbers entered
num_numbers = 0
# Prompt the user to enter numbers until they enter a non-integer value
while(True):
     number=input("Enter marks or done")

     if number=='done':
         break

     number=int(number)
     if number<0 or number>100:
         continue
     num_numbers += 1
     if first_lowest is None or number < first_lowest:
         second_lowest = first_lowest
         first_lowest = number
     elif second_lowest is None or number < second_lowest:
         second_lowest = number
if num_numbers < 2:
     print("Please enter at least two numbers to find the first and second␣
↪lowest.")
else:
     print("The first lowest number is:", first_lowest)
     print("The second lowest number is:", second_lowest)
```

```
Enter marks or done4
Enter marks or done6
Enter marks or done8
Enter marks or done10
Enter marks or done12
Enter marks or done58
Enter marks or done1
Enter marks or done3
Enter marks or done5
Enter marks or done2
Enter marks or done7
Enter marks or donedone
The first lowest number is: 1
The second lowest number is: 2
```

11 (Hardy-Ramanujan number

[9]:

```
n = 1729  # Number we are checking (Hardy-Ramanujan number)

count = 0  # Counter to keep track of distinct pairs
a1 = b1 = a2 = b2 = 0  # Variables to store the two distinct pairs

# First for loop to iterate over 'a'
for a in range(1, int(n ** (1/3)) + 1):
    # Second for loop to iterate over 'b' (b starts from a to avoid duplicates)
    for b in range(a, int(n ** (1/3)) + 1):
        if a**3 + b**3 == n:
            count += 1
            if count == 1:  # Store the first pair (a1, b1)
                a1 = a
                b1 = b
            elif count == 2:  # Store the second pair (a2, b2)
                a2 = a
                b2 = b

# Output the result if two distinct pairs are found
if count == 2:
    print(f"{n} is a Hardy-Ramanujan number. It can be expressed as the sum of␣
 ↪cubes:")
    print(f"{a1}^3 + {b1}^3")
    print(f"{a2}^3 + {b2}^3")
else:
    print(f"{n} is not a Hardy-Ramanujan number.")
```

```
1729 is a Hardy-Ramanujan number. It can be expressed as the sum of cubes:
1^3 + 12^3
9^3 + 10^3
```

[11]:

```
for n in range(1,5000):#Number we are checking (Hardy-Ramanujan number)

    count = 0  # Counter to keep track of distinct pairs
    a1 = b1 = a2 = b2 = 0  # Variables to store the two distinct pairs

    # First for loop to iterate over 'a'
    for a in range(1, int(n ** (1/3)) + 1):
            # Second for loop to iterate over 'b' (b starts from a to avoid␣
    ↪duplicates)
            for b in range(a, int(n ** (1/3)) + 1):
                if a**3 + b**3 == n:
                    count += 1
                    if count == 1:  # Store the first pair (a1, b1)
                        a1 = a
                        b1 = b
                    elif count == 2:  # Store the second pair (a2, b2)
                        a2 = a
                        b2 = b

        # Output the result if two distinct pairs are found
        if count == 2:
            print(f"{n} is a Hardy-Ramanujan number. It can be expressed as the sum␣
    ↪of cubes:")
            print(f"{a1}^3 + {b1}^3")
            print(f"{a2}^3 + {b2}^3")
```

```
1729 is a Hardy-Ramanujan number. It can be expressed as the sum of cubes:
1^3 + 12^3
9^3 + 10^3
4104 is a Hardy-Ramanujan number. It can be expressed as the sum of cubes:
2^3 + 16^3
9^3 + 15^3
```

12 Python Program to Test Collatz Conjecture for a Given Numberdef collatz(n): The Collatz conjecture is a conjecture that a particular sequence always reaches 1. The sequence is defined as: start with a number n. The next number in the sequence is n/2 if n is even and 3n + 1 if n is odd

[12]:

```
def collatz(n):
    # Print the starting number
    print(f"Starting number: {n}")

    # Continue until n becomes 1
    while n != 1:
        if n % 2 == 0:
            # If n is even, divide by 2
            n = n // 2
        else:
            # If n is odd, apply 3n + 1
            n = 3 * n + 1
        # Print the current value of n after each step
        print(n)

# Test the Collatz Conjecture for a given number
num = int(input("Enter a positive integer: "))
collatz(num)
```

```
Enter a positive integer: 10
Starting number: 10
5
16
8
4
2
1
```

13 add money in your pigi nank. start from monday with rs.1. increase rs.1 every day. next monday start with rs. 2 ...follow same as...calculate moiney after 30 days

[15]:

```
sum=0
d=int(input("enter day: "))
m=1
sun=7
s=d//7
#d=d%7
for i in range(s):
     for j in range(m,sun+1):
         sum+=j
     m+=1
     sun+=1
for k in range(d%7):
     sum=sum+m
     m=m+1
print(sum)
```

```
enter day: 30
165
```

14 A Disarium number is a number that is equal to the sum of its digits raised to the power of their respective positions.

[19]:

```python
# Function to calculate the number of digits in the number
def calculate_length(n):
    length = 0
    while n > 0:
        length += 1
        n = n // 10
    return length

# Function to check if the number is a Disarium number
def is_disarium(num):
    temp = num
    length = calculate_length(temp)
    sum_of_digits = 0

    # Calculating the sum of digits raised to the power of their positions
    while temp > 0:
        digit = temp % 10
        sum_of_digits += digit ** length
        length -= 1
        temp = temp // 10

    # Check if the sum of digits is equal to the original number
    return sum_of_digits == num

# Input from the user
num = int(input("Enter a number: "))

# Check if the number is a Disarium number
if is_disarium(num):
    print(f"{num} is a Disarium number.")
else:
    print(f"{num} is not a Disarium number.")
```

```
Enter a number: 89
89 is a Disarium number.
```

15 The problem is to find the smallest positive integer that satisfies the following conditions:

- Leftmost Digit Manipulation:
- Given a positive integer num, extract its leftmost digit.Form a new number by appending this leftmost digit to the right end of the original number and removing it from the leftmost position.

- Comparison with Original Number: The new number obtained after the leftmost digit manipulation should be 3.5 times larger than the original number.
- The goal of the program is to iteratively check integers starting from 1, modify each integer according to the described digit manipulation, and check if the resulting number is 3.5 times the original number. The smallest integer that satisfies this condition is then returned.

[20]:

```
def find_smallest_integer():
    num = 1  # Start from the smallest positive integer
    while True:
        # Calculate the number of digits in the current number
        num_digits = 0
        temp_num = num
        while temp_num > 0:
            temp_num //= 10
            num_digits += 1

        # Extract leftmost digit and calculate the new number
        divisor = 10 ** (num_digits - 1)
        leftmost_digit = num // divisor
        new_num = (num * 10) + leftmost_digit - (leftmost_digit * divisor * 10)

        # Check if the new number is 3.5 times larger than the original number
        if new_num == 3.5 * num:
            return num  # Return the smallest integer satisfying the property

        num += 1  # Move to the next integer

# Execute the function and print the result
result = find_smallest_integer()
print("The smallest positive integer satisfying the property is:", result)
```

```
The smallest positive integer satisfying the property is: 153846
```

16 Python program to print all happy numbers between 1 and 100

[21]:

```
for i in range(1,101):
     sum=0
     temp=i
     while (i>0 or sum>9):
         sum=sum+(i%10)**2
         if i==0:
             i=sum
             sum=0
         i//=10
     if sum==1:
         print(temp,end=" ")
```

```
1 4 7 10 13 14 23 26 31 32 33 36 38 40 41 45 54 57 59 62 63 66 68 70 75 79 83 86
89 95 97 98 100
```

17 HCF

[2]:

```
x=12
y=8
if x > y:
     smaller = y
else:
     smaller = x
for i in range(1,smaller + 1):
     if((x % i == 0) and (y % i == 0)):
         hcf = i
print(hcf)
```

```
4
```

18 LCM

[3]:

```
x=12
y=8
if x > y:
     greater = x
else:
     greater = y
while(True):
     if((greater % x == 0) and (greater % y == 0)):
         lcm = greater
         break
     greater += 1
print(lcm)
```

```
24
```

19 WAP to calculateSimple and Compound Interest

[23]:

```
principal = float(input('Enter amount: '))
time = float(input('Enter time: '))
rate = float(input('Enter rate: '))

simple_interest = (principal*time*rate)/100
compound_interest = principal * ( (1+rate/100)**time - 1)

print('Simple interest is: %f' % (simple_interest))
print('Compound interest is: %f' %(compound_interest))
```

```
Enter amount: 153200
Enter time: 2
Enter rate: 12
Simple interest is: 36768.000000
Compound interest is: 38974.080000
```

20 Program to convert seconds into hour:min:seconds format.

[24]:
```python
sec = int(input("Enter seconds: "))
hour = sec // 3600
sec %= 3600
min = sec // 60
sec %= 60
print("seconds value in hours:",hour)
print("seconds value in minutes:",min)
print("Time is: ",hour,":",min,":",sec)
```

```
Enter seconds: 7365
seconds value in hours: 2
seconds value in minutes: 2
Time is:  2 : 2 : 45
```

21 Python Program to Read a Number n and Compute n+nn+nnn

[9]:
```python
n=int(input("Enter a number n: "))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
comp=n+int(t1)+int(t2)
print("The value is:",comp)
```

```
Enter a number n: 15
The value is: 153045
```

22 Python Program to Find the Roots of a Quadratic Equation

[25]:
```python
print("Equation: ax^2 + bx + c ")
a=int(input("Enter a: "))
b=int(input("Enter b: "))
c=int(input("Enter c: "))
d=b**2-4*a*c
d1=d**0.5
if(d<0):
    print("The roots are imaginary. ")
else:
    r1=(-b+d1)/2*a
    r2=(-b-d1)/2*a
    print("The first root: ",round(r1,2))
    print("The second root: ",round(r2,2))
```

```
Equation: ax^2 + bx + c
Enter a: 6
Enter b: 8
Enter c: 2
The first root:  -12.0
The second root:  -36.0
```

23 Take three variables and get their values from the user. Without using any sorting functions or Python builtin data structures (lists, tuples, dictionaries, etc.), sort these numbers in ascending and descending orders

[22]:

```
a=int(input("Enter 1st No.: "))
b=int(input("Enter 2nd No.: "))
c=int(input("Enter 3rd No.: "))

if a<b and a<c:
    if b<c:
        x,y,z=a,b,c
    else:
        x,y,z=a,c,b
elif b<a and b<c:
    if a<c:
        x,y,z=b,a,c
    else:
        x,y,z=b,c,a
elif c<a and c<b:
    if b<a:
        x,y,z=c,b,a
    else:
        x,y,z=c,a,b
else:
    print("Enter correct input")

print("Ascending Order:",x,y,z)
print("Descending Order:",z,y,x)
```

```
Enter 1st No.: 15
Enter 2nd No.: 2
Enter 3rd No.: -23
Ascending Order: -23 2 15
```

```
Descending Order: 15 2 -23
```

24 Calculate the number of basic American coins given a value less than 1 dollar. A penny is worth 1 cent, a nickel is worth 5 cents, a dime is worth 10 cents, and aquarter is worth 25 cents. It takes 100 cents to make 1 dollar. So given an amount less than 1 dollar (if using floats, convert to integers for this exercise), calculate the number of each type of coin necessary to achieve the amount, maximizing the number of larger denomination coins. For example, given $ 0.76, or 76 cents, the correct output would be "3 quarters and 1 penny." Output such as "76 pennies" and "2 quarters, 2 dimes, 1 nickel, and 1 penny" are not acceptable.

[4]:

```python
dollar=float(input("Enter value less than 1 dollar: "))
if 0<dollar<1:
    cent=int(dollar*100)
    if cent>=25:
        quarter=cent//25
        cent %= 25
        print(quarter,"quarters",end=",")
    if cent>=10:
        dime=cent//10
        cent %=10
        print(dime,"dimes",end=",")
    if cent>=5:
        nickel=cent//5
        cent %=5
        print(nickel,"nickels",end=",")
    if cent>=1:
        penny=cent
        print(penny,"pennies")
else:
    print("Enter correct value")
```

```
Enter value less than 1 dollar: 0.97
3 quarters,2 dimes,2 pennies
```

25 Write a Python program that prompts the user to enter numbers and stops only when the user enters “done”. After this, print the average of the numbers and the minimum number and the maximum number fromamong all the numbers entered by the user. Note: You are not allowed to use any built-in structures like lists, tuples, etc. or any built-in functions like max, min, etc.

- E.g., if the user enters: 3, 4, 1, 6 ,“done”
- The program should print the
- Average: 3.5
- Minimum Number: 1
- Maximum Number: 6

[5]:

```python
count=sum=0
while(True):
    n=input("Enter Number or 'done' to stop")
    if n=='done':
        break
    n=int(n)
    count+=1
    if count==1:
        min=max=n
    sum += n
    if n>max:
        max=n
    if n<min:
        min=n
print("Average:",sum/count)
print("Minimum Number:",min)
print("Maximum Number",max)
```

```
Enter Number or 'done' to stop2
Enter Number or 'done' to stop3
Enter Number or 'done' to stop5
Enter Number or 'done' to stop-3
Enter Number or 'done' to stop0
Enter Number or 'done' to stop5
Enter Number or 'done' to stopdone
Average: 2.0
Minimum Number: -3
Maximum Number 5
```

26 WAP to find prime numbers from a given range

[6]:

```
for i in range(2,100):
    flag=0
    for j in range(2,i):
        if i%j==0:
            flag=1
    if flag==0:
        print(i,end=" ")
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

27 WAP to find prime factors of a given number by user

[7]:

```
n=int(input("enter no.: "))
for i in range(2,n+1):
    if n%i==0:
        flag=0
    for j in range(2,i):
        if i%j==0:
            flag=1
    if flag==0:
        print(i,end=" ")
```

```
enter no.: 60
2 3 5
```

28 WAP to check if a number is armstrong or not. No. is armstrong if 123..n=1^{n+2}n+3ˆn....

[2]:

```
n=int(input("enter no.: "))
temp=n
count=0
while n != 0:
    n //=10
    count+=1
n=temp
sum=0
while n !=0:
    r=n%10
    sum += r**count
    n //=10
if sum==temp:
    print("Armstrong Number")
else:
    print("Not Armstrong Number")
```

```
enter no.: 153
Armstrong Number
```

29 WAP to print armstrong no. in given range. No. is armstrong if 123..n=1^{n+2}n+3ˆn....

[3]:

```
for n in range(1,10000):
    temp=n
    count=0
    while n != 0:
        n //=10
        count+=1
    n=temp
    sum=0
    while n !=0:
        r=n%10
        sum += r**count
        n //=10
    if sum==temp:
        print(temp,end=" ")
```

```
1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474
```

30 Python Program to Find Numbers which are Divisible by 7 and Multiple of 5 in a Given Range

[6]:

```
lower=int(input("Enter the lower range:"))
upper=int(input("Enter the upper range:"))
for i in range (lower,upper+1):
    if(i%7==0 and i%5==0):
        print(i,end=" ")
```

```
Enter the lower range:1
Enter the upper range:1000
35 70 105 140 175 210 245 280 315 350 385 420 455 490 525 560 595 630 665 700
735 770 805 840 875 910 945 980
```

31 Python Program to Reverse a Number

[7]:

```
#n=int(input("Enter number: "))
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
```

```
print("Reverse of the number:",rev)
```

```
Enter number: 45321
Reverse of the number: 12354
```

32 Python Program to Check if a Number is a Palindrome

[8]:

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

```
Enter number:23432
The number is a palindrome!
```

33 Python Program to Check Whether a Given Number is Perfect Number if sum of divisiors of the number=the number, it is known as perfect number for ex. 6= 1+2+3, so 6 is perfect number

[18]:

```
n = int(input("Enter any number: "))
sum1 = 0
for i in range(1, n):
    if(n % i == 0):
        sum1 = sum1 + i
if (sum1 == n):
    print("The number is a Perfect number!")
else:
    print("The number is not a Perfect number!")
```

```
Enter any number: 6
The number is a Perfect number!
```

34 Python Program to Check if a Number is a Strong Number if 123..n=1!+2!+3!+...+n!, then it is strong number

[24]:

```
sum1=0
num=int(input("Enter a number:"))
temp=num
while(num):
    i=1
    f=1
    r=num%10
    while(i<=r):
        f=f*i
        i=i+1
    sum1=sum1+f
    num=num//10
if(sum1==temp):
    print("The number is a strong number")
else:
    print("The number is not a strong number")
```

```
Enter a number:145
The number is a strong number
```

35 Take three variables and get their values from the user. Without using any sorting functions or Python builtin data structures lists, tuples, dictionaries, etc.), sort these numbers in ascending and descending orders.

[4]:

```
a=int(input("Enter 1st No.: "))
b=int(input("Enter 2nd No.: "))
c=int(input("Enter 3rd No.: "))
if a<b and a<c:
     if b<c:
         x,y,z=a,b,c
     else:
         x,y,z=a,c,b
elif b<a and b<c:
     if a<c:
         x,y,z=b,a,c
     else:
         x,y,z=b,c,a
elif c<a and c<b:
     if b<a:
         x,y,z=c,b,a
     else:
         x,y,z=c,a,b
```

```python
else:
 print("Enter correct input")
print("Ascending Order:",x,y,z)
print("Descending Order:",z,y,x)
```

```
Enter 1st No.: 5
Enter 2nd No.: 7
Enter 3rd No.: 4
Ascending Order: 4 5 7
Descending Order: 7 5 4
```

36 Python Program to Check If Two Numbers are Amicable Numbers or Not Find the sum of the proper divisors of both the numbers. Check if the sum of the proper divisors is equal to the opposite numbers.If they are equal, they are amicable numbers.

[20]:

```python
x=int(input('Enter number 1: '))
y=int(input('Enter number 2: '))
sum1=0
sum2=0
for i in range(1,x):
    if x%i==0:
        sum1+=i
for j in range(1,y):
    if y%j==0:
        sum2+=j
if(sum1==y and sum2==x):
    print('Amicable!')
else:
    print('Not Amicable!')
```

```
Enter number 1: 284
Enter number 2: 220
Amicable!
```

1st Method (with this method you can change any number with any number in digit.)

[11]:

```python
#Take a number as user input. replace digit 0 with digit 1 in that number and print new number
#for ex. 2030 should print 2131.

i=int(input("Enter No.: "))
rev=0
updated_num=0
while i>0:
    r=i%10
    if r == 0:
        r=1
    rev=rev*10+r
    i=i//10
while rev>0:
    rem=rev%10
    updated_num=updated_num*10+rem
    rev=rev//10
print(updated_num)
```

```
Enter No.: 30502
31512
```

2nd Method

[13]:

```
#Take a number as user input. replace digit 0 with digit 1 in that number and print new number
#for ex. 2030 should print 2131.

n=int(input("Enter No.: "))
temp=n
count=0
while n != 0:
    r=n%10
    if r==0:
        temp=temp+1*(10**count) #you can change any number with 0 by changing 1(with whatever number) in this line
    count+=1
    n //=10
print(temp)
```

```
Enter No.: 30502
31512
```

37 Same type of program (just replace 0 with 5)

[14]:

```
#Take a number as user input. replace digit 0 with digit 5 in that number and print new number
#for ex. 2030 should print 2131.

n=int(input("Enter No.: "))
temp=n
count=0
while n != 0:
    r=n%10
    if r==0:
        temp=temp+5*(10**count) #you can change any number with 0 by changing 5(with whatever number) in this line
    count+=1
    n //=10
print(temp)
```

```
Enter No.: 30105
35155
```

38 A jar of Halloween candy contains an unknown amount of candy and if you can guess exactly how much candy is in the bowl, then you win all the candy. You ask the person in charge the following: If the candy is divided evenly among 5 people, how many pieces would be left over? The answer is 2 pieces. You then ask about dividing the candy evenly among 6 people, and the amount left over is 3 pieces. Finally, you ask about dividing the candy evenly among 7 people, and the amount left over is 2 pieces. By looking at the bowl, you can tell that there are less than 200 pieces. Write a program to determine how many pieces are in the bowl.

[16]:

```
x=200
for i in range(x):
    if i % 5 == 2 and i % 6 == 3 and i % 7 == 2:
                print(i, 'candies are in the bowl!')
```

```
177 candies are in the bowl!
```

[27]:

```
#Python Program to Find the Sum of the Series 1/1!+1/2!+1/3!+…1/N!

n=int(input("Enter the number of terms: "))
sum1=0
fact=1
for i in range(1,n+1):
    fact=fact*i
    sum1=sum1+(1/fact)
print("The sum of series is",round(sum1,2))
```

```
Enter the number of terms: 3
The sum of series is 1.67
```

39 Python Program to Find the Sum of the Series: 1 + x^2/2 + x^3/3 + ... x^n/n

[28]:

```
n=int(input("Enter the number of terms:"))
x=int(input("Enter the value of x:"))
sum1=1
for i in range(2,n+1):
    sum1=sum1+((x**i)/i)
print("The sum of series is",round(sum1,2))
```

```
Enter the number of terms:6
Enter the value of x:3
The sum of series is 204.85
```

40 Python Program to Find the LCM of Two Numbers

[29]:

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
if(a>b):
    min1=a
else:
    min1=b
while(1):
    if(min1%a==0 and min1%b==0):
        print("LCM is:",min1)
        break
    min1=min1+1
```

```
Enter the first number:20
Enter the second number:30
LCM is: 60
```

41 Write a program to read a date from user and print the date of a next day. Also check if the entered date is correct or not.

[17]:

```
day=int(input("Enter date: "))
month=int(input("Enter month: "))
year=int(input("Enter year: "))

if month==12 and day==31:
    month=1
    year+=1
    day=1
elif (month==1 or month==3 or month==5 or month==7 or month==8 or month==10)␣
 ↪and day==31:
    month+=1
    day=1
elif (month==4 or month==6 or month==9 or month==11) and day==30:
    month+=1
    day=1
elif month==2 and (day==28 or day==29):
    if ((year % 400 == 0) or (year % 100 != 0) and (year % 4 == 0)):
        if day==28:
            day+=1
        else:
            day=1
            month+=1
    elif day==29:
        day+=1
    else:
        day=1
        month+=1
else:
    day+=1
if 1<=day<=31 and 1<=month<=12 and month!=2:
    print(day,"/",month,"/",year)
elif month==2 and 1<=day<=29:
    print(day,"/",month,"/",year)
else:
    print("enter correct date")
```

```
Enter date: 28
Enter month: 2
Enter year: 2022
1 / 3 / 2022
```

42 reverse triangle pattern

[26]:

```
# Initialize start and stop values
rows = int(input("Enter the number of rows: "))
start = 1
stop = 2
currentNumber = stop
# Generate the pattern
for row in range(2, rows+2):
    for col in range(start, stop):
        currentNumber -= 1
        print(currentNumber, end=' ')
    print("") # Move to the next line
    start = stop
    stop += row
    currentNumber = stop
```

```
Enter the number of rows: 5
1
3 2
6 5 4
10 9 8 7
15 14 13 12 11
```

43 Python Program to Find All Pythagorean Triplets in the Range pythagorean triplets are the three numbers which follows pythagoros therom, a^{2+b}2=cˆ2 –>then a,b,c is triplets.

[22]:

```
limit=int(input("Enter upper limit:"))
c=0
m=2
while(c<limit):
    for n in range(1,m+1):
        a=m*m-n*n
        b=2*m*n
        c=m*m+n*n
        if(c>limit):
            break
        if(a==0 or b==0 or c==0):
            break
        print(a,b,c)
    m=m+1
```

```
Enter upper limit:10
3 4 5
8 6 10
```

44 WAP print identity matrix

[21]:

```
n=int(input("Enter a number: "))
for i in range(0,n):
    for j in range(0,n):
        if(i==j):
            print("1",end=" ")
        else:
            print("0",end=" ")
    print()
```

```
Enter a number: 4
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

[2]:
```
for row in range(0,7):
    for column in range(0,7):
        if (((column == 1 or column == 5) and row != 0) or ((row == 0 or row ==␣
 ↪3) and (column > 1 and column < 5))):
            print("*",end="")
        else:
            print(" ",end="")
    print()
```

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

[4]:
```
for row in range(7):
    for col in range(5):
        if (col==0) or (col==4 and (row!=0 and row!=3 and row!=6)) or ((row==0␣
 ↪or row==3 or row==6) and (col>0 and col<4)):
              print("*",end=" ")
        else:
            print(" ", end=" ")
    print()
```

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

[8]:
```
userInput = int(input("Please input side length of diamond: "))

if userInput > 0:
    for i in range(userInput):
        for s in range (userInput - i) :
            print(" ", end="")
        for j in range((i * 2) - 1):
            print("*", end="")
        print()
    for i in range(userInput, 0, -1):
        for s in range (userInput - i) :
            print(" ", end="")
        for j in range((i * 2) - 1):
            print("*", end="")
        print()
```

```
Please input side length of diamond: 5

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

[21]:

```
for i in range(6):
    for j in range(i,6):
        print(" ",end="")
    for k in range(0,i+1):
        print("*",end=" ")
    print()
for i in range(7,0,-1):
    for j in range(i,7):
        print(" ",end="")
    for k in range(0,i):
        print("*",end=" ")
    print()
```

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

45 WAP swap first and last digit

[31]:

```
n=int(input("Enter a num: "))
count=0
temp=n
while n!=0:
    r=n%10
    if count==0:
        lastdigit=r
    count+=1
    n//=10
firstdigit=r
diff=firstdigit-lastdigit
temp=temp+diff-diff*(10**(count-1))
print(temp)
```

```
Enter a num: 598954613
398954615
```

46 Problem Definition: Test Score Analysis Program You are tasked with writing a program that processes the test scores of 10 students, analyzing various aspects such as the highest score, second highest score, lowest score, and average score. The program also checks for invalid inputs where any score is greater than 100, flagging such cases.

[6]:

```
sum=0
count=0
highest=0
lowest=100
while count<10:
    n=int(input("Enter Test Score: "))
    if n>100:
        print("Entered score is more than hundered, so enter again")
        continue
    count+=1
    sum=sum+n

    if n>highest:
        second_largest=highest
        highest=n
    elif n>second_largest and n!=highest:
        second_largest=n

    if n<lowest:
        second_lowest=lowest
        lowest=n
    elif n<second_lowest:
        second_lowest=n

new_sum=sum-(second_lowest+lowest)
```

```
print("Highest Number is:",highest)
print("Lowest Number is:",lowest)
print("Avergae Test Score is:",sum/count)
print("Second Largest Number is:",second_largest)
print("Average after dropping the two lowest scores:",new_sum/(count-2))
```

```
Enter Test Score: 45
Enter Test Score: 42
Enter Test Score: 10
Enter Test Score: 9
Enter Test Score: 8
Enter Test Score: 55
Enter Test Score: 74
Enter Test Score: 77
Enter Test Score: 85
Enter Test Score: 95
Highest Number is: 95
Lowest Number is: 8
Avergae Test Score is: 50.0
Second Largest Number is: 85
Average after dropping the two lowest scores: 60.375
```

47 understand for loop

[7]:

```
x = 2
for i in range(x):
    x -= 2
    print (x)
```

```
0
-2
```

[25]:

```
for i in range(1,5):
    print(i,"hello")
    i+=3
    print(i,"welcome")
else:
    print(i,"last")
```

```
1 hello
4 welcome
2 hello
5 welcome
3 hello
6 welcome
4 hello
7 welcome
7 last
```

[8]:

```
var = 10
for i in range(5):
    for j in range(2, 3, 1):
        if var%2 == 0:
            break
        var += 1
    var+=1
else:
    var+=1
print(var)
```

```
20
```

[9]:

```
A=70
if A>90:
    print('Grade A')
elif A>70 and A<90:
    print('Grade B')
elif A>50 and A<70:
    print('Grade C')
elif A>35 and A<50:
    print('Grade D')
else:
    print('Fail')
```

```
Fail
```

[10]:

```
seconds=3650
if seconds>=3600:
    hour=seconds//3600
    seconds %= 3600
    print(hour,"hours",end=" ")
if seconds>=60:
    minute=seconds//60
    seconds %= 60
    print(minute,"minutes",end=" ")
if seconds>0:
    print(seconds,"seconds")
```

```
1 hours 50 seconds
```

[11]:

```
A=0
for i in range(4):
    if i%2==0:
        pass
    else:
        continue
        break
    A+=1
```

```
print(A)
```

```
2
```

[12]:

```
A=0
for i in range(4):
    if i%2==0:
        pass
    else:
        continue
        break
    A+=1
print(A)
```

```
2
```

[ ]: