Python Introduction Lab
Assignment 1

Name: Jaskirat Singh

Roll number: 05413202717

Section: CSE-1 (8th Semester)

GitHub Repository:

Q1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. Extras:

1. Add on to the previous program by asking the user for another number and printing out that many copies of the previous message.
2. Print out that many copies of the previous message on separate lines. (Hint: the string "\n is the same as pressing the ENTER button)

Sol:

Pseudocode:

```
name = input('Enter your name: ')
age = input('Enter your age: ')

year = 2021 + 100 - int(age)

result = name + ' will be 100 years old in ' + str(year)+'.' + '\n'
print(result)

# Extras:
print('Extras:')

repeat_msg = input('Enter number of copies of the previous message: ')
print(result * int(repeat_msg))
```

Program screenshot and Output:

```
In [1]: name = input('Enter your name: ')
        age = input('Enter your age: ')

        year = 2021 + 100 - int(age)
        result = name + ' will be 100 years old in ' + str(year)+'.' + '\n'
        print(result)

        # Extras:
        print('Extras:')
        repeat msg = input('Enter number of copies of the previous message: ')

        print(esult * int(repeat msg))

        Enter your name: Japneet Singh
        Enter your age: 22
        Japneet Singh will be 100 years old in 3099.

        Extras:
        Enter number of cop1es of     the previous message: 10
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
        Japneet Singh will be 100 years old in 3099.
```

Q2. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.

Extras:

1. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.

2. Write this in one line of Python.

3. Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.

Sol:

Pseudocode:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

for element in a:
    if( int(element) < 5 ):
        print(str(element)+"\n")

print('Extras:')
#Extras:

new_list = [ ele for ele in a if ele < 5 ]

print('New List is : ' + str(new_list))

num = input('Enter a number to print all the numbers that are smaller than that number: ')

print([ ele for ele in a if ele < int(num) ])
```

Program screenshot and Output:

In [2]:
```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

for element in a:
    if( int(element) < 5 ):
        print(str(element)+"\n")

print('Extras:')
#Extras:

new_list = [ ele for ele in a if ele < 5 ]
print('New List is : ' + str(new_list))

num = input('Enter a number to print all the numbers that are smaller than that number: ')
print([ ele for ele in a if ele < int(num) ])
```

```
1

1

2

3

Extras:
New List is : [1, 1, 2, 3]
Enter a number to print all the numbers that are smaller than that number: 21
[1, 1, 2, 3, 5, 8, 13]
```

Q3. Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate.(Hint: The Fibonacci seqence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, ...).

Sol:

Pseudocode:

```
def fibonacci(nterms):
    n1, n2 = 0, 1
    count = 0
    if nterms <= 0:
        print("Please enter a positive integer")
    elif nterms == 1:
        print("Fibonacci sequence upto",nterms,":")
        print(n1)
    else:
        print("Fibonacci sequence:")
        while count < nterms:
            print(n1)
            nth = n1 + n2
            # update values
            n1 = n2
            n2 = nth
            count += 1
num_lmt = input('Enter the length of Fibonacci series: ')
fibonacci(int(num_lmt))
```

Program screenshot and Output:

In [3]:
```
def fibonacci(nterms):
    n1, n2 = 0, 1
    count = 0
    if nterms <= 0:
        print("Please enter a positive integer")
    elif nterms == 1:
        print("Fibonacci sequence upto",nterms,":")
        print(n1)
    else:
        print("Fibonacci sequence:")
        while count < nterms:
            print(n1)
            nth = n1 + n2
            n1 = n2
            n2 = nth
            count += 1
num_lmt = input('Enter the length of Fibonacci series: ')
fibonacci(int(num_lmt))
```

```
Enter the length of Fibonacci series: 14
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
55
89
144
233
```

Q4. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

Extras:

● Write two different functions to do this - one using a loop and constructing a list, and another using sets.

Sol:

Pseudocode:

```
def duplicate_f1(x):
    new_list = []
    for i in x:
        if i not in new_list:
            new_list.append(i)
    return new_list

def duplicate_f2(x):
    return list(set(x))

user_list=[]
limit = int(input('Enter the length of the list: '))
for i in range(limit):
    ele = int(input('Enter '+str(i)+'th element:'))
    user_list.append(ele)

print(user_list)
print ('Function using loop: '+str(duplicate_f1(user_list)))
print ('Function using set: '+str(duplicate_f2(user_list)))
```

Program screenshot and Output:

In [4]:
```
def duplicate_f1(x):
    new_list = []
    for i in x:
        if i not in new_list:
            new_list.append(i)
    return new_list

def duplicate_f2(x):
    return list(set(x))

user_list=[]
limit = int(input('Enter the length of the list: '))
for i in range(limit):
    ele = int(input('Enter '+str(i)+'th element:'))
    user_list.append(ele)

print(user_list)
print ('Function using loop: '+str(duplicate_f1(user_list)))
print ('Function using set: '+str(duplicate_f2(user_list)))
```

```
Enter the length of the list: 6
Enter 0th element:0
Enter 1th element:1
Enter 2th element:2
Enter 3th element:3
Enter 4th element:2
Enter 5th element:1
[0, 1, 2, 3, 2, 1]
Function using loop: [0, 1, 2, 3]
Function using set: [0, 1, 2, 3]
```

Q5. Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.). Use functions.

Sol:

Pseudocode:

```
def prime(num):
    flag =  False
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                flag = True
                break
    if flag:
        print(num, "is not a prime number.")
    else:
        print(num, "is a prime number.")
num = int(input('Enter the number to check whether it is prime or not: '))
prime(num)
```

Program screenshot and Output:

In [5]:
```
def prime(num):
    flag = False
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                flag = True
                break
    if flag:
        print(num, "is not a prime number.")
    else:
        print(num, "is a prime number.")
num = int(input('Enter the number to check whether it is prime or not: '))
prime(num)
```

```
Enter the number to check whether it is prime or not: 19
19 is a prime number.
```