

*#1 Write a Python program that simulates a basic calculator, performing addition, subtraction, multiplication, and division.*

```
a = int(input('Enter a number:'))
b = int(input('Enter another number:'))
print("addition:", a + b)
print("subtraction:", a - b)
print("multiplication:", a * b)
print("division:", a // b)
```

```
Enter a number: 10
Enter another number: 2
```

```
addition: 12
subtraction: 8
multiplication: 20
division: 5
```

*#2 Write a Python program that converts a given decimal number to its binary equivalent.*

```
decimal_number = int(input('Enter decimal number:'))
binary_conversion = bin(decimal_number)
print(binary_conversion)
```

```
Enter decimal number: 15
0b1111
```

*#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.*

```
age = int(input('Enter your age:'))
if age <= 18:
    print("The user is minor.")
elif age <= 50:
    print("The user is adult.")
else:
    print("The user is senior.")
```

```
Enter your age: 25
The user is adult.
```

*#4 Write a Python program to swap the values of two variables without using a third variable.*

```
a = int(input('Enter first value'))
b = int(input('Enter second value'))
```

```
#swapping values
a = a + b
b = a - b
a = a - b
```

```
print(f"After swapping: a = {a}, b = {b}")
```

```
Enter first value 2
Enter second value 3
```

```
After swapping: a = 3, b = 2
```

*#5 Write a Python program to print the first 10 numbers of the Fibonacci series.*

```
num = 10
a = 0
b = 1
print(a)
print(b)
for i in range(2,num):
    c=a+b
    a=b
    b=c
    print(c)
```

```
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.*

*#taking input from the user*

```
num = int(input('Enter a number: '))
```

*#checking whether the number is equal to or less than 1*

```
if num <= 1:
    print("The given number is not a prime number")
if num > 1:
```

*#checking whether the number is divisible by other number or not by using for loop*

```
    for i in range (2, num):
```

```
        if num % i == 0:
            print("The given number is not a prime number")
            break
    else:
        print("The given number is a prime number")
```

Enter a number: 10

The given number is not 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*

```
a = int(input('Enter first number:'))
b = int(input('Enter second number:'))
c = int(input('Enter third number:'))
```

*#using logical operator*

```
if (a + b == c) and (c >= a or c >= b):
    print("c is the sum of a and b")
else:
    print("c is not the sum of a and b")
```

Enter first number: 2

Enter second number: 3

Enter third number: 5

c is the sum of a and b

*#8 Write a Python program that imports a custom module you created with a function that returns the factorial of a number.*

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

import my_math_module

num = int(input("Enter a number: "))
result = my_math_module.factorial(num)
print("Factorial of", num, "is", result)
```

Enter a number: 5

Factorial of 5 is 120

*#9 Write a Python program that takes two numbers as input and performs division, handling the case where the divisor is zero.*

```
try:
    #taking input from user
    x = int(input("Enter a number: "))
    y = int(input("Enter another number: "))
    #perform division
    div = x // y
    print("The result is:", div)
except ZeroDivisionError:
    #handle if the divisor is zero
    print("You cannot divide by zero.")
```

Enter a number: 10

Enter another number: 0

You cannot divide by zero.

*#10 Write a Python function that takes a list of numbers and returns the maximum value in the list.*

```
def find_max(numbers):
    if not numbers:
        raise ValueError("The input list is empty.")

    max_value = numbers[0]
    for num in numbers[1:]:
        if num > max_value:
            max_value = num
    return max_value
```

*# Get input from the user*

```
user_input = input("Enter numbers separated by spaces: ")
numbers = list(map(int, user_input.split()))
```

*# Find and print the maximum value*

```
max_number = find_max(numbers)
print(f"The maximum value in the list is: {max_number}")
```

Enter numbers separated by spaces: 1 2 3 4 5

The maximum value in the list is: 5

*#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.*

```
name = input('Enter your name: ')
age = input("Enter your age or skip: ")
```

```
if not age:
```

```
    age = 25  #Default age if not provided
else:
    age = int(age)

print(f"Hello, {name}! You are {age} years old.")
```

Enter your name: Niroj  
Enter your age or skip:

Hello, Niroj! You are 25 years old.

*#12 Write a Python program to count the number of vowels in a given string.*

```
a = input("Enter a string: ")
a.lower()
count = 0
list = ["a", "e", "i", "o", "u"]
for char in a:
    if char in list:
        count = count + 1

print("The number of vowel in the given string is ",count)
```

Enter a string: My name is Niroj

The number of vowel in the given string is 5

*#13 Write a Python program that prints a multiplication table up to (numberx10)*

```
num = int(input("Enter a number to generate its multiplication table: "))
print(f"Multiplication Table for {num}:")
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

Enter a number to generate its multiplication table: 9

Multiplication Table for 9:

```
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
```

*#14 Write a Python program to print a right-angled triangle of '\*' with a given number of rows.*

```
#taking input from user
num_rows = int(input("Enter the number of rows: "))
#loop each row
for i in range(1, num_rows + 1):
    #print * for each column in the row
    print('* ' * i)
```

Enter the number of rows: 5

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

*#15 Write a Python program to print a pyramid of '\*' with a given number of rows.*

```
rows = int(input("Enter the number of rows: "))
for i in range(1, rows + 1):
    #print spaces
    print(' ' * (rows - i), end='')
    # Print '*' for the pyramid
    print('*' * (2 * i - 1))
```

Enter the number of rows: 5

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

*#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: 121

The number is a palindrome

*#2 ) Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.*

```
nums = list(map(int, input("Enter the array of integers separated by
spaces: ").split()))
```

```
#Initialize result to 0
result = 0
```

```
#XOR all numbers in the array
for num in nums:
    result ^= num
```

```
print("The single number is:", result)
```

Enter the array of integers separated by spaces: 1 2 5 1 2 5 4 6 4

The single number is: 6

*#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.*

```
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:
num = [2, 7, 11, 15, 13]
target = 13

print(twoSum(num, target))

[0, 2]
```

*#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: 68

68 is 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: 26 25 85 65 58

The array does not contain any duplicate values.