

Programs

Question 1:

Program to calculate the simple interest using function
(Take P,T,R from user input)

```
p = int(input("Enter the principal: "))
t = int(input("Enter the Time period: "))
r = int(input("Enter the rate of interest: "))

def simpleInterest(p,t,r):
    si = (p*t*r)/100
    return si

res = simpleInterest(p, t, r)
print(res)
```

Question 2:

Program to count the number of digits in a number

```
num = int(input("Enter the number: "))
count = 0
while num>0:
    count=count+1
    num=num//10

print("Total Number of digits: ",count)
```

Question 3:

Program to print first sum of first 10 even numbers

```
i = 2
count=1
sum=0
while count<=10:
    if i % 2==0:
        sum=sum+i
        count+=1
```

```
i=i+1  
print(sum)
```

Question 4:

Program to check whether a year is leap year or not.

condition:

The year should be divisible by 4.

If the year is divisible by 100, it should also be divisible by 400.

Ex:

Is 2100 divisible by 4? Yes, it is.

Is 2100 divisible by 100? Yes, it is.

Is 2100 divisible by 400? No, it is not.

Therefore, 2100 is not a leap year.

```
if year % 4 == 0:  
    if year % 100 == 0:  
        if year % 400 == 0:  
            print("The year is a leap year!")  
        else:  
            print("The year is not a leap year!")  
    else:  
        print("The year is a leap year!")  
else:  
    print("The year is not a leap year!")
```

```
def is_leap_year(year):  
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
        return True  
    else:  
        return False  
  
year = int(input("Enter a year: "))  
  
if is_leap_year(year):  
    print(year, "is a leap year.")  
else:  
    print(year, "is not a leap year.")
```

OUTPUT Questions

1

```
lang1 = "Python"
def progVersion():
    global ver
    ver = "3"
    return lang1+" "+ver

print(progVersion())
```

Python 3

2

```
num = 8
num &= 3

print(num)
```

0

```
num = 8      # Binary: 1000
3  = 3      # Binary: 0011
-----
num &= 3     # Result of bitwise AND: 0000
```

3

```
a = False
b = True
c = False

if a and b or c:
    print ("SILICON")
else:
    print ("silicon")
```

```
silicon
```

If the first operand is False, the result will be False regardless of the value of the second operand. This is known as short-circuiting behavior.

Similarly, in a logical OR operation, if the first operand is True, the result will be True without evaluating the second operand.

4

```
num = 8
num |= 3
print(num)
```

11

```
num = 8      # Binary: 1000
3  = 3      # Binary: 0011
-----
num |= 3     # Result of bitwise OR: 1011
```

5

```
name = "Ravi"
age = 23

print("Hello, my name is"+ name+ "and I am"+ age+ "years old.")
```

Error

6

```
num = 11
num |= 3

print(num)
```

```
11
```

```
num = 11      # Binary: 1011
3   = 3       # Binary: 0011
-----
num |= 3      # Result of bitwise OR: 1011 = 11
```

7

```
x = "Good"

def myfunc():
    x = "Better"

myfunc()
print("Python is: "+x)
```

```
Python is: Good
```

8

```
num = 6
num ^= 4

print(num)
```

```
2
```

```
num = 6      # Binary: 0110
4   = 4      # Binary: 0100
-----
num ^= 4     # Result of bitwise XOR: 0010 = 2
```

9

```
x = 1
while x<10:
    if(x%2!=0):
        print(x)
    x=x+1
```

```
1
3
5
7
9
```

10

```
def display(a, b):
    if a > b:
        print(a)
    elif a == b:
        print("Equal")
    else:
        print(b)
display(31, 44)
```

```
44
```

11

```
x = 50
def func(x):
    print('x is', x)
    x = 2
    print('x is', x)
func(x)
print('x is now', x)
```

```
x is 50
x is 2
x is now 50
```

12

```
def displayName(fname, lname):  
    return fname+" "+lname  
  
res = displayName("John", "Wick")  
print(res)
```

John Wick

13

```
fname = "John"  
def displayName():  
    global lname  
    lname= "Wick"  
    return lname+" "+fname  
  
res = displayName()  
print(res)
```

Wick John

14

```
num = 456  
while num!=0:  
    print(num)  
    num=num//10
```

456
45
4

15

```
x = bool(None)  
y = bool(()) # Empty Tuple  
z = bool(False)  
  
print(x, y, z)
```

False False False

16

```
def check(num):  
    return isinstance(num, float)  
  
print(check(5.4))  
print(check("5.4"))  
print(check(5))
```

True
False
False

17

```
i = 1  
while True:  
    if i%2 == 0:  
        break  
    print(i)  
    i += 2
```

Infinte loop

18

```
true = False  
while 1>0:  
    print(true)  
    break
```

False

19


```
def get_double(n):  
    n *= 2  
  
num = 5  
get_double(num)  
print(num)
```

5

20

```
a=10  
b=20  
a=a^b  
b=a^b  
a=a^b  
print(a,b)
```

20 10