Functions

function :- A function is a block of code which only runs

```
          when it is called
```

# Advantages

```
- Reusability of code
- It reduces the complexity of the code
- Easy debugging
```

Functions are divided into 2 types

```
    - built-in functions/Predefined  :- It can be developed by the developers
       ex:- print(),min(),max(),sum(),os(),math()..........etc
    - user defined functions :- It can be developed by the users

* Again user defined functions are divided into 4 types
        - with argument with return value
        - with argument without return value
        - without argument with return value
        - without argument without return value
```

def keyword is used to create functions

# Syntax for functions

def function_name(arguments/parameters): # function definition statements
function_name(arguments/parameters) # function calling

```python
# addition of 2 numbers
# with argument with return value
a,b = 6,9
def add1(a,b):
    return a+b
add1(a,b)
```

```
15
```

```python
# with arguments without return value
x,y = int(input()),int(input())
def add2(x,y):
    print(x+y)
add2(x,y)
```

```
 67
 89
```

```
156
```

```python
# without arguments with return value
c,d,e=int(input()),int(input()),int(input())
def add3(): # function definition
    return c+d+e
add3()  # function calling
```

```
 56
 67
 23
```

```
146
```

```python
# without arguments without return value
m,n = 9,5
def add4():
    print(m+n)
add4()
```

```
14
```

```python
add1(67,89)
```

```
156
```

```python
add2(89,89)
```

```
178
```

```python
def evenodd(n):
    if(n%2==0):
        print("even")
    else:
        print("odd")
n=int(input())
evenodd(n)
```

```
 7
```

```
odd
```

```
# perfect numbers using functions
# perfect number = 6
# 6 = 1 2 3 6 = 1+2+3=6 ,28
# sum of the factors = given input number

n = int(input())
def perfect(n):
    f_sum=0
    for i in range(1,n):
        if(n%i==0):  # 6%1(0)==0(T) 6%4()==0(T)
            f_sum=f_sum+i  # o+1=1 1+2=3 3+3=6
    #print("Factors sum=",f_sum)
    if(f_sum==n):
        return True
    else:
        return False
perfect(n)
```

```
 28
```

```
True
```

```
# perfect numbers betweeen range using functions
s,e=int(input()),int(input())
def perfect_range(s,e):  # i=1
    for i in range(s,e+1):
        if(perfect(i)==True):  # perfect(1)
            print(i,end=' ')  #
perfect_range(s,e)
```

```
 1
 1000
```

```
6 28 496
```

```
# prime number using function
n = int(input())
def prime(n):
    f_count=0
    for i in range(1,n+1):
        if(n%i==0):
            f_count=f_count+1
    if(f_count==2):
        return True
    else:
        return False
prime(n)
```

```
 8
```

```
False
```

```
# prime numbers betweeen range using functions
s,e = int(input()),int(input())
def prime_range(s,e):
    for i in range(s,e+1):
        if(prime(i)==True):
            print(i,end=' ')
prime_range(s,e)
```

```
 1
 100
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

# Task -01

```
- Write a python program to check whether a number is
strong number or not using functions
```