Set A

```
''' 1. Write a python program that simulates a basic calulator performing
addition subtraction multiplication and division.'''

# Addition fuction
def add(x, y):
    return x + y

# Subtraction funtion
def subtract(x, y):
    return x - y

# Multiplication
def multiply(x, y):

   return x * y

# Division
def divide(x, y):
    if y == 0:
        return "You can not divide by zero."
    else:
        return x / y
```

```
# Main program
print("Which Operation you want.")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")

# Taking  input from user that perform operation.
user_operation = input("Enter  your operation. (1/2/3/4): ")

# Get numbers from the user
num1 = float(input("Enter First Number: "))
num2 = float(input("Enter Second Number: "))

if user_operation == '1':
    print(f" The addition of number is : {num1} + {num2} = {add(num1, num2)}")
elif user_operation == '2':
    print(f" The subtraction of number is : {num1} - {num2} = {subtract(num1, num2)}")
elif user_operation == '3':
    print(f" The multiplication numbmer is : {num1} * {num2} = {multiply(num1, num2)}")
elif user_operation == '4':
    print(f" The division of number is : {num1} / {num2} = {divide(num1, num2)}")
else:
    print("You Enter Out Of Range.")
```

```
Which Operation you want.
1. Add
2. Subtract
3. Multiply
4. Divide
Enter  your operation. (1/2/3/4):  3
Enter First Number:  5
Enter Second Number:  6
 The multiplication numbmer is : 5.0 * 6.0 = 30.0
```

```python
''' 2. Write a Python program that converts a given decimal number to its binary equivalent'''

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

# Convert to binary using bin()
binary_result = bin(decimal_num)

# Output the binary equivalent
print(f"The binary equivalent of {decimal_num} is: {binary_result[2:]}")
```

```
Enter a decimal number:  100
The binary equivalent of 100 is: 1100100
```

```python
'''3. Write a Python program that asks for the user's age and then prints a message stating whether
the user is a minor, an adult, or a senior'''
# Taking Age from user
user_age = int(input("Enter Your Age: "))

# Checking the user's age category
if user_age <=18:
    print("You are a minor.")
elif user_age > 40:
    print("You are a senior.")
else:
    print("You are an adult.")
```

```
Enter Your Age:  46
You are a senior.
```

```python
'''4. Write a Python program to swap the values of two variables without using a third variable.'''
# Taking input from the user
a = int(input("Enter the value of a: "))
b = int(input("Enter the value of b: "))

# Swapping the values using tuple unpacking
a, b = b, a

# Displaying the swapped values
print("After swapping:")
print("Value of a:", a)
print("Value of b:", b)
```

```
Enter the value of a:  12
Enter the value of b:  23
After swapping:
Value of a: 23
Value of b: 12
```

```
 '''5. Write a Python program to print the first 10 numbers of the Fibonacci series.'''
# Function to print the first 10 numbers of Fibonacci series
def fibonacci_series(n):
# Starting values for the Fibonacci series
    a, b = 0, 1

# Printing the first 'n' Fibonacci numbers
    print("Fibonacci Series of 10 numbers.")
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b

# Calling the f
fibonacci_series(10)
```

```
Fibonacci Series of 10 numbers.
0 1 1 2 3 5 8 13 21 34
```

```
#6) Write a Python program to check if a given number is prime or not.
# define a flag variable
num = int(input("Enter the number: "))
flag = False
if num == 0 or num == 1:
    print(num, "is not a prime number")
elif num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break
# check if flag is True
    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

```
Enter the number:  13
13 is a prime number
```

```
'''7. Write a Python program that takes three numbers as input and checks if the third number is the
sum of the first two numbers using logical operators.'''

#Taking input from user
num1 = int(input("Enter your first number"))
num2 = int(input("Enter your second number"))
num3 = int(input("Enter your third number"))

#check if the third number is the sum of the first two numbers
if (num1 + num2) == num3:
    print(f"{num3} is the sum of two number {num1} and {num2}")
else:
    print(f"{num3} is not the sum of twor number {num1} and {num2}")
```

```
Enter your first number 23
Enter your second number 12
Enter your third number 45
45 is not the sum of twor number 23 and 12
```

```
'''
8) Write a Python program that imports a custom module you created with a function that returns
the factorial of a number
'''
# Importing the custom module
import module

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

# Using the factorial function from the custom module
result = module.factorial(num)

# Displaying the result
print(f"The factorial of {num} is: {result}")
```

```
Enter a number to calculate its factorial:  3
The factorial of 3 is: 6
```

```
# Function to calculate the factorial of a number
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result
```

```
'''9. Write a Python program that takes two numbers as input and performs division, handling the
case where the divisor is zero.
'''
def divide_num(num1, num2):
    try:
        # Attempt division
        result = num1 / num2
    except ZeroDivisionError:
        # Handle case where divisor is zero
        return "Error! Division by zero is not allowed."
    return result

# Taking input from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Performing the division
result = divide_num(num1, num2)

# Displaying the result
print("Result:", result)
```

```
Enter the first number:  10
Enter the second number:  0
Result: Error! Division by zero is not allowed.
```

```
'''10) Write a Python function that takes a list of numbers and returns the maximum value in the list.
'''
def find_max(numbers):
    # Check if the list is not empty
    if not numbers:
        return "The list is empty"

    max_value = numbers[0]  # Start by assuming the first element is the maximum

    # Iterate through the list to find the maximum value
    for num in numbers:
        if num > max_value:
            max_value = num  # Update max_value if a larger number is found

    return max_value

# Input: Asking user to enter numbers
user_input = input("Enter numbers separated by spaces: ")

# Convert the input string into a list of integers
numbers = list(map(int, user_input.split()))

# Find the maximum value using the function
result = find_max(numbers)
```

```
# Find the maximum value using the function
result = find_max(numbers)

# Displaying the result
print(f"The maximum value in the list is: {result}")
```

```
Enter numbers separated by spaces:  23 45 4
The maximum value in the list is: 45
```

```
'''
11) Write a Python function that takes a name and an optional age parameter and prints a greeting.
If the age is not provided, it should default to 25.
'''
def greet(name, age=25):
    print(f"Hello, {name}! You are {age} years old.")

# Example Usage
greet("Dhiraj")          # Using default age
greet("Ram", 30)        # Using provided age
```

```
Hello, Dhiraj! You are 25 years old.
Hello, Ram! You are 30 years old.
```

```
'''
12) Write a Python program to count the number of vowels in a given string
'''
# Function to count the vowels in a string
def count_vowels(s):
    vowels = "aeiouAEIOU"  # String containing both lowercase and uppercase vowels
    count = 0  # Variable to store the count of vowels

    # Iterate through each character in the string
    for char in s:
        if char in vowels:
            count += 1  # Increment the count if the character is a vowel

    return count

# Example usage
input_string = input("Enter a string: ")  # Take input from the user
vowel_count = count_vowels(input_string)  # Call the function to count vowels
print(f"The number of vowels in the string is: {vowel_count}")
```

```
Enter a string:  dhiraj rai solukhumbu
The number of vowels in the string is: 8
'''
```

```
13) Write a Python program that prints a multiplication table up to (numberx10).
'''
def print_multiplication_table(number):
    print(f"Multiplication Table for {number}:")
    for i in range(1, 11):
        result = number * i
        print(f"{number} x {i} = {result}")

# Taking input from the user
num = int(input("Enter a number to get its multiplication table: "))

# Calling the function to print the multiplication table
print_multiplication_table(num)
```

```
Enter a number to get its multiplication table:  5
Multiplication Table for 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
14) Write a Python program to print a right-angled triangle of '*' with a given number of rows. For
example, if the number of rows is 5, the output should be:
*
**
***
****
*****
'''
# Function to print a right-angled triangle of '*' with the given number of rows
def print_triangle(rows):
    for i in range(1, rows + 1):
        print('*' * i)

# Taking input from the user
num_rows = int(input("Enter the number of rows: "))

# Calling the function to print the triangle
print_triangle(num_rows)
```

```
Enter the number of rows:  5
*
**
***
****
*****
```

15) Write a Python program to print a pyramid of '*' with a given number of rows. For example, if the number of rows is 5, the output should be:

```
*
 ***
 *****
*******'''
```

```
# Function to print a pyramid of '*' with the given number of rows
def print_pyramid(rows):
    for i in range(1, rows + 1):
        # Print leading spaces for alignment
        print(' ' * (rows - i), end='')
        # Print stars
        print('*' * (2 * i - 1))
# Taking input from the user
num_rows = int(input("Enter the number of rows: "))

# Calling the function to print the pyramid
print_pyramid(num_rows)
```

```
Enter the number of rows:  5
    *
   ***
  *****
 *******
*********
```

Set B

```python
# 1. Given an integer x, return true if x is a palindrome, and false otherwise.
# Taking input from the user
number = int(input("Enter the number: "))

# Storing the original number in a variable for comparison
original_number = number

# Variable to store the reversed number
rev = 0

# Reversing the digits of the number
while number > 0:
    rev = (rev * 10) + number % 10
    number = number // 10

# Check if the original number is equal to its reversed version
if (original_number == rev):
    print("The number is a palindrome")
else:
    print("The number is not a palindrome")
```

```
Enter the number:  212
The number is a palindrome
```

```python
'''
 2.  Given a non-empty array of integers nums, every element appears twice except for one. Find
that single one.
'''

def singleNumber(nums):
    result = 0
    for num in nums:
        result ^= num  # XOR each number with result
    return result

# Example usage:
nums = [4, 1, 2, 1, 2, 4, 6]
print(singleNumber(nums))
```

```
6
```

```python
'''3.. Given an array of integers nums and an integer target, return indices of the two numbers such
that they add up to target. You may assume that each input would have exactly one solution,
and you may not use the same element twice. You can return the answer in any order.'''
```

```python
def twoSum(nums, target):
    # Dictionary to store the number and its index
    num_map = {}
    # Iterate through the list with index
    for index, num in enumerate(nums):
        # Calculate the complement (target - num)
        complement = target - num
        # Check if the complement exists in the map
        if complement in num_map:
            # If found, return the indices of the complement and the current number
            return [num_map[complement], index]

        # Otherwise, store the current number and its index in the map
        num_map[num] = index

    # In case there is no solution (although problem guarantees one solution)
    return None

# Example usage:
nums = [2, 7, 11, 15, 13]
target = 13

print(twoSum(nums, target))  # Output: [0, 1]
```

```
[0, 2]
```

```python
# 4 Write an algorithm to determine if a number n is happy.
def happy(n):
    # Set to store numbers that we have seen during the process
    seen = set()

    # Loop until n becomes 1 or we detect a cycle
    while n != 1:
        # Calculate the sum of the squares of the digits of n
        n = sum(int(digit) ** 2 for digit in str(n))

        # If n is already in the set, it means we are in a cycle, so it's not a happy number
        if n in seen:
            return False

        # Add the current number to the set of seen numbers
        seen.add(n)

    # If n becomes 1, then it's a happy number
    return True


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

```
# Check if the number is a happy number
if happy(n):
    print(f"{n} is a Happy Number!")
else:
    print(f"{n} is not a Happy Number.")
```

```
Enter a number:  34
34 is not a Happy Number.
```

```
'''
5...Given an integer array nums, return true if any value appears at least twice in the array, and
return false if every element is distinct.
'''
def containsDuplicate(nums):
    # Initialize an empty set to track the numbers we have seen
    seen = set()

    # Iterate through each number in the array
    for num in nums:
        # If the number is already in the set, we have found a duplicate
        if num in seen:
            return True
            return True
        # If the number is not in the set, add it to the set
        seen.add(num)

    # If no duplicates were found, return False
    return False


# Input: Get the list of numbers from the user
# The user should input the numbers separated by spaces, like "1 2 3 4 5"
input_string = input("Enter the numbers separated by spaces: ")

# Convert the input string to a list of integers
nums = list(map(int, input_string.split()))

# Check if the array contains duplicates
if containsDuplicate(nums):
    print("The array contains duplicate values.")
else:
    print("The array does not contain any duplicate values.")
```

```
Enter the numbers separated by spaces:  23 45 56 67 65
The array does not contain any duplicate values.
```