
ASSIGNMENT

Submitted by,

Jesso Joseph

Roll No : 27

1. GCD of two numbers

```
a= int(input("Enter First Number:"))
b= int(input("Enter Second Number:"))

for i in range (1, min(a,b)+1):
    if a%i ==0 and b%i ==0:
        gcd=i
print(gcd)
```

Output:

Enter First Number:10

Enter Second Number:20

10

2. Factorial

```
a= int(input("Enter the Number:"))
fact =1
for i in range (1,a+1):
    fact = fact*i
print(fact)
```

Output:

Enter the Number:5

120

3. Fibonacci Series

```
n =int(input("Enter the number of termes:"))
f1,f2=0,1
f3=f1+f2
print("Fionacci seies of first", n, "termes")
print(f1)
print(f2)
for i in range (3, n+1):
    print(f3)
    f1=f2
    f2=f3
    f3=f1+f2
```

Output:

Enter the number of termes:4

Fionacci seies of first 4 termes

0

1

1

2

4. Count of vowels

```
p = input("Enter a string:")
count=0
for i in p:
    if i in 'aeiouAEIOU':
        count+=1
print("Count of vowels :", count)
```

Output:

Enter a string:Amaljyothi

Count of vowels : 4

5. Sum of all items in list

```
list=[]
limit=int(input("Enter a limit: "))
print("Enter numbers: ")
for i in range(1, limit+1):
    nums=int(input())
    list.append(nums)
print(list)
sum=0
for i in range(len(list)):
    sum=sum+list[i]
print(sum)
```

Output:

Enter a limit: 4

Enter numbers:

1

2

3

4

[1, 2, 3, 4]

10

6. Pyramid

```
def print_number_pyramid(num_lines):  
    for i in range(1, num_lines + 1):  
        for j in range(1, i + 1):  
            print(j, end=" ")  
        print()  
  
num_lines = int(input("Enter the number of lines for the number pyramid: "))  
print_number_pyramid(num_lines)
```

Output:

Enter the number of lines for the number pyramid: 5

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

7. Pythagorean Triads

```
for i in range(1,50):  
    for j in range(1,i):  
        for k in range(1,j):  
            if k*k + j*j == i*i:  
                flag=0  
                for l in range(2,i):  
                    if i%l==0 and j%l==0 and k%l==0:  
                        flag=1  
                        break  
                if flag: continue  
            print("a=",k,"b=",j,"c=",i)
```

Output:

```
a= 3 b= 4 c= 5  
a= 5 b= 12 c= 13  
a= 8 b= 15 c= 17  
a= 7 b= 24 c= 25  
a= 20 b= 21 c= 29  
a= 12 b= 35 c= 37  
a= 9 b= 40 c= 41
```

8. Chess – Bishop movements

```
def generate_bishop_moves(board_size, bishop_position):
    r, c = bishop_position
    moves = []

    i, j = r - 1, c - 1
    while i >= 0 and j >= 0:
        moves.append((i, j))
        i -= 1
        j -= 1

    i, j = r - 1, c + 1
    while i >= 0 and j < board_size:
        moves.append((i, j))
        i -= 1
        j += 1

    i, j = r + 1, c - 1
    while i < board_size and j >= 0:
        moves.append((i, j))
        i += 1
        j -= 1

    i, j = r + 1, c + 1
    while i < board_size and j < board_size:
        moves.append((i, j))
        i += 1
        j += 1

    return moves

board_size = int(input("Enter the board size: "))

bishop_row = int(input("Enter the Bishop's row (0-based index): "))
bishop_col = int(input("Enter the Bishop's column (0-based index): "))
bishop_position = (bishop_row, bishop_col)

bishop_moves = generate_bishop_moves(board_size, bishop_position)
print("Valid Bishop moves:", bishop_moves)
```

Output:

Enter the board size: 6

Enter the Bishop's row (0-based index): 2

Enter the Bishop's column (0-based index): 5

Valid Bishop moves: [(1, 4), (0, 3), (3, 4), (4, 3), (5, 2)]

9. Count a number in list

```
num_list = input("Enter a list of numbers: ").split()
num_list = [int(num) for num in num_list]
target_number = int(input("Enter the number to count: "))
count = num_list.count(target_number)
print(f"The number {target_number} appears {count} times in the list.")
```

Output:

Enter a list of numbers: 2 5 4 8 6 2 4 2

Enter the number to count: 2

The number 2 appears 3 times in the list.

10. n copies of first 2 characters of string

```
input_string = input("Enter a string: ")
n = int(input("Enter the nof copies: "))

first_two_chars = input_string[:2]
if len(input_string) < 2:
    result = input_string * n
else:
    result = first_two_chars * n
print("Result:", result)
```

Output:

Enter a string: Amal Jyothi College

Enter the nof copies: 2

Result: AmAm

11. Check whether value contained in list of values

```
values = input("Enter a list of values : ").split()
values = [int(value) for value in values]
specified_value = int(input("Enter the number to search for: "))
if specified_value in values:
    print(f"{specified_value} is present in the list.")
else:
    print(f"{specified_value} is not present in the list.")
```

Output:

Enter a list of values : 2 5 4 8 2 4 6

Enter the number to search for: 2

2 is present in the list.

12. Print even no upto 237

```
numbers = [10, 15, 20, 30, 237, 40, 50, 60, 70]
for num in numbers:
    if num == 237:
        break
    if num % 2 == 0:
        print(num)
```

Output:

10

20

30

13. Write a Python program to print all even numbers from a given numbers list in the same order and stop the printing if any numbers that come after 237 in the sequence.

```
input_str = input("Enter a list of numbers :")
numbers = [int(num) for num in input_str.split(',')]
encountered_237 = False
for num in numbers:
    if num % 2 == 0:
        print(num)
    if num > 237:
        encountered_237 = True
        break
if encountered_237:
    print("Stopped printing because a number greater than 237 was encountered.")
```

Output:

Enter a list of numbers :2,5,7,6

2

6

14. LCM of two positive integers.

```
import math
try:
    num1 = int(input("Enter the first positive integer: "))
    num2 = int(input("Enter the second positive integer: "))
```

```

    if num1 <= 0 or num2 <= 0:
        print("Please enter positive integers.")
    else:
        result = math.lcm(num1, num2)
        print(f"The LCM of {num1} and {num2} is {result}.")
except ValueError:
    print("Invalid input. Please enter valid positive integers.")

```

Output:

Enter the first positive integer: 2

Enter the second positive integer: 5

The LCM of 2 and 5 is 10.

15. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string
If the string length is less than 2, return instead the empty string.

```

input_string = input("Enter a string: ")
if len(input_string) < 2:
    result = ""
else:
    result = input_string[:2] + input_string[-2:]

print("Result:", result)

```

Output:

Enter a string: Python Programming

Result: Pyng

16. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string is already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

```

str1=input("Enter a String")
length = len(str1)
if length > 2:
    if str1[-3:] == 'ing':
        str1 += 'ly'
    else:
        str1 += 'ing'
print("New String:",str1)

```

Output:

Enter a String: Python Program

New String: Python Programing

17. Write a Python function that takes a list of words and returns the length of the longest one.

```
str1 = input("Enter the string : ")
word_list = str1.split()
word_new=[]
for i in word_list:
    word_new.append((len(i),i))
    word_new.sort()
print("Longest : ",word_new[-1][1])
```

Output:

Enter the string : apple orange pearl kiwi

Longest : orange

18. Remove odd index character

```
def odd_values_string(input_string):
    result = ""
    for i in range(len(input_string)):
        if i % 2 == 0:
            result = result + input_string[i]
    return result

user_input = input("Enter a string: ")
result_string = odd_values_string(user_input)
print("Resulting string with odd index characters:", result_string)
```

Output:

Enter a string: python programming

Resulting string with odd index characters: pto rgamn

19. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

```
str1 = input("Enter the string : ")
word_list = str1.split(',')
word_new=[]
for i in word_list:
    word_new=(','.join(sorted(list(set(word_list)))))
print(word_new)
```

Output:

Enter the string : Amal,Jyothi,college

Amal,Jyothi,college

20. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

```
def match_words(words):
    ctr = 0

    for word in words:
        if len(word) > 1 and word[0] == word[-1]:
            ctr += 1
    return ctr

user_input = input("Enter a list of words separated by spaces: ")
user_words = user_input.split()
result = match_words(user_words)
print(result)
```

Output:

Enter a list of words separated by spaces: cat bob pop dog am pat abba

3

21. Check list is empty or not.

```
my_list = input("Enter a list : ").split()
if not my_list:
    print("The list is empty.")
else:
    print("The list is not empty.")
```

Output:

Enter a list : 1 2 3 4 5

The list is not empty.

22. List of words that are longer.

```
words = input("Enter a list of words separated by spaces: ").split()
n = int(input("Enter the value of 'n': "))
longer_than_n = [word for word in words if len(word) > n]
print(f"Words longer than {n} characters: {longer_than_n}")
```

Output:

Enter a list of words separated by spaces: aby amal anandhu febin joseph

Enter the value of 'n': 2

Words longer than 2 characters: ['aby', 'amal', 'anandhu', 'febin', 'joseph']

23. Remove Indices

```
my_list = input("Enter a list of elements separated by spaces: ").split()
my_list = [int(item) for item in my_list]
indices_to_remove = [0, 2, 4, 5]
result_list = [element for index, element in enumerate(my_list) if index not in
indices_to_remove]
print(result_list)
```

Output:

Enter a list of elements separated by spaces: 4 8 7 6 4 2 5

[8, 6, 5]

24. Write a Python program to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30 (both included).

```
my_list = []
for i in range(1, 21):
    my_list.append(12)

print(my_list[:5])
print(my_list[-5:])
```

Output:

[12, 12, 12, 12, 12]

[12, 12, 12, 12, 12]

25. Printing dictionary.

```
n = int(input("Enter a value for 'n': "))
result_dict = {x: x*x for x in range(1, n + 1)}
print(result_dict)
```

Output:

Enter a value for 'n': 4

{1: 1, 2: 4, 3: 9, 4: 16}

26. Convert temperature.

```
def celsius_to_fahrenheit(celsius):
    return (celsius * 9/5) + 32

def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5/9
```

```

choice = input("Choose conversion type (C to F or F to C): ").upper()

if choice == "C TO F":
    celsius = float(input("Enter temperature in Celsius: "))
    fahrenheit = celsius_to_fahrenheit(celsius)
    print(f"{celsius}°C is equal to {fahrenheit}°F")

elif choice == "F TO C":
    fahrenheit = float(input("Enter temperature in Fahrenheit: "))
    celsius = fahrenheit_to_celsius(fahrenheit)
    print(f"{fahrenheit}°F is equal to {celsius}°C")

else:
    print("Invalid choice. Please choose 'C to F' or 'F to C'.")

```

Output:

Choose conversion type (C to F or F to C): c to f

Enter temperature in Celsius: 37

37.0°C is equal to 98.6°F

27. Reverse string.

```

user_input = input("Enter a string: ")
reversed_string = user_input[::-1]
print("Reversed string:", reversed_string)

```

Output:

Enter a string: Amal Jyothi

Reversed string: ihtoyJ lama

28. Count odd and even numbers from list.

```

numbers = input("Enter a list of numbers separated by spaces: ").split()
numbers = [int(number) for number in numbers]
even_count = 0
odd_count = 0
for number in numbers:
    if number % 2 == 0:
        even_count += 1
    else:
        odd_count += 1
print(f"Number of even numbers: {even_count}")
print(f"Number of odd numbers: {odd_count}")

```

Output:

Enter a list of numbers separated by spaces: 4 8 1 3 7 6

Number of even numbers: 3

Number of odd numbers: 3

29. Divisibility by 5.

```
binary_sequence = input("Enter a sequence of comma-separated binary numbers (4
digits each): ").split(',')
divisible_by_5 = []
for binary_number in binary_sequence:
    decimal_number = int(binary_number, 2)
    if decimal_number % 5 == 0:
        divisible_by_5.append(binary_number)
result_sequence = ','.join(divisible_by_5)
print("Numbers divisible by 5:", result_sequence)
```

Output:

Enter a sequence of comma-separated binary numbers (4 digits each): 0101,0110,0111,1010

Numbers divisible by 5: 0101,1010

30. Numbers between 100 and 400

```
even_digit_numbers = []
for number in range(100, 401):
    num_str = str(number)

    if all(int(digit) % 2 == 0 for digit in num_str):
        even_digit_numbers.append(str(number))

result_sequence = ','.join(even_digit_numbers)

print("Numbers with all even digits between 100 and 400:", result_sequence)
```

Output:

Numbers with all even digits between 100 and 400:

200,202,204,206,208,220,222,224,226,228,240,242,244,246,248,260,262,264,266,268,280,282,284,286,
288,400

Functions

1. Write a python function to check whether a number is even or odd

```
def is_even_or_odd(number):  
    if number % 2 == 0:  
        return "Even"  
    else:  
        return "Odd"  
  
num = int(input("Enter a number: "))  
result = is_even_or_odd(num)  
print(f"The number {num} is {result}.")
```

Output:

Enter a number: 7

The number 7 is Odd.

2. Write a python program to calculate the sum of three given numbers,if the values are equal then return thrice of their sum

```
def sum_of_three_numbers(a, b, c):  
    if a == b == c:  
        return 3 * (a + b + c)  
    else:  
        return a + b + c  
  
num1 = float(input("Enter the first number: "))  
num2 = float(input("Enter the second number: "))  
num3 = float(input("Enter the third number: "))  
result = sum_of_three_numbers(num1, num2, num3)  
print(f"The result is: {result}")
```

Output:

Enter the first number: 3

Enter the second number: 3

Enter the third number: 3

The result is: 27

3. Write a python function to get a new string from a given string where "Is" has been added to the front. If the given string already begins with "Is" then return the string unchanged.

```
def new_string(s):  
3.     if len(s) >= 2 and s[:2] == "is":  
4.         return s  
5.     return "is" + s  
6.
```

```
7.     str1 = input("Enter a string: ")
8.     print("New String:", new_string(str1))
```

Output:

Enter a string: Happy Day

New String: isHappy Day

4. Write a python program to get a string which is n(non-negative integer) copies of a given string

```
def larger_str(s, n):
    result = ""
    for _ in range(n):
        result += s
    return result

str1 = input("Enter a string: ")
n = int(input("Enter the number of copies: "))
print(larger_str(str1, n))
```

Output:

Enter a string: Morning

Enter the number of copies: 5

MorningMorningMorningMorningMorning

5. Write a python function that will return true if the two given integer values are equal or their sum or difference is 5.

```
def check_integer_values(num1, num2):
    return num1 == num2 or num1 + num2 == 5 or abs(num1 - num2) == 5

num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))
result = check_integer_values(num1, num2)

if result:
    print("True")
else:
    print("False")
```

Output:

Enter the first integer: 3

Enter the second integer: 3

True

6. Write a python program to display Fibonacci series using recursion.

```
def fibonacci_recursive(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    elif n == 2:
        return [0, 1]
    else:
        fib_series = fibonacci_recursive(n - 1)
        fib_series.append(fib_series[-1] + fib_series[-2])
        return fib_series

n = int(input("Enter the number of terms for Fibonacci series: "))
fib_series = fibonacci_recursive(n)
print("Fibonacci Series (First", n, "terms):", fib_series)
```

Output:

Enter the number of terms for Fibonacci series: 5

Fibonacci Series (First 5 terms): [0, 1, 1, 2, 3]

7. Write a python function to find the sum of digits of a number.

```
def sum_of_digits(n):
    num_str = str(n)
    digit_sum = 0
    for digit in num_str:
        digit_sum += int(digit)
    return digit_sum

n = 123456
result = sum_of_digits(n)
print(result)
```

Output:

123456

21

8. Write a python function to concatenate two strings.

```
def concatenate_strings(str1, str2):
    return str1 + str2

string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
result = concatenate_strings(string1, string2)
print("Concatenated string:", result)
```

Output:

Enter the first string: Hello

Enter the second string: World

Concatenated string: HelloWorld

9. Write a python function called compare which takes two strings s1 and s2 and an integer n as arguments. The function should return True if first n characters of both the strings are same else the function should return False.

```
def compare(s1, s2, n):  
    return s1[:n] == s2[:n]  
  
s1 = "exam"  
s2 = "example"  
n = 3  
result = compare(s1, s2, n)  
print(result)
```

Output:

True