WEEK 3: Assignment

Q1.Calculate the Salary using Python Programming

Read the question carefully and follow the input and output format.

Karen got salary for this month and she spends 20% of her salary for food, 10% of her salary for entertainment and 10% of her salary for travel.If she takes care of other shifts she will get 4% of the salary per day. Given her salary and the number of shifts she handled.Calculate how much she can save in her pocket after spending all these?

Input and Output Format : First line of input consists of an integer, salary. Next line correspond to the number of shifts. Output consist of an integer, which is saving.

1. Print "Salary too large" when salary is greater than 10000.
2. Print "Shifts too small" when the shift is less than 0.
3. Print "Salary too small" when the salary is less than 0. Include a function named calculateSal() to calcuate salary and print saving.

In [24]:
```python
def calculateSal():
    salary=int(input("Enter the salary="))
    if salary>10000:
        print("Salary too large")
    elif salary<0 :
        print("Salary too small")
    else:
        shifts=int(input("enter the no of shifts="))
        if shifts<0:
            print("Shift too small")
        else:
            savings=(salary*0.6)+(salary*0.04*shifts)
            return savings
```

In [25]:
```python
calculateSal()
```

```
Enter the salary=7000
enter the no of shifts=-1
Shift too small
```

In [ ]:
```python
Q2.write python code to read n as integer,from STDIN. For all non-negative integers i<n ,print i**2.

Example
The list of non-negative integers that are less than n=3 is[0,1,2]. Print the square of each number on a separate line.
0
1
4
```

In [17]:
```python
n=int(input("How many square number you want:"))
print("The square of each number:")
for i in range(n):
 print(i**2)
 print()
```

```
How many square number you want:5
The square of each number:
0

1

4

9

16
```

In [ ]:
```python
Q3.An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap

In the Gregorian calendar, three conditions are used to identify leap years:
The year can be evenly divided by 4, is a leap year, unless:
The year can be evenly divided by 100, it is NOT a leap year, unless:
The year is also evenly divisible by 400. Then it is a leap year.
This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. Source

Task
Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True, otherwise return False.

Note that the code stub provided reads from STDIN and passes arguments to the is_leap function. It is only necessary to complete the is_leap function.
```

In [18]:
```python
def is_leap(year):
    return bool((year%4==0 and year%100!=0) or (year%400==0))
year=int(input("Enter The Year"))
is_leap(year)
```

```
Enter The Year1998
```

Out[18]:
```
False
```

In [ ]: