

Ex No.05	User defined function	Reg. No: URK24CS9068
14.02.25		

**Q2. Write a python program to check whether a number is perfect or not using a function.**

**Aim:** To check whether a given number is a perfect number using a Python function.

**Objective:** To understand and apply the concept of proper divisors to identify perfect numbers using Python.

**Algorithm:**

- Step 1: Start
- Step 2: Define a function that accepts a number
- Step 3: Initialize a variable to store the sum of divisors
- Step 4: Loop from 1 to the number (excluding the number)
- Step 5: If a number divides the given number exactly, add it to the sum
- Step 6: After the loop, compare the sum with the original number
- Step 7: Print whether the number is perfect or not
- Step 8: End

**Program:**

```
print("D. Brian Gabriel")
print("URK24CS9068")
print("-----")
def is_perfect(num):
    sum_div = 0
    for i in range(1, num):
        if num % i == 0:
            sum_div += i
    if sum_div == num:
        print(num, "is a perfect number.")
    else:
        print(num, "is not a perfect number.")
number = int(input("Enter a number: "))
is_perfect(number)
```

**Output:**

```
D. Brian Gabriel
URK24CS9068
-----
28 is a perfect number.
```

**Result:** The program correctly checks whether a number is perfect using its proper divisors.

**Q4. Write a Python program to check whether a number is a Happy Number or not.**

**Aim:** To check whether a given number is a Happy Number using Python.

**Objective:** To understand and apply the logic of digit square sum iterations to determine Happy Numbers in Python.

**Algorithm:**

- Step 1: Start
- Step 2: Define a function to find the sum of squares of digits
- Step 3: Repeat the process until the number becomes 1 (happy) or loops (unhappy)
- Step 4: Use a set to track seen numbers to detect cycles
- Step 5: If 1 is reached, print Happy Number; if 4 or a repeat occurs, print Unhappy Number
- Step 6: End

**Program:**

```
print("D. Brian Gabriel")
print("URK24CS9068")
print("-----")
def is_happy_number(num):
    seen = set()
    while num != 1 and num not in seen:
        seen.add(num)
        num = sum(int(digit) ** 2 for digit in str(num))
    if num == 1:
        print("Happy Number")
    else:
        print("Unhappy Number")
number = int(input("Enter a number: "))
is_happy_number(number)
```

**Output:**

```
D. Brian Gabriel
URK24CS9068
-----
Unhappy Number
```

**Result:** The program correctly checks whether a number is a Happy Number by summing the squares of its digits repeatedly.

**Q6. Write a Python function to use the arbitrary function by getting a sequence of numbers as input and only print the divisors of the last input from the passing numbers.**

**Aim:** To write a Python program that accepts a list of numbers and prints all the divisors of the last number in the list.

**Objective:** To understand how to work with lists and loops in Python to find divisors of a number using user-defined functions.

**Algorithm:**

- Step 1: Start
- Step 2: Define a function that accepts a list of numbers
- Step 3: Extract the last number in the list using numbers[-1]
- Step 4: Loop from 1 to the last number
- Step 5: If the last number is divisible by the current loop index, print the index
- Step 6: Repeat for each input list
- Step 7: End

**Program:**

```
print("D. Brian Gabriel")
print("URK24CS9068")
print("-----")
def fun1(numbers):
    last = numbers[-1]
    print(f"{last} :", end=" ")
    for i in range(1, last + 1):
        if last % i == 0:
            print(i, end=" ")
    print()
fun1([6, 7, 8])
fun1([1, 2, 5, 7, 9])
fun1([1, 3, 5, 7, 9, 12])
```

**Output:**

```
D. Brian Gabriel
URK24CS9068
-----
8 : 1 2 4 8
9 : 1 3 9
12 : 1 2 3 4 6 12
```

**Result:** The program successfully prints the divisors of the last number in a list using a function that accepts a sequence of numbers.