problem/04.py

คำนวณผลรวมของข้อมูล ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
num_1 = float(input("Enter number 1 :"))

num_2 = float(input("Enter number 2 :"))

num_3 = float(input("Enter number 3 :"))

num_4 = float(input("Enter number 4 :"))

num_5 = float(input("Enter number 5 :"))

sum = (num_1 + num_2 + num_3 + num_4 + num_5)

avg =  sum / 5

print("Average: ", avg)

print("Sum: ", sum)
```

---

problem/05.py

และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

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

i = 1

while(i <= num):

    if(num % i == 0):

        print(i)

    i+=1
```

---

problem/06.py

ประกาศฟังก์ชัน `check_prime(number)` ; ตรวจสอบ/หาจำนวนเฉพาะในช่วงที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def check_prime(number):

    if number > 1 :

        print(f"Validate Input: {number} is greater than 1, proceed to next step.")

        print("Prime Check: ", end="")

        isPrime = True

        divsible = ""

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

            if i > 1 and i!= number and number % i == 0 :

                isPrime = False

                divsible += f", {i}"

        if isPrime :

            print(f"{number} is only divisible by 1 and {number}.")

            print("is prime")

        else:

            print(f"{number} is divisible by 1{divsible}, {number}.")

            print("is not prime")

    else:

        print("is not prime")

number = int(input("Enter number that greater than 1: "))

check_prime(number)
```

---

problem/07.py

ประกาศฟังก์ชัน `prime_numbers_in_range(start, end)` ; ตรวจสอบ/หาจำนวนเฉพาะในช่วงที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def prime_numbers_in_range(start, end):
    result = []
    if start < 1 or start > 1000 or end < 1 or end > 1000:
        print("out of range")
        return None
    for i in range (start, end+1):
        is_Prime = True
        for j in range(1, i+1):
            if j > 1 and j < i and i % j == 0 :
                is_Prime = False
        if is_Prime == True and j != 1 :
            result.append(j)
    return(result,sum(result))
start = int(input("Please enter starting number: "))
end = int(input("Please enter ending number: "))
print(prime_numbers_in_range(start, end))
```

---

problem/08.py

ประกาศฟังก์ชัน `separate_by_index(word)` ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def separate_by_index(word):
    even_char = ""
    odd_char = ""
    if(len(word) > 100):
        print("out of range")
        return(None)
    for i , char in enumerate(word):
        if i == 0 or i % 2 == 0 :
            even_char += char
        else:
            odd_char += char
    print(even_char, odd_char)
word = input("Enter word: ")
separate_by_index(word)
```

---

problem/09.py

ประกาศฟังก์ชัน `average_length_of_strings(list_words)` ; คำนวณผลรวมของข้อมูล ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def average_length_of_strings(list_words):
    word_lenth = []
    for word in list_words:
```

```
        word_lenth.append(len(word))
    return(sum(word_lenth) / len(word_lenth))
list_words = []
for i in range(5):
    while(True):
        words = input(f"please enter word {i+1} :")
        if len(words) < 100 :
            list_words.append(words)
            break
        else:
            print("Out of range please try again.")
print(average_length_of_strings(list_words))
```

---

problem/10.py

ประกาศฟังก์ชัน `character_frequency(words)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def character_frequency(words):
    result = {}
    for word in words:
        for char in word:
            if char not in result:
                result[char] = 1
            else:
                result[char] += 1
                pass
    print(result)
words = []
for i in range (5):
    while(True):
        word = input(f"please input word {i+1}: ")
        if(len(word) < 50 ):
            words.append(word)
            break
        else:
            print("Out of range please try again")
character_frequency(words)
```

---

problem/11.py

ประกาศฟังก์ชัน `contains_vowel(word)` ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def contains_vowel(word):
    vowel = ['a','e','i','o','u']
    is_vowel = False
    for char in word:
```

```
        if char.lower() in vowel:
            is_vowel = True
    return(is_vowel)
word = input("Enter your word: ")
print(contains_vowel(word))
```

---

problem/12.py

ประกาศฟังก์ชัน `replace_characters(word)` ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def replace_characters(word):
    word = word.replace("a","@")
    word = word.replace("l","1")
    word = word.replace("o","0")
    return(word)
word = input("Please input word: ")
print(replace_characters(word))
```

---

problem/13.py

ประกาศฟังก์ชัน `reverse_string(string)` ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def reverse_string(string):
    reverse_string = "".join(reversed(string))
    return(reverse_string)
string = input("Enter a word: ")
print(reverse_string(string))
```

---

problem/14.py

ประกาศฟังก์ชัน `collect_unique_words()` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def collect_unique_words():
    words = []
    while (len(words) < 5):
        word = input("Enter word: ")
        if word not in words:
            words.append(word)
    return(words)
print(collect_unique_words())
```

---

problem/15.py

ประกาศฟังก์ชัน `count_word_occurrences(words)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def count_word_occurrences(words):
    word_count = {}
    for word in words:
        if word not in word_count:
            word_count[word] = 1
        else:
```

```
            word_count[word] += 1
    return(word_count)
words = ["apple","banana","apple","orange","banana","apple"]
print(count_word_occurrences(words))
```

---

problem/16.py

ประกาศฟังก์ชัน `insert_at_front(words)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def insert_at_front(words):
    result = []
    for word in words:
        print(f"Add word: {word} ->",end="")
        result.append(word)
        result.reverse()
        print(f"List become{result}")
    return(result)
words = ["apple","kuy", "banana", "cherry"]
print(insert_at_front(words))
```

---

problem/17.py

ประกาศฟังก์ชัน `is_word_in_list(word_list)` ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def is_word_in_list(word_list):
    search_term = input("Enter word u want to search: ")
    if search_term in (word_list):
        return(True)
    else:
        return(False)
word_list = ["apple", "banana", "cherry", "date", "elderberry", "fig", "grape", "honeydew","kiwi", "lem
print(is_word_in_list(word_list))
```

---

problem/18.py

ประกาศฟังก์ชัน `remove_word_from_list(words)` ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def remove_word_from_list(words):
    remove_word = input("Which word u want to remove?: ")
    if remove_word in words:
        words.remove(remove_word)
    return(words)
words = ["apple", "banana", "cherry", "date", "elderberry", "fig", "grape","honeydew", "kiwi", "lemon"]
print(remove_word_from_list(words))
```

---

problem/19.py

ประกาศฟังก์ชัน `sum_matrices(matrix1, matrix2)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def sum_matrices(matrix1, matrix2):
    result = []
```

```
    for i in range(len(matrix1)):
        result_sum = []
        for j in range (len(matrix1[i])):
            total = matrix1[i][j] + matrix2[i][j]
            result_sum.append(total)
        result.append(result_sum)
    return(result)
matrix1 = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
matrix2 = [[4,3,2,1],[4,3,2,1],[4,3,2,1]]
print(sum_matrices(matrix1, matrix2))
```

---

problem/20.py

ประกาศฟังก์ชัน `transpose_matrix(matrix1)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def transpose_matrix(matrix1):
    num_cols = len(matrix1[0])
    result = []
    for i in range(num_cols):
        result.append([])
    for i in range(len(matrix1)):
        for j in range(len(matrix1[i])):
            result[j].append(matrix1[i][j])
    return(result)
matrix1 = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
print(transpose_matrix(matrix1))
```

---

problem/21.py

ประกาศฟังก์ชัน `calculate_statistics(numbers)` ; คำนวณผลรวมของข้อมูล ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def calculate_statistics(numbers):
    result = []
    for num in numbers:
        result.append(num*num)
    maximum = max(result)
    minimum = min(result)
    sum_value = sum(result)
    average = sum_value/len(result)
    return(result,maximum,minimum,sum_value,average)
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(calculate_statistics(numbers))
```

---

problem/22.py

ประกาศฟังก์ชัน `create_dictionary(list1, list2)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```python
def create_dictionary(list1, list2):
    result = {}
    for key in list1:
        result[key] = ""
    for i, keys in enumerate(result):
        result[keys] = list2[i]
    return(result)
list1 = [1, 2, 3, 4]
list2 = ["blue", "green", "pink", "yellow"]
print(create_dictionary(list1, list2))
```

---

problem/23.py

ประกาศฟังก์ชัน `create_dictionary(tuple1, tuple2)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```python
def create_dictionary(tuple1, tuple2):
    result = {}
    for key in tuple1:
        result[key] = ""
    for i, keys in enumerate(result):
        result[keys] = tuple2[i]
    return(result)
tuple1 = (1, 2, 3, 4)
tuple2 = ("ant", "cat", "dog", "cow")
print(create_dictionary(tuple1, tuple2))
```

---

problem/24.py

ประกาศฟังก์ชัน `store_student_info(student_data)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```python
def store_student_info(student_data):
    result = {}
    for student_id, name in student_data:
        result[student_id] = name
    return result
student_data = [("123456", "Alice" ), ("654321", "Bob"), ("112233", "Charlie")]
print(store_student_info(student_data))
```

---

problem/25.py

ประกาศฟังก์ชัน `store_student_scores(student_data)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```python
def store_student_scores(student_data):
    result = {}
    for std_id , name , score in student_data:
        result[std_id] = {}
        result[std_id]["name"] = name
        result[std_id]["score"] = score
```

```
    return(result)
student_data = [
("123456", "Alice", 85.5),
("654321", "Bob", 92.0),
("112233", "Charlie", 78.0)
]
print(store_student_scores(student_data))
```

---

problem/26.py

ประกาศฟังก์ชัน `search_countries_by_letter(country_data)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def search_countries_by_letter(country_data):
    result = []
    while True:
        letter = input("Please input 1 letter: ")
        if len(letter) != 1 :
            print("Please try again")
        else:
            break
    for code, country in country_data.items():
        for i , char in enumerate(country):
            if i == 0 and letter.lower() == char.lower():
                result.append(country)
                break
            else:
                break
    result.sort()
    return(result)
country_data = {
"+1": "United States",
"+44": "United Kingdom",
"+91": "India",
"+81": "Japan",
"+49": "Germany",
"+86": "China"
}
print(search_countries_by_letter(country_data))
```

---

problem/27.py

ประกาศฟังก์ชัน `build_set()` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def build_set():
    int_set = set()
    while len(int_set) < 5 :
```

```
        number = int(input(f"Please Enter number {len(int_set)+1} :"))
        if(number) not in (int_set):
            int_set.add(number)
        else:
            print("Please try again")
    return(int_set)
print(build_set())
```

---

problem/28.py

ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def check_membership(s: set, input_value: str) -> bool:
    input_string = input_value
    if input_value.isdigit():
        input_string = int(input_value)
    if input in s:
        return (True)
    else:
        return(False)
s = {1, 2, 3, 'a', 'e', 'i', 'o', 'u', "red", "green", "blue"}
input_value = input("Please input something: ")
print(check_membership(s, input_value))
```

---

problem/29.py

ประกาศฟังก์ชัน `set_operations(set1, set2)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และแสดงผลลัพธ์ทางหน้าจอ

```
def set_operations(set1, set2):
    return(sorted(set1.union(set2)) , sorted(set1.intersection(set2)))
set1 = {'a', 'e', 'i', 'o', 'u'}
set2 = {'h', 'e', 'l', 'l', 'o'}
print(set_operations(set1, set2))
```

---

problem/30.py

ประกาศฟังก์ชัน `calculate_set_differences(set1, set2)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_set_differences(set1, set2):
    return(set1.difference(set2), set2.difference(set1))
set1 = {'a', 'b', 'c'}
set2 = {'b', 'c', 'd'}
print(calculate_set_differences(set1, set2))
```

---

problem/31.py

ประกาศฟังก์ชัน `highest_sales_country(sales_data)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def highest_sales_country(sales_data):
    result = []
```

```
    max_amount = 0
    for country , amount in sales_data.items():
        if amount > max_amount:
            result = []
            result.append(country)
            result.append(amount)
            max_amount = amount
        if amount  == max_amount and country not in (result):
            result.append(country)
            result.append(amount)
    return(tuple(result))
sales_data = {
"Thailand": 1500,
"Laos": 1200,
"Vietnam": 1800,
"Japan": 1700,
"China": 2000,
"asd": 2000,
"h": 2000
}
print(highest_sales_country(sales_data))
```

---

problem/32.py

ประกาศฟังก์ชัน `calculate_median(numbers)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_median(numbers):
    numbers.sort()
    if(len(numbers) % 2 == 0):
        result = (numbers[len(numbers)//2] + numbers[(len(numbers)//2) -1]) / 2
    else:
        result = numbers[len(numbers)//2]
    print(numbers)
    return result
numbers = [8, 4, 7, 4, 6, 2, 10, 9, 3, 7, 1,10]
print(calculate_median(numbers))
```

---

problem/33.py

ประกาศฟังก์ชัน `calculate_median(provinces)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def calculate_median(provinces):
    values = sorted(provinces.values())
    n = len(values)
#   sorted_provinces = list(sorted(provinces.items(), key=lambda x : x[1]))
    if n % 2 == 1:
```

```
        median = values[n // 2]
    else:
        median = (values[n // 2 - 1] + values[n // 2]) / 2
    result = [(country, count) for country, count in provinces.items() if count == median]
    return result
provinces = {'Thailand':76, 'Laos':17, 'Vietnam':58, 'Japan':47,'Hang':46, 'China':23}
print(calculate_median(provinces))
```

---

problem/34.py

ประกาศฟังก์ชัน `print_rectangle_pattern(rows, columns)` ; วนลูปตามช่วงตัวเลขที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def print_rectangle_pattern(rows, columns):
    for i in range(rows):
        for j in range(columns):
            print("*", end="")
        print("\t")
rows = int(input("How many rows?: "))
columns = int(input("How many colums?: "))
print(print_rectangle_pattern(rows, columns))
```

---

problem/35.py

วนลูปตามช่วงตัวเลขที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def print_diamond_pattern(n: int) -> None:
    for i in range(1,n+1):
        for j in range(i):
            print("*", end="")
        print("\t")
    for i in range(n-1,0,-1):
        for j in range(i):
            print("*", end="")
        print("\t")
n = int(input("How many rows?: "))
print(print_diamond_pattern(n))
```

---

problem/36.py

วนลูปตามช่วงตัวเลขที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def print_diamond_pattern(n: int) -> None:
    median = n / 2
    for i in range(1,int(median)):
        # print(int(median)-i)
        for k in range(1,int(median)-i+1):
            print("*",end="")
        for k in range(int(median)-i,int(median)+i):
            print("-",end="")
```

```
        for k in range(int(median) +i, n):
            print("*",end="")
        print("")
    for i in range(int(median)-1,0,-1):
        # print(int(median)-i)
        for k in range(1,int(median)-i+1):
            print("*",end="")
        for k in range(int(median)-i,int(median)+i):
            print("-",end="")
        for k in range(int(median) +i, n):
            print("*",end="")
        print("")
n = int(input("How many rows?: "))
print_diamond_pattern(n)
```

---

problem/37.py

วนลูปตามช่วงตัวเลขที่กำหนด ; และรับอินพุตจากผู้ใช้ , วนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def print_number_pattern(rows: int) -> None:
    for i in range(1,rows+1):
        for j in range(0,rows-i):
            print("-",end="")
        for j in range(rows-i,rows):
            print(rows-j,end="")
        print("")
rows = int(input("How many rows?: "))
print_number_pattern(rows)
# def print_full_number_pyramid(rows: int) -> None:
#     for i in range(1, rows+1):
#         print(" " * (rows - i), end="")
#         for j in range(1, i+1):
#             print(j, end="")
#         for j in range(i-1, 0, -1):
#             print(j, end="")
#         print("")
# rows = int(input("How many rows?: "))
# print_full_number_pyramid(rows)
```

---

problem/38.py

ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และรับอินพุตจากผู้ใช้ , แสดงผลลัพธ์ทางหน้าจอ

```
def add(a: float, b: float) -> float:
    return(a+b)
def subtract(a: float, b: float) -> float:
    return(a-b)
def multiply(a: float, b: float) -> float:
```

```
    return(a*b)
def divide(a: float, b: float) -> float:
    if b == 0:
        return None
    else:
        return(a/b)
a = float(input("Enter your 1st number:"))
b = float(input("Enter your 2nd number:"))
result_add = add(a,b)
result_subtract = subtract(a,b)
result_multiply = multiply(a,b)
result_divide = divide(a,b)
print(f"{result_add}\t{result_subtract}\t{result_multiply}\t{result_divide}")
```

---

problem/39.py

ประกาศฟังก์ชัน `remove_word(sentence: str, word_to_remove: str)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def remove_word(sentence: str, word_to_remove: str):
    list_word = sentence.split()
    if word_to_remove not in list_word:
        return sentence
    list_word.remove(word_to_remove)
    join_sentence = ' '.join(list_word)
    return(join_sentence)
sentence = "Python is a popular programming language"
word_to_remove ="popularadad"
print(remove_word(sentence,word_to_remove))
```

---

problem/40.py

ประกาศฟังก์ชัน `calculate_profit(sales, costs)` ; คำนวณผลรวมของข้อมูล ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def calculate_profit(sales, costs):
    list_sale = list(sales)
    result = []
    for i in range(len(sales)):
        profit = sales[i] - costs[i]
        result.append(profit)
    sum_profit = sum(result)
    return(tuple(result),sum_profit)
sales = (10000.0, 15000.0, 20000.0, 25000.0, 30000.0)
costs = (7000.0, 8000.0,9000.0, 11000.0, 12000.0)
print(calculate_profit(sales, costs))
```

---

problem/41.py

ประกาศฟังก์ชัน `calculate_discounted_prices(prices, discount_percentage)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def calculate_discounted_prices(prices, discount_percentage):
    result = []
    for i in range(len(prices)):
        discount = prices[i] * (1 - discount_percentage/ 100)
        result.append(discount)
    return(result)
prices = [100.0, 250.0, 75.0]
discount_percentage = 20.0
print(calculate_discounted_prices(prices, discount_percentage))
```

---

problem/42.py

ประกาศฟังก์ชัน `calculate_jumps(d, s)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_jumps(d, s):
    if d == s :
        return(1)
    else:
        if d % s == 0:
            return(d //s)
        else:
            return(d //s +1)
d = 22
s = 7
print(calculate_jumps(d, s))
```

---

problem/43.py

ประกาศฟังก์ชัน `calculate_coins(amount)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_coins(amount):
    coin_10 = amount // 10
    remain = amount%10
    coin_5 = remain // 5
    remain = remain % 5
    coin_2 = remain // 2
    remain = remain % 2
    coin_1 = remain // 1
    return(coin_10,coin_5,coin_2,coin_1)
amount = 50
print(calculate_coins(amount))
```

---

problem/44.py

ประกาศฟังก์ชัน `calculate_investment_growth(principal,annual_rate, years)` ; คำนวณผลรวมของข้อมูล ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def calculate_investment_growth(principal,annual_rate, years):
    result = []
```

```
    for year in range(1,years+1):
        sum = principal * (1+annual_rate/100)**year
        result.append(f"{sum:.2f}")
    return(result)
principal = 1000
annual_rate = 5
years = 5
print(f"{calculate_investment_growth(principal,annual_rate, years)}",)
```

---

problem/45.py

ประกาศฟังก์ชัน `calculate_annual_return(initial_investment, final_investment, years)` ; ประมวลผลข้อมูลและเก็บผลในดิกชันนารี ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_annual_return(initial_investment, final_investment, years):
    annual_return = (final_investment/initial_investment)**(1/years)-1
    percent = annual_return * 100
    return(f"{percent:.2f}")
initial_investment = 1000
final_investment = 1500
years = 5
print(calculate_annual_return(initial_investment, final_investment, years))
```

---

problem/46.py

ประกาศฟังก์ชัน `calculate_perimeter(a, b, c)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_perimeter(a, b, c):
    return(a+b+c)
a = 3
b = 4
c = 5
print(calculate_perimeter(a, b, c))
```

---

problem/47.py

ประกาศฟังก์ชัน `calculate_rectangle_area(width,length)` ; และแสดงผลลัพธ์ทางหน้าจอ

```
def calculate_rectangle_area(width,length):
    return width * length
width = 5
length = 10
print(calculate_rectangle_area(width,length))
```

---

problem/48.py

คำนวณผลรวมของข้อมูล ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def cumulative_sum(n: int) -> int:
    result = 0
    for i in range(1,n+1):
        result+=i
    return result
```

```
n = 5

print(cumulative_sum(n))
```

---

problem/49.py

สลับตัวพิมพ์เล็ก/ใหญ่ในสตริง ; และแสดงผลลัพธ์ทางหน้าจอ

```
def toggle_case(s: str) -> str:

    return s.swapcase()

s = "Hello World!"

print(toggle_case(s))
```

---

problem/50.py

ประกาศฟังก์ชัน `find_words_of_length(words, length)` ; วนลูปรายการ/สตริง แล้วเก็บผลลัพธ์ลงลิสต์ ; และวนลูปประมวลผลข้อมูล , แสดงผลลัพธ์ทางหน้าจอ

```
def find_words_of_length(words, length):

    result = []

    for word in words:

        if len(word) == length:

            result.append(word)

    return result

words = ["apple", "banana", "cherry", "date", "fig", "grape"]

length = 6

print(find_words_of_length(words, length))
```