23. Write a program to print n prime numbers after n^{th} Prime number

Sample Input:
N = 3
Sample Output:
3^{rd} Prime number is 5
3 prime numbers after 5 are: 7, 11, 13
Test cases:

1. N = P
2. N = 0
3. N = -4
4. N = 11
5. N = 7.2

```
# Prompt the user for input

N = int(input("N: "))



# Find the nth prime number

count = 0

number = 2



while count < N:

    is_prime = True

    for i in range(2, int(number**0.5) + 1):

        if number % i == 0:

            is_prime = False

            break

    if is_prime:

        count += 1

    number += 1
```

```
nth_prime = number - 1

# Find the n prime numbers after the nth prime number
prime_numbers = []
count = 0
number = nth_prime + 1

while count < N:
    is_prime = True
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            is_prime = False
            break
    if is_prime:
        prime_numbers.append(number)
        count += 1
    number += 1

# Print the results
print(N, "prime number is", nth_prime)
print(N, "prime numbers after", nth_prime, "are:", ', '.join(str(num) for num in prime_numbers))
```

24. Write a program to find the prime number in the array of numbers

Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output:

Prime numbers in Array elements = {23, 19}

Test cases:

1.Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4.Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

```
# Prompt the user for input
elements = input("Array of elements (comma-separated): ").split(", ")
array = [int(num) for num in elements]

# Find the prime numbers in the array
prime_numbers = []
for num in array:
    if num < 2:
        continue
    is_prime = True
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            is_prime = False
            break
    if is_prime:
        prime_numbers.append(num)

# Print the result
print("Prime numbers in Array elements =", prime_numbers)
```

25. Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

Test cases:

1. A = 11, B = 11

2. A = 20, B = 10

3. A = 0, B = 0

4. A = -5, B = 5

5. A = 7, B = -12

```
# Prompt the user for input
A = int(input("A: "))
B = int(input("B: "))

# Initialize a list to store the composite numbers
composite_numbers = []

# Iterate through the range from A to B (inclusive)
for num in range(A, B+1):
    if num > 1:
        for i in range(2, num):
            if num % i == 0:
                composite_numbers.append(num)
                break

# Print the composite numbers
print(", ".join(str(num) for num in composite_numbers))
```

26. Write a program to find the number of composite numbers in an array of elements

Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Number of Composite Numbers = 5
Test cases:
1.Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

```
# Prompt the user for input
elements = input("Array of elements (comma-separated): 
").split(", ")
array = [int(num) for num in elements]
```

```
# Count the number of composite numbers
composite_count = 0
for num in array:
    if num < 2:
        continue
    is_composite = False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            is_composite = True
            break
    if is_composite:
        composite_count += 1

# Print the result
print("Number of Composite Numbers =", composite_count)
```

33. Write a program to reverse an array

Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Reverse Array elements = {19, 21 23, 16, 27, 18, 16}
Test cases:
1.Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

```
# Prompt the user for input
elements = input("Array of elements (comma-separated): ").split(", ")
array = [int(num) for num in elements]

# Reverse the array
reversed_array = array[::-1]

# Print the reversed array
print("Reverse Array elements =", reversed_array)
```

34. Write a program to find the given number is Harshad number or not .

Note: Harshad number means **an integer that is divisible by the sum of its digits when written in that base**

Sample Input:

Enter the number : 21

Sample Output:

Given number is Harshad number

Test cases:

1. 6804
2. 378
3. 111
4. 0
5. 145.678

```
# Prompt the user for input

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


# Calculate the sum of the digits

digit_sum = sum(int(digit) for digit in str(number))


# Check if the number is divisible by the sum of its digits

if number % digit_sum == 0:

   print("Given number is a Harshad number")

else:

   print("Given number is not a Harshad number")
```

35. Write a program to print the given number even or odd

Sample Input:

Enter the number : 6561

Sample Output:

The given number is odd

Test cases:

1. 0
2. -1254

3. A144
4. 145.23
5. 23.456

36. Write a program to print the number of Odd numbers and number of even numbers in between M and N?

Sample Input:
M = 60
N = 300
Sample Output:

```
Number of Odd Numbers = 120
Number of Even Numbers = 119
```

Test cases:
1. M = 100, N = 100
2. M = 500, N = 100
3. M = -5, N = 4
4. M = A, N = 6
5. M = 12, N = -12

```
# Prompt the user for input
M = int(input("M: "))
N = int(input("N: "))

# Initialize variables
odd_count = 0
even_count = 0

# Iterate through the range from M to N (inclusive)
for num in range(M, N+1):
    if num % 2 != 0:  # odd number
        odd_count += 1
    else:  # even number
        even_count += 1
```

```
# Print the results

print("Number of Odd Numbers =", odd_count)

print("Number of Even Numbers =", even_count)
```

37. Write a program to print the all Odd numbers and number of even numbers in between M and N?

Sample Input:
M = 6
N = 15

Sample Output:

```
All Odd Numbers = 7,9,11,13
All Even Numbers = 8,10,12,14
```

Test cases:
1. M = 100, N = 100
2. M = 500, N = 100
3. M = -5, N = 4
4. M = 72, N = -72
5. M = 0, N = 0

```
# Prompt the user for input
M = int(input("M: "))
N = int(input("N: "))

# Initialize variables
odd_numbers = []
even_count = 0

# Iterate through the range from M to N (inclusive)
for num in range(M, N+1):
   # Check if the number is odd or even
   if num % 2 != 0:  # odd number
      odd_numbers.append(num)
   else:  # even number
      even_count += 1

# Print the odd numbers and even count
print("All Odd Numbers =", ', '.join(str(num) for num in odd_numbers))
print("All Even Numbers =", ', '.join(str(num) for num in range(M+1,
N) if num % 2 == 0))
```

38. Find the nth odd number after n odd number

Sample Input:
N : 4
Sample Output:
4th Odd number after 4 odd numbers = 15
Test cases:
1. N = 0
2. N = -6
3. N = 2021
4. N = -14.5
5. N = -196

39. Write a program to find the Mean of first ‘N odd numbers, even numbers, square numbers and cube numbers (using switch case)

Sample Input:
Enter N value : 5
Case: 2
Sample Output:
Mean of first 5 even numbers: 6
Test cases:
1. N = 16
2. N = -8
3. N = 0
4. N = -10.01
5. N = 11.22

```
# Prompt the user for input
N = int(input("Enter N value: "))
case = int(input("Case: "))

# Define the variables to store the sum and count
sum_of_numbers = 0
count = 0

# Calculate the sum based on the case
if case == 1:  # Odd numbers
    for i in range(1, N*2, 2):
        sum_of_numbers += i
        count += 1
elif case == 2:  # Even numbers
    for i in range(2, N*2 + 1, 2):
        sum_of_numbers += i
        count += 1
elif case == 3:  # Square numbers
    for i in range(1, N+1):
        sum_of_numbers += i**2
        count += 1
elif case == 4:  # Cube numbers
    for i in range(1, N+1):
        sum_of_numbers += i**3
        count += 1
else:
    print("Invalid case number")

# Calculate the mean
mean = sum_of_numbers / count

# Print the result
if case == 1:
    print("Mean of first", N, "odd numbers:", mean)
elif case == 2:
    print("Mean of first", N, "even numbers:", mean)
elif case == 3:
    print("Mean of first", N, "square numbers:", mean)
elif case == 4:
    print("Mean of first", N, "cube numbers:", mean)
```

41. Find the LCM and GCD of n numbers?

Sample Input:
N value = 2
Number 1 = 16
Number 2 = 20
Sample Output:
LCM = 80
GCD = 4

Test cases:

1. N = 3, {12, 25, 30}

2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}
4. N = -2, {52, 60}
5. N = 2, {30, 45}

```
n=int(input("enter the n:"))
l=[]
for i in range(n):
    num=int(input("enter n1: "))
    l.append(num)
gcd=l[0]
for i in range(1,len(l)):
    a=gcd
    b=l[i]
    while b!=0:
        a,b=b,a%b
    gcd=a
lcm=l[0]
for i in range(1,len(l)):
    a=lcm
    b=l[i]
    lcm=(a*b)//gcd
print(gcd)
print(lcm)
```

42. Write a program to Print M multiples of N number

Sample Input:
M = 6
N = 3
Sample Output:
6 multiples of 3: 3, 6, 9, 12, 15, 18
Test cases:
1. M = 0, N = 5
2. M = 5, N = 0
3. M = -5, N = 4

4. M = A, N = 10
5. M = 3, N = P

write the pyhton program to print sum of series 1!/1+2!/2+3!/3+4!/

```
n = 5 # Number of terms in the series

series_sum = 0

for i in range(1, n + 1):

    factorial = 1

    for j in range(1, i + 1):

        factorial *= j

    term = factorial / i

    series_sum += term

print("Sum of the series:", series_sum)
```

You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Test Case: 1.Input: n = 2 Output: 2

2.Input: n = 3 Output: 3 3.Input: n = 4 Output: 5 4.Input: n = 1 Output: 1 5.Input: n = 5 Output: 8

```
n = 5  # Number of steps

if n <= 1:

    distinct_ways = 1

else:

    distinct_ways = [0] * (n + 1)

    distinct_ways[1] = 1

    distinct_ways[2] = 2

    for i in range(3, n + 1):

        distinct_ways[i] = distinct_ways[i - 1] + distinct_ways[i - 2]


print("Number of distinct ways to climb the staircase:", distinct_ways[n])
```

pattern:

```
2

4 4

16 16 16

256 256 256 256

65536 65536 65536 65536 65536
```

```
n=int(input("enter :"))

num=2

for i in range(n):

    for j in range(i+1):

        print(num,end=" ")

    print()

    num*=num
```

2.Write a program to print the following pattern.

```
    0.1
   0.10.2
  0.10.20.3
 0.10.20.30.4
0.10.20.30.40.5
```

```
i=int(input("enter the no.of rows"))

for i in range(1, i+1):
    r=0.1
    for j in range(1, i + 1):

        print("%.1f"%r,end='')
        r=r+0.1
    print()
```

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

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

for i in range(n):

    for j in range(n):
        if i == 0 or i == n - 1 or j == 0 or j == n - 1:
            print("*", end=" ")
        else:
            print(" ", end=" ")
    print()
```

16. Write a program to print rectangle star pattern.

```
n= int(input("Enter the number of rows: "))
for i in range(n):
    for j in range(n):
        print("$",end=" ")
    print()
```

17. Write a program to print the following pattern.

```
    1
   12
  123
 1234
12345
```

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

18. Write a program to print hollow Square Dollar pattern and full rectangle Star Pattern?

```
n= int(input("Enter the number of rows: "))
for i in range(n):
    for j in range(n):
        if i == 0 or i == n - 1 or j == 0 or j == n - 1:
            print("$", end=" ")
        else:
            print(" ", end=" ")
    print()
```

2.

```
n= int(input("Enter the number of rows: "))
n1= int(input("Enter the number of columns: "))
for i in range(n):
    for j in range(n1):
      print("$",end=" ")
    print()
```

16.Write a program to print the below pattern

```
1

2 2

3  3  3

4  4   4   4

3  3  3

2  2

 1
```

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

for i in range(1,n+1):
    for j in range(1,i+1):
        print(i,end=" ")
    print()
for i in range(n-1,-1,-1):
    for j in range(1,i+1):
        print(i,end=" ")
    print()
```

2. Write a program to print the below pattern

```
1

2 2

3  3  3

4    4   4   4
5    5    5   5   5
```

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

for i in range(1,n+1):
    for j in range(1,i+1):
        print(i,end=" ")
    print()
```

15. Write a program to print the below pattern

```
0.1
0.04   0.09
0.16   0.25     0.36
0.49 0.64     0.81     1.00
```

```
n= int(input("Enter the number of rows:"))
c=0.1
for i in range(1, n + 1):
    for j in range(1, i + 1):
        print("{:.2f}".format(c** 2), end="\t")
        c=c+0.1
    print()
```

3. Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: ^

Max Number of time printed: 3

```
^

^ ^

^ ^ ^
```

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


for i in range(1,n+1):

   for j in range(1,i+1):

      print("^",end="")

   print()
```

4. Write a program to print the below pattern?

```
            1

        1       1

    1       2       1

1       3       3       1
```

```
                1    4    6    4    1
```

```
num_rows = int(input("Enter the number of rows: "))
for i in range(num_rows):
    print(" " * (num_rows - i - 1), end="")
    num = 1
    for j in range(i + 1):
        print(num, end=" ")
        num = int(num * (i - j) / (j + 1))
    print()
```

1. Write a program to print symbol pattern?
   Get the symbol from user and choices from the user.

Choices:

Pattern Type: Hollow or Full

Pattern Size: Square or Rectangle

```
# Prompt the user for symbol and pattern choices
symbol = input("Enter the symbol to be printed: ")
pattern_type = input("Enter the pattern type (Hollow or Full): ").lower()
pattern_size = input("Enter the pattern size (Square or Rectangle): ").lower()

# Determine the number of rows and columns based on pattern size
if pattern_size == "square":
    rows = 5
    cols = 5
elif pattern_size == "rectangle":
    rows = 5
    cols = 10

# Generate and print the pattern
for i in range(rows):
    for j in range(cols):
        if pattern_type == "full" or (pattern_type == "hollow" and (i == 0 or i
    == rows - 1 or j == 0 or j == cols - 1)):
            print(symbol, end="")
        else:
            print(" ", end="")
    print()
```

5. Write a program to print the Inverted Full Pyramid pattern?

```
# Prompt the user for the number of rows

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


# Iterate through each row in reverse order

for i in range(num_rows, 0, -1):

    # Print spaces before the stars

    print(" " * (num_rows - i), end="")


    # Print the stars for the current row

    print("*" * (2 * i - 1))
```

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

121121

121121121

121121

121
```

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

max_prints = int(input("Max Number of times printed: "))

for i in range(1, max_prints + 1):
    print(num * i)

  for i in range(max_prints - 1, 0, -1):

    print(num * i)
```

a program to accept a sentence and print only the first letter of each word of the sentence in capital letters separated by a full stop.

```
sentence = input("Enter a sentence: ")


# Split the sentence into words

words = sentence.split()


# Extract the first letter of each word and capitalize it

capitalized_letters = [word[0].upper() for word in words]


# Join the capitalized letters and add a full stop at the end

result = '.'.join(capitalized_letters) + '.'
```

1. Write a program to reverse a word using loop?(Not to use inbuilt functions)Sample Input:
String: TEMPLE
Sample Output:
Reverse String: ELPMET
   Test cases:
   1. SIGN UP
   2. AT-LEAST3.
   1245
   4. !@#$%
   5. 145*999=144855

Answer:

```
string = input("Enter a string: ")
reversed_string = ""

for i in range(len(string)-1, -1, -1):
reversed_string += string[i]

print("Reversed string:", reversed_string)
```

2. Write a program to calculate Pow(x,n), Add(x,n), Sub(x,n), Mul(x,n), Div(x,n)? Get the input and choice from the user.

Sample Input:
X = 2
N = 4
Choice : 2
Sample Output:
Add(X,N) = 6

Test cases:

1. X = 0 , N = 4
2. X = 5 , N = 0
3. X = -3 , N = 3
4. X = 0 , N = 0
5. X = 123, N = 123

```
choice = int(input("Enter your choice (1-5): "))
```

```
if choice in range(1, 6):
    x = float(input("Enter the value of x: "))
    n = float(input("Enter the value of n: "))

    if choice == 1:
        result = 1
        for _ in range(n):
            result *= x
        print(f"Result: {x} raised to the power of {n} is {result}")
    elif choice == 2:
        result = x + n
        print(f"Result: {x} + {n} = {result}")
    elif choice == 3:
        result = x - n
        print(f"Result: {x} - {n} = {result}")
    elif choice == 4:
        result = x * n
        print(f"Result: {x} * {n} = {result}")
    elif choice == 5:
        if n != 0:
            result = x / n
            print(f"Result: {x} / {n} = {result}")
        else:
            print("Error: Division by zero is not allowed!")
else:
    print("Invalid choice! Please enter a valid option (1-5).")
```

3. Write a program to count all the prime and composite numbers entered by the user.Sample Input:

Enter the numbers

4

54

29

71

7

59

98

23

Sample Output:

```
Composite number:3

Prime number:5
```

Test cases:

1. 33, 41, 52, 61,73,90
2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE
3. 45, 87, 09, 5.0 ,2.3, 0.4
4. -54, -76, -97, -23, -33, -98
5. 45, 73, 00, 50, 67, 44

```
composite_count = 0

prime_count = 0


n = int(input("Enter the number of values: "))


for a in range(n):

  num = int(input("Enter a number: "))


  is_prime = True

  if num < 2:

    is_prime = False

  else:

    for i in range(2, int(num ** 0.5) + 1):

      if num % i == 0:
```

```
                is_prime = False

                break


    if is_prime:

        prime_count += 1

    else:

        composite_count += 1


print("Composite  number:",  composite_count)

print("Prime number:", prime_count)
```

4. Find the Mth maximum number and Nth minimum number in an array and then find thesum of it and difference of it.

Sample Input:
Array of elements = {14, 16, 87, 36, 25, 89, 34}
M = 1
N = 3
Sample Output:
1st Maximum Number = 89
3rd Minimum Number = 25
Sum = 114
Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1
2. {0, 0, 0, 0}, M = 1, N = 2
3. {-12, -78, -35, -42, -85}, M = 3 , N = 3
4. {15, 19, 34, 56, 12}, M = 6 , N = 3
5. {85, 45, 65, 75, 95}, M = 5 , N = 7

```
# Input array

arr = input("Enter the array elements (space-separated): ")

arr1=arr.split()

arr1 = [int(num) for num in arr1]


# Input values for M and N

M = int(input("Enter the value of M: "))

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

```
# Sort the array in ascending and descending order
sorted_arr_asc = sorted(arr1)
sorted_arr_desc = sorted(arr1, reverse=True)

# Find the Mth maximum and Nth minimum numbers
mth_max = sorted_arr_desc[M - 1]
nth_min = sorted_arr_asc[N - 1]

# Calculate the sum and difference
sum_result = mth_max + nth_min
diff_result = abs(mth_max - nth_min)

# Print the results
print(f"{M}th Maximum Number:", mth_max)
print(f"{N}th Minimum Number:", nth_min)
print("Sum:", sum_result)
print("Difference:", diff_result)
```

5.Write a program to print the total amount available in the ATM machine with the conditionsapplied.

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

Sample Input:

```
Enter the 1st Denomination: 500
Enter the 1st Denomination number of notes: 4
Enter the 2nd Denomination: 100
Enter the 2nd  Denomination number of notes: 20
Enter the 3rd  Denomination: 200
Enter the 3rd  Denomination number of notes: 32
Enter the 4th Denomination: 2000
Enter the 4th Denomination number of notes: 1
```

Sample Output:

```
Total Available Balance in ATM: 12400
```

Test Cases:

3 Hidden Test cases (Think Accordingly based on Denominations)

```
# Initialize the denomination values
denominations = [2000, 500, 200, 100]

# Initialize the total balance
total_balance = 0

# Get the denomination priority and number of notes from the user
for i in range(len(denominations)):
    denomination = int(input(f"Enter the {i+1}st Denomination: "))
    num_notes = int(input(f"Enter the {i+1}st Denomination number of notes: "))

    # Calculate the total amount for the denomination and add it to the total balance
    amount = denomination * num_notes
    total_balance += amount

# Print the total available balance
print("Total Available Balance in ATM:", total_balance)
```

6. Write a program to print the following

```
         pattern.1
        12
       123
      1234
     12345
```

7. Write a program to print hollow Square Dollar

```
pattern?length=int(input("Enter the side of the square
:"))
for i in range(length):
    for j in range(length):
        if(i == 0 or i == length-1 or j == 0 or j == length-1):
            print("$",end=" ")
        else:
          print(" ",end= " ")
```

```
   print()
```

8. 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
   b. If taxable income is 1,50,001 – 3,00,000 the charge 10% tax
   c. If taxable income is 3,00,001 – 5,00,000 the charge 20% tax
   d. If taxable income is above 5,00,001 then charge 30% tax

Sample Input:

Enter the income:200000

Sample Output:

Tax= 20000

Test cases:

1. 400700
2. 2789239
3. 150000
4. 00000
5. -125486

```
try:
   income=float(input("Enter the income:"))
   if income<=150000:
         tax=0
   elif 150001 <= income <= 300000:
         tax=(income-150000)*10/100
   elif 300001 <= income <= 500000:
         tax=150000*10/100+(income-300000)*20/100
   else:
         tax=150000*10/100+200000*20/100+(income-500000)*30/100
   print(tax)
except ValueError:
      print("String is not allowed please enter valid data")
```

9. Program to remove duplicates from the sorted array

Sample Input:
Array = {15, 14, 25, 14, 32, 14, 31}
Sample Output:
Sorted Array = {14, 15, 25, 31, 32}
Test cases:

1. {16, 16, 16 16, 16}
2. {0, 0, 0, 0}
3. {-12, -78, -35, -42}
4. {1,2,3,7,8,9,4,5,6}
5. {1-2,2-3,3-4,4-5,5-6}

```
my_list = []

n=int(input("enter the number of elements:"))

for i in range(1,n+1):

        n1=int(input("Enter the elememts:"))

        my_list.append(n1)

print("List Before ", my_list)

temp_list = []


for i in my_list:

   if i not in temp_list:

      temp_list.append(i)

my_list = temp_list

my_list.sort()

print("List After removing duplicates ", my_list)
```

10. Python Program to create a list of all numbers in a range which are perfect squares and thesum of the digits of the number is less than 10.

Sample Input & Output:

```
Enter lower range: 1

Enter upper range: 40

[1, 4, 9, 16, 25, 36]
```

Test case:

```
1.Enter lower range: 50

  Enter upper range:

  100
```

2.Enter lower range: 5

Enter upper range: 8

3.Enter lower range: 10

Enter upper range: 5

4.Enter lower range: 500

Enter upper range: 500

5.Enter lower range: 0

Enter upper range: -

100

```
lower_range = int(input("Enter lower range: "))
upper_range = int(input("Enter upper range: "))

result = []

for num in range(lower_range, upper_range + 1):
    square_root = int(num ** 0.5) # Calculate the square root of the number

    if square_root ** 2 == num:
        digit_sum = sum(int(digit) for digit in str(num)) # Calculate the sum of digits

        if digit_sum < 10:
            result.append(num)

print(result)
```

11. Write a program to print the number of vowels in the given statement?Sample Input:

Saveetha School of Engineering

Sample Output:

Number of vowels = 12

Test cases:

1. India is my country
2. All are my brothers and sisters
3. Why dry sky
4. Shy Try Cry
5. EDUCATION

```
s=input("Enter the string:")
vc=0
cc=0
v="AEIOUaeiou"
n="0123456789"
for i in s:
   if s[0] in n:
      print("String cannot start with a number")
      break
   elif i in v:
      vc=vc+1
   elif i in n:
      print("Please enter perfect string")
      break
   else:
      cc=cc+1
print("The count of vowels in string is =",vc)
print("The count of consonants in string is=",cc)
```

12. Write a program to print unique permutations of a given number

Sample Input:
Given Number: 143
Sample Output:
Permutations are:
134
143
314
341
413
431

Test cases:
1. 0
2. 111
3. 505
4. -143
5. -598

```
from itertools import permutations

number = input("Enter a number: ")

# Use set to store unique permutations
unique_permutations = set(permutations(number))

# Print the unique permutations
for perm in unique_permutations:
   print(".join(perm))
```

13. Python Program to Create a List of Tuples with the First Element as the Numberand Second Element as the Square of the Number.

Sample Input:

```
        Enter the lower range:45
        Enter the upper range:49
```

Sample Output:

```
        [(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]
```

Test case:

1. Enter lower range: 50

   Enter upper range:

   100

2. Enter lower range: 5

   Enter upper range: 8

3. Enter lower range: 10

   Enter upper range: 5

4. Enter lower range: 500

   Enter upper range: 500

5. Enter lower range: 0

   Enter upper range: -

   100

```
l1=[]
n1=int(input("enter a lower number:"))
n2=int(input("enter a upper number:"))
for i in range(n1,n2+1):
    s=(n1,n1**2)
    l1.append(s)
    n1=n1+1
    n2=n2+1
print(l1)
```

14.Python Program to Generate Random Numbers from A to B and Append Them to the List

Sample Input & Output:

```
Enter A Value: 20
Enter B Value:50
Enter number of elements:5
```

Sample Input & Output:

```
Randomized list is: [41, 39, 43, 24, 42]
```

Test Case:

1. A = 10, B = 0, Number of elements = 3
2. A = 100, B = 200, Number of elements = 30
3. A = 30, B = 270, Number of elements = 300
4. A = 0, B = 0, Number of elements = 5
5. A = -420, B = 420, Number of elements = -45

```
import random
a=[]
n=int(input("Enter number of elements:"))
A=int(input("Enter Value of A:"))
B=int(input("Enter the value of B:"))
for i in range(n):
    a.append(random.randint(A,B))
print(" Randomised list:",a)
```

15. Python Program to Remove the Duplicate Items from a List

Sample Input:

```
Enter the number of elements in list:7
Enter element1:10
```

```
            Enter element2:20
            Enter element3:20
            Enter element4:30
            Enter element5:40
            Enter element6:40
            Enter element7:50
Sample Output:
            Non-duplicate items:
            [10, 20, 30, 40, 50]
```

```
my_list = []
n=int(input("enter the number of elements:"))
for i in range(1,n+1):
        n1=int(input("Enter the elememts:"))
        my_list.append(n1)
print("List Before ", my_list)
temp_list = []

for i in my_list:
    if i not in temp_list:
        temp_list.append(i)
my_list = temp_list
my_list.sort()
print("List After removing duplicates ", my_list)
```

16. Find the maximum of three binary values using looping

Sample Input:

Given Numbers: 1101, 1011, 1001

Sample Output:

Maximum Number: 1101

```
# Get the numbers from the user as space-separated binary values
numbers = input("Enter the binary values (space-separated): ").split()

# Initialize the maximum number as the first number in the list
maximum = numbers[0]
```

```
# Iterate over the remaining numbers
for number in numbers[1:]:
    # Compare the current number with the maximum number
    # If the current number is greater, update the maximum number
    if number > maximum:
        maximum = number

# Print the maximum number
print("Maximum Number:", maximum)
```

17.Write a program to check if a given year is leap year or not. If it is leap year then print thenext leap year, if it is non leap year then print the previous leap year.

```
Sample Input:
Enter Date : 1947
Sample Output:
Given year is Non Leap Year
Leap Year: 1944
```

Test cases:

1. 19.47
2. 1936
3. 0
4. 2000
5. -1428

```
year = int(input("Enter Year: "))


# Check if the year is divisible by 4
if year % 4 == 0:
    # Check if the year is divisible by 100
    if year % 100 == 0:
        # Check if the year is divisible by 400
        if year % 400 == 0:
            print("Given year is Leap Year")
            next_leap_year = year + 4 - (year % 4)
            print("Next Leap Year:", next_leap_year)
        else:
            print("Given year is Non Leap Year")
            previous_leap_year = year - (year % 4)
            print("Leap Year:", previous_leap_year)
    else:
        print("Given year is Leap Year")
        next_leap_year = year + 4 - (year % 4)
        print("Next Leap Year:", next_leap_year)
else:
    print("Given year is Non Leap Year")
    previous_leap_year = year - (year % 4)
    print("Leap Year:", previous_leap_year)
```

18. Write a program that accepts a string from user and re displays the same string afterremoving vowels from it.

Sample Input & Output:

```
            Enter a string: we can play the game
The string without vowels is: w cn ply th gm
```

```
text =input("Enter the String: ")
vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']
newtext = ""
textlen = len(text)
for i in range(textlen):
    if text[i] not in vowels:
        newtext = newtext + text[i]
print("\nString after removing Vowels: ")
text = newtext
print(text)
```

19. Write a program to find the sum of digits of N digit number (sum should be singledigit)

Sample Input:

```
Enter N value : 3
Enter 3 digit number: 143
```

Sample Output:

Sum of 3 digit number: 8

Test cases:

1. N = 2, 158
2. N = 3, 14
3. N = 4, 0148
4. N = 1, 0004
5. N = 4, 7263

```
while True:
    try:
        s=int(input("Enter the size of the number:"))
        n=int(input("Enter the number:"))
        l=len(str(n))
        if(l!=s):
            print("Enter a equal number in size")
        else:

            s1=0
            while(n>0):
                dig=n%10
                s1=s1+dig
                n=n//10
            print("The sum of the digits is:",s1)
    except ValueError:
        print("Enter a valid input")
        continue
    else:
        break
```

20. Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:
Enter the word : MOSQUE
Sample Output:
Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION
2. MATRICULATION

3. MANIPULATION
4. SATISFACTION
5. DEDICATION

```
word = input("Enter the word: ")

# Convert the word to a list of characters
letters = list(word)

# Sort the letters in reverse alphabetical order
letters.sort(reverse=True)

# Join the sorted letters into a single string
sorted_word = ' '.join(letters)

print("Alphabetical Order:", sorted_word)
```

21. Find the LCM and GCD of n numbers?

Sample Input:
N value = 2
Number 1 = 16
Number 2 = 20
Sample Output:
LCM = 80
GCD = 4

Test cases:

1. N = 3, {12, 25, 30}
2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}
4. N = -2, {52, 60}
5. N = 2, {30, 45}

```
import math

n = int(input("Enter the value of N: "))
numbers = []
```

```
# Input n numbers
for i in range(1, n + 1):
    num = int(input(f"Number {i}: "))
    numbers.append(num)

# Calculate LCM
lcm = numbers[0]
for i in range(1, n):
    lcm = lcm * numbers[i] // math.gcd(lcm, numbers[i])

# Calculate GCD
gcd = numbers[0]
for i in range(1, n):
    gcd = math.gcd(gcd, numbers[i])

print("LCM =", lcm)
print("GCD =", gcd)
```

22.Write a program to print numbers from P to Q but except the digit R?

Sample Input:
P = 60
Q = 70
R = 3
Sample Output:

```
Numbers are = 60, 61, 62, 64, 65, 66, 67, 68, 69, 70
```

Test cases:

1. P = 200, Q = 200, R = 5
2. P = 100, Q = 200, R = 0
3. P = -100, Q = 100, R = 5
4. P = 1073, Q = 1075, R = 4
5. P = 444, Q = 499, R = 4

```
P = int(input("P = "))
Q = int(input("Q = "))
R = int(input("R = "))
```

```
numbers = []

for num in range(P, Q + 1):
    if str(R) not in str(num):
        numbers.append(num)

print("Numbers are =", ", ".join(map(str, numbers)))
```

23.Write a program to read a character until a * is encountered. Also count the number ofuppercase, lowercase, and numbers entered by the users.

```
Sample Input:
Enter * to exit…
Enter any character: W
Enter any character: d
Enter any character: A
Enter any character: G
Enter any character: g
Enter any character: H
Enter any character: *
Sample Output:
Total count of lower case:2
Total count of upper case:4
Total count of numbers =0
```

Test cases:

1. 1,7,6,9,5
2. S, Q, l, K,7, j, M
3. M, j, L, &, @, G
4. D, K, I, 6, L, *
5. *, K, A, e, 1, 8, %, *

```
upper_count = 0
lower_count = 0
number_count = 0

while True:
    char = input("Enter any character: ")

    if char == '*':
        break

    if char.isupper():
        upper_count += 1
    elif char.islower():
        lower_count += 1
```

```
                elif char.isdigit():
                    number_count += 1

            print("Total count of lowercase:", lower_count)
            print("Total count of uppercase:", upper_count)
print("Total count of numbers:", number_count)
```

24. Write a program using choice to check

Case 1: Given string is palindrome or not

Case 2: Given number is palindrome or not

Sample Input:

Case = 1

String = MADAM

Sample Output:

Palindrome

Test cases:

1. MONEY
2. 5678765
3. MALAY12321ALAM
4. MALAYALAM5.

1234.4321

```
import random


case = int(input("Case (1: String, 2: Number): "))


if case == 1:

    string = input("String: ").lower()

    reversed_string = string[::-1]


    if string == reversed_string:

        print("Palindrome")

    else:
        print("Not a Palindrome")

elif case == 2:
    number = int(input("Number: "))
    original_number = number
    reversed_number = 0

    while  number >  0:
        digit = number % 10
        reversed_number = (reversed_number * 10) + digit
        number //= 10

    if original_number == reversed_number:
        print("Palindrome")
    else:
        print("Not a Palindrome")

else:
    print("Invalid case selection")
```

25. Write a program to find the number of student users in the college, get the total users, staff users details from the client. Note for every 3 staff user there is one Non teaching staff user assigned by default.

```
Sample Input:
Total Users: 856
Staff Users: 126
Sample Output:
```

Student Users: 688

Test Cases:

1. Total User: 0

2. Total User: -143

3. Total User: 1026, Staff User: 1026

4. Total User: 450, Staff User: 540

5. Total User: 600, Staff User: 450

```
total_users = int(input("Total Users: "))
staff_users = int(input("Staff Users: "))

# Calculate the number of non-teaching staff users
non_teaching_staff_users = staff_users // 3

# Calculate the number of student users
student_users = total_users - staff_users - non_teaching_staff_users

print("Student Users:", student_users)
```

79. Write a program to print the Fibonacci series.

```
    Sample Input:
        Enter the n value: 6
```

Sample Output:

```
0    1    1    2    3    5
```

Test Condition: Implement negative Fibonacci series

```
n = int(input("Enter the n value: "))


# Check if n is positive or negative
if n >= 0:
    # Positive Fibonacci series
    a, b = 0, 1
    print(a, end=" ")
    for _ in range(n):
        print(b, end=" ")
        a, b = b, a + b
else:
    # Negative Fibonacci series
    a, b = 0, -1
    print(a, end=" ")
    for _ in range(abs(n)):
        print(b, end=" ")
        a, b = b, a - b
```

81. Write a program to convert the Binary to Decimal, Octal

Sample Input:
Given Number: 1101
Sample Output:
Decimal Number: 13
Octal:15

Test cases:

1. 211
2. 11011
3. 22122
4. 111011.011
5. 1010.0101

```
binary = input("Given Number: ")
```

```
decimal = int(binary, 2)

octal = oct(decimal).lstrip("0o")


print("Decimal Number:", decimal)

print("Octal:", octal)
```

82. Write a program to find the number of special characters in the given statement

Sample Input:
Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.
Sample Output:
Number of special Characters: 5

```
statement = input("Given statement: ")

special_chars = "!@#$%^&*()_+|<>?-=\{}[];',./:\""
count = 0

for char in statement:
   if char in special_chars:
      count += 1

print("Number of special Characters:", count)
```

84. Write a program to find the square, cube of the given decimal

numberSample Input:
Given Number: 0.6
Sample Output:
Square Number: 0.36
Cube Number:0.216

Test cases:

1. 12
2. 0
3. -0.5
4. 14.25
5. -296

```
number = float(input("Given Number: "))


square = number ** 2

cube = number ** 3


print("Square Number:", square)
print("Cube Number:", cube)
```

85. Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

Sample Input:

```
        Enter the string: I am a programmer
        Enter the character to be searched: p
```

Sample Output:

```
        P is found in string at index: 8
```

Note: Check for non available Character in the given statement as Hidden Test case.

```
string = input("Enter the string: ")
character = input("Enter the character to be searched: ")

index = -1
for i in range(len(string)):
   if string[i] == character:
      index = i
      break

if index != -1:
   print(character, "is found in the string at index:", index)
else:
   print(character, "is not found in the string.")
```

86. Write a program to find the number of letters repeatedly present in the given

wordSample Input:

Enter the word : TEMPLE

Sample Output:

```
    Number of repeated letters = 1
```

Test Case:

1. HYPOTHECATION
2. MATRICULATION
3. MANIPULATION
4. SIMPLIFICATION
5. DEDICATION

```
word = input("Enter the word: ")

repeated_letters = set()


for letter in word:

   if word.count(letter) > 1:
```

```
            repeated_letters.add(letter)


num_repeated_letters = len(repeated_letters)


print("Number of repeated letters =", num_repeated_letters)
```

88. Write a program to arrange the digits of the number in ascending or descending, get thechoice from user.{Note: A - Ascending, D - Descending , B - Both}

```
   Sample Input:
   Enter the number  :  6581
   Enter your choice (A/D/B): B
   Sample Output:
      Ascending order = 1 5 6 8
      Descending order = 8 6 5 1
      Test cases:
      1. 12121212
      2. 12345678
      3. 98784565
      4. ADSSDDR
      5. JK78SD98
```

```
number = int(input("Enter the number: "))

choice = input("Enter your choice (A/D/B): ")


digits  =  []

temp = number


# Extract the digits from the number

while temp > 0:

    digit = temp % 10

    digits.append(digit)

    temp //= 10


if choice == 'A':

    ascending_order = sorted(digits)

    print("Ascending order =", " ".join(str(d) for d in ascending_order))

elif choice == 'D':
    descending_order = sorted(digits, reverse=True)

    print("Descending order =", " ".join(str(d) for d in descending_order))

elif choice == 'B':

    ascending_order = sorted(digits)

    descending_order = sorted(digits, reverse=True)

    print("Ascending order =", " ".join(str(d) for d in ascending_order))

    print("Descending order =", " ".join(str(d) for d in descending_order))

else:

    print("Invalid choice!")
```

89.Write a program to print the number of negative numbers in an array of

numbersSample Input;:
Array of elements = {16, -18, 27, -16, 23, -21, 19}
Sample Output:
Number of negative numbers in Array elements = 3
Test cases:
1.Array of elements = {-26, 28, 37, -26, 33, -31, -29}
2. Array of elements = {1.6, 1.8, 2.7, -1.6, 2.3, -2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {-16, 2.8, -7, -1.5, 2.8, -2.8, -.19}
5. Array of elements = {-160, -160, -180, -270, -160, -230, -210, 1-90, 0}

```
array = [16, -18, 27, -16, 23, -21, 19]

count = 0


for num in array:

    if num < 0:

        count += 1


print("Number of negative numbers in Array elements =", count)
```

71. Write a program to merge two lists to the third

list?# Input lists

```
list1 = [1, 2, 3]

list2 = [4, 5, 6]
```

```
# Merge lists

list3 = list1 + list2


# Print the merged list

print("Merged List:", list3)
```

72. Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:
M = 50
N = 100
K = 7
Sample Output:
50, 58, 65, 72, …..
Test cases:
1. M = 15, N = 05, K = 02
2. .M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0 , N = 0 , K = 2
5. M = 200 , N = 200 , K = 50

```
    # Input values
    M = int(input("Enter the starting number (M): "))
    N = int(input("Enter the ending number (N): "))
    K = int(input("Enter the number of skips (K): "))

    # Print numbers with skipping
    for number in range(M, N + 1, K+1):
        print(number, end=" ")

print()
```

74. Find the Mean, Median, Mode of the array of numbers?

Sample Input;:
Array of elements = {16, 18, 27, 16, 23, 21, 19}
Sample Output:
Mean = 20
Median = 19
Mode = 16
Test cases:
1.Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 160, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100, 100}

```
from statistics import mean, median, mode
```

```
# Array of elements

numbers = [16, 18, 27, 16, 23, 21, 19]

# Mean

mean_value = mean(numbers)

# Median

median_value = median(numbers)

# Mode

mode_value = mode(numbers)

print("Mean =", mean_value)

print("Median =", median_value)

print("Mode =", mode_value)
```

76. Write a program to print consonants and vowels separately in the given word

Sample Input:
Given Word: Engineering
Sample Output:
Consonants: n g n r n g
Vowels: e i e e i

Test cases:

1. TRY
2. MEDIAN
3. ONE
4. KNOWLEDGE
5. EDUCATION

```
word = input("Enter the word: ")

consonants = []
vowels = []

for char in word:
    if char.isalpha():
        if char.lower() in ['a', 'e', 'i', 'o', 'u']:
            vowels.append(char)
        else:
            consonants.append(char)

print("Consonants:", ' '.join(consonants))
print("Vowels:", ' '.join(vowels))
```

Find the number of factors for the given number

Sample Input:
Given number: 100
Sample Output:
Number of factors = 9
Test cases:
1. 343
2. 1080
3. -243
4. 101010
5. 0

```
# Get the input from the user

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


# Initialize the count variable

count = 0


# Calculate the factors

for i in range(1, number + 1):

    if number % i == 0:

        count += 1


# Print the result

print("Number of factors =", count)
```

Write a program to calculate the factorial of number using recursive function.

Sample Input & Output:
    Enter the value of n: 6
Sample Input & Output:
    The factorial of 6 is: 720
Test cases:
1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

```
# Get the input from the user

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

```
# Initialize the factorial variable

factorial = 1


# Calculate the factorial

if n >= 0:

    for i in range(1, n + 1):

        factorial *= i


# Print the result

print("The factorial of", n, "is:", factorial)
```

53. Find the year of the given date is leap year or

notSample Input:
Enter Date : 04/11/1947
Sample Output:
Given year is Non Leap Year

Test cases:
1. 04/11/19.47
2. 11/15/1936
3. 31/45/1996
4. 64/09/1947
5. 00/00/2000

```
# Get the date from the user

date = input("Enter Date (MM/DD/YYYY): ")


# Extract the year from the date

year = int(date.split("/")[-1])


# Check if the year is a leap year

is_leap_year = False

if year % 4 == 0:

    if year % 100 == 0:

        if year % 400 == 0:

            is_leap_year = True
        else:
            is_leap_year = False
    else:
        is_leap_year = True
else:
    is_leap_year = False


# Print the result
if is_leap_year:
    print("Given year is a Leap Year")
else:
    print("Given year is not a Leap Year")
```

55. Write a program to convert Decimal number equivalent to Binary number andoctal numbers?

Sample Input:

Decimal Number: 15

Sample Output:

Binary Number = 1111

Octal = 17

Test cases:

1. 111
2. 15.2
3. 0
4. B12
5. 1A.2

```
# Get the decimal number from the user
decimal_number = int(input("Decimal Number: "))


# Convert to binary
binary_number = bin(decimal_number)[2:] # Remove the '0b' prefix


# Convert to octal
octal_number = oct(decimal_number)[2:] # Remove the '0o' prefix
```

```
# Print the results
print("Binary Number =", binary_number)
print("Octal =", octal_number)
```

56. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonuson salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Sample Input & Output:

```
Enter the grade of the employee: B
Enter the employee salary: 50000
Salary=50000
Bonus=5000.0
Total to be paid:55000.0
```

Test cases:

1. Enter the grade of the employee:
   AEnter the employee salary: 8000
2. Enter the grade of the employee:
   CEnter the employee salary: 60000
3. Enter the grade of the employee:
   BEnter the employee salary: 0
4. Enter the grade of the employee:
   38000Enter the employee salary: A
5. Enter the grade of the employee:
   BEnter the employee salary: -8000

```
# Get the grade and salary from the user
grade = input("Enter the grade of the employee: ")
salary = float(input("Enter the employee salary: "))


# Calculate the bonus based on grade
if grade == "A":
   bonus = 0.05 * salary
elif grade == "B":
   bonus = 0.10 * salary
```

```
# Check if the salary is less than $10,000
if salary < 10000:
    bonus += 0.02 * salary


# Calculate the total salary
total_salary = salary + bonus


# Print the salary and bonus
print("Salary =", salary)
print("Bonus =", bonus)
print("Total to be paid:", total_salary)
```

57. A Pythagorean triplet is a set of three integers a, b and c such that $a^2 + b^2 = c^2$. Given alimit, generate all Pythagorean Triples with values smaller than given limit?

```
# Get the limit from the user
limit = int(input("Enter the limit: "))

# Iterate through all possible values of a, b, and c
for a in range(1, limit):
    for b in range(a, limit):
        for c in range(b, limit):
            # Check if the numbers satisfy the Pythagorean theorem
            if a**2 + b**2 == c**2:
                # Print the Pythagorean triple
                print(a, b, c)
```

58.Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percent rate of interest; for all other customers, the ROI is 10 percent.

```
Sample Input:
        Enter the principal amount: 200000
        Enter the no of years: 3
        Is customer senior citizen (y/n): n
Sample Output:
        Interest: 60000
```

Test Cases:

1. Principal: 2000 , Years: 0
2. Principal: 20000 , Years: -2
3. Principal: -2000 , Years: 2
4. Principal: 2 , Years: 2000
5. Principal: 0 , Years: 5

```
# Get inputs from the user
principal = float(input("Enter the principal amount: "))
years = int(input("Enter the number of years: "))
senior_citizen = input("Is customer a senior citizen (y/n): ").lower() == 'y'

# Determine the rate of interest based on customer type
if senior_citizen:
    rate_of_interest = 12
else:
    rate_of_interest = 10

# Calculate the interest
interest = (principal * rate_of_interest * years) / 100

# Display the interest
print("Interest:", interest)
```

59. 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 Second 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:

a) 18, 76,93,65

b) 73,78,79,75
c) 98,106,120,95
d) 96,73, -85,95
e) 78,59.8,76,79

```
marks = []


# Get marks from the user for each subject
marks.append(float(input("Enter the marks in python: ")))
marks.append(float(input("Enter the marks in c programming: ")))
marks.append(float(input("Enter the marks in Mathematics: ")))
marks.append(float(input("Enter the marks in Physics: ")))

# Calculate total and aggregate
total = sum(marks)
aggregate = total / len(marks)

# Determine the grade based on the aggregate
if aggregate > 75:
   grade = "DISTINCTION"
elif aggregate  >=  60:
   grade = "First Division"
elif aggregate >= 50:
   grade = "Second Division"
elif aggregate >= 40:
   grade = "Third Division"
else:
   grade = "Fail"

# Display the results
print("Total =", total)
print("Aggregate =", aggregate)
print(grade)
```

61. Write a program that would sort a list of names in alphabetical order Ascending orDescending, choice get from the user?

```
Sample Input:
Banana
Carrot
Radish
Apple
Jack
Order(A/D) : A
Sample Output:
Apple
Banana
Carrot
Jack
Radish
```

```
names = []


# Get input from user until they enter '-1'
while True:
    name = input("Enter a name (enter -1 to stop): ")
    if name == '-1':
        break
    names.append(name)


order = input("Enter the order (A for ascending, D for descending): ")


# Sort the names based on the order
if order.lower() == 'a':
    names.sort()
elif order.lower() == 'd':
    names.sort(reverse=True)


# Print the sorted names
for name in names:
    print(name)
```

62. Write a program for matrix multiplication and matrix

addition?Sample Input:

```
Mat1 = 1  2

        5  3

Mat2 = 2   3

        4   1
```

Sample Output:

```
Mat Sum = 10    5

            22    18
```

```
mat1=[[1,2],[3,4]]
mat2=[[5,6],[7,8]]
# Perform matrix multiplication
mat_mult = [[0, 0], [0, 0]]
for i in range(2):
    for j in range(2):
        for k in range(2):
            mat_mult[i][j] += mat1[i][k] * mat2[k][j]

# Perform matrix addition
mat_sum = [[0, 0], [0, 0]]
for i in range(2):
    for j in range(2):
        mat_sum[i][j] = mat1[i][j] + mat2[i][j]

# Print the result
print("Matrix Multiplication:")
for row in mat_mult:
    print("\t".join(str(num) for num in row))

print("Matrix Addition:")
for row in mat_sum:
print("\t".join(str(num) for num in row))
```

63. Write a program to print the multiplication table of number m up to

n.Sample Input:
M = 4
N = 5
Sample Output:

```
1x4=4
2x4=8
3x4=12
4x4=16
5x4=20
```

Test cases:

1. M = 6, N = -3
2. M = -3, N = 5
3. M = 4, N = 0
4. M = 0, N = 0
5. M = -5, N = -5

```
# Prompt the user for input
M = int(input("M: "))
N = int(input("N: "))

# Print the multiplication table
for i in range(1, N + 1):
    result = M * i
    print("{}x{}={}".format(i, M, result))
```

64. Write a program to print the special characters separately and print number of Specialcharacters in the line?

```
line = input("Enter a line of text: ")
special_chars = ""

for char in line:
    if not char.isalnum() and char != " ":
        special_chars += char
special_count = len(special_chars)


print("Special characters:", special_chars)

print("Number of special characters:", special_count)
```

65. Write a program to read the numbers until -1 is encountered. Find the average ofpositive numbers and negative numbers entered by user.

Sample Input:

```
        Enter -1 to exit…
        Enter the number: 7
        Enter the number: -2
        Enter the number: 9
        Enter the number: -8
        Enter the number: -6
        Enter the number: -4
        Enter the number: 10
        Enter the number: -1
```

Sample Output:

```
        The average of negative numbers is: -5.0
        The average of positive numbers is : 8.66666667
```

Test cases:

1. -1,43, -87, -29, 1, -9
2. 73, 7-6,2,10,28,-1
3. -5, -9, -46,2,5,0
4. 9, 11, -5, 6, 0,-1
5. -1,-1,-1,-1,-1

```
positive_sum = 0

positive_count = 0

negative_sum = 0

negative_count = 0


while True:

   number = int(input("Enter the number (enter -1 to exit): "))


   if number == -1:

      break


   if number >= 0:
        positive_sum += number
        positive_count += 1
    else:
        negative_sum += number
        negative_count += 1

if positive_count > 0:
    positive_average = positive_sum / positive_count
    print("The average of positive numbers is:", positive_average)

if negative_count > 0:
    negative_average = negative_sum / negative_count
    print("The average of negative numbers is:", negative_average)
```

67. Python Program to Find the Nth Largest Number in a List

Sample Input:
List : {14, 67, 48, 23, 5, 62}
N = 4
Sample Output:
4th Largest number: 23

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

```
# Input list
lst = [14, 67, 48, 23, 5, 62]
N = int(input("Enter the value of N: "))

# Sort the list in descending order
sorted_list = sorted(lst, reverse=True)

# Check if N is within the range of the list
if 1 <= N <= len(sorted_list):
        nth_largest = sorted_list[N - 1]

        print(f"{N}th Largest number: {nth_largest}")

    else:

        print("Invalid value of N")
```

. Write a program to find the square root of a perfect square number(print both the positive and negative values)

Sample Input:

Enter the number : 6561
Sample Output:

Square Root: 81, -81
Test cases:

1. 1225
2. 9801
3. 1827
4. -100
5. 0

```
import math


# Get the input from the user

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


# Calculate the square root

square_root = math.isqrt(number)


# Print the positive and negative square roots

print("Square Root:", square_root, ",", -square_root)
```

45. 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:

1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2

```
# Get the input from the user

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


# Find and print the first n perfect
numberscount = 0

num = 1
perfect_numbers =
[]


while count < n:
    divisor_sum = 0

    for i in range(1, num):
        if num % i == 0:

            divisor_sum += i


    if divisor_sum == num:
        perfect_numbers.append(num)
        count += 1


    num += 1


# Print the first n perfect numbers

print("First", n, "perfect numbers are:", ", ".join(str(num) for num in
perfect_numbers))
```

47. Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6
Sample Output:

Its a Perfect Number

Test cases:

1. 17
2. 26!

3. 143

4. 84.1

5. -963

```
# Get the input from the user
number = int(input("Enter the number: "))


# Find the sum of proper divisors of the
numberdivisor_sum = 0

for i in range(1, number):
    if number % i == 0:

        divisor_sum += i


# Check if the number is a perfect
numberif divisor_sum == number:

    print("It's a Perfect
Number")else:

    print("It's not a Perfect Number")
```

49. Write a program to print number of factors and to print nth factor of thegiven number.

Sample Input:

Given Number: 100

N = 4

Sample Output:

Number of factors = 9
4th factor of 100 = 5
Test Cases:

1. Given Number = 512 , N = 6
2. Given Number = 343 , N = 7
3. Given Number = 1024 , N = 0
4. Given Number = -6561 , N = 3
5. Given Number = 0 , N = 2

```
# Get the input from the user
```

```
number = int(input("Enter the number: "))


# Calculate the number of
factorscount = 0

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

        count += 1


# Print the number of factors
print("Number of factors =",
count)


# Get the input for the nth factor
n = int(input("Enter the value of n: "))


# Find the nth
factorfactor_count
= 0

nth_factor = 0
for i in range(1, number + 1):
    if number % i == 0:

        factor_count += 1
        if factor_count == n:
            nth_factor = i
            break


# Print the nth factor
print(f"{n}th factor of {number} =", nth_factor)
```

3. Write a program to reverse a number using loop?(Get the input from user)

Sample Input:
Number: 145677
Sample Output:
Reverse Number: 76541

Test cases:

1. -45721

2. 000

3. AD1947

4. !@#$%

5. 145*999=144855

```
try:
 # Input number
    number = int(input("Enter a number: "))


# Initialize variables
    reverse = 0


# Reverse the number using a loop
    while number > 0:
        remainder = number % 10
        reverse = (reverse * 10) + remainder
        number = number // 10


# Print the reverse number
    print("Reverse Number:", reverse)
except:
   print("print the numbers")
```

27. Write a program to check the entered user name is valid or not. Get both the inputs from the user.

Sample Input:

```
Enter the user name: Saveetha@789
Reenter the user name: Saveetha@123
```

Sample Output:

```
User name is Invalid
```

```
# Get the inputs from the user
user_name = input("Enter the user name: ")
re_entered_user_name = input("Reenter the user name: ")
```

```
# Check if the user names match and meet the validity criteria
if user_name == re_entered_user_name:
    # Check the validity criteria (e.g., no special characters allowed)
    if user_name.isalnum():
        print("User name is Valid")
    else:
        print("User name is Invalid")
else:
    print("User names do not match")
```

Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

Sample Input:
Enter your age:
7
Sample output:
You are allowed to vote after 11 years

Test cases:

1. 25
2. Eighteen
3. 12
4. -18
5. 34.5

```
# Get the age input from the user
age = int(input("Enter your age: "))


# Check if the person is eligible to vote
if age >= 18:
    print("You are eligible to vote.")
else:
    years_left = 18 - age
    print("You are not eligible to vote.")
    print("You are allowed to vote after", years_left, "years.")
```

Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number : 153
Sample Output:

Given number is Armstrong number

Test cases:

1. 370
2. 1
3. 371
4. 145678
5. 0.21345

```
# Get the number input from the user
number = int(input("Enter number: "))


# Calculate the number of digits
num_of_digits = len(str(number))


# Initialize sum variable
sum = 0

temp = number


# Calculate the sum of each digit raised to the power of the number ofdigits

while temp > 0: digit
    = temp % 10

    sum += digit ** num_of_digits
    temp //= 10


# Check if the number is an Armstrong numberif
number == sum:

    print("Given number is Armstrong number")
else:

    print("Given number is not Armstrong number")
```

Given a string s representing a valid expression, implement a basic calculator to evaluate it, and return the result of the evaluation.

Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval().F

Example 1:

Input: s = "1 + 1"

Output: 2

```
s = "1 + 1"

stack = []
num = 0
sign = 1
result = 0

for char in s:
    if char.isdigit():
        num = num * 10 + int(char)
    elif char == '+':
        result += sign * num
        num = 0
        sign = 1
    elif char == '-':
        result += sign * num
        num = 0
        sign = -1
    elif char == '(':
        stack.append(result)
        stack.append(sign)
        result = 0
        sign = 1
    elif char == ')':
        result += sign * num
        num = 0
        result *= stack.pop()
        result += stack.pop()

result += sign * num
print(result)
```