Joshua Wiseman HW1 Question 6

In [1]:

```
# A
def sum_two_numbers(a, b):
    return a + b # return sum of a and b

# A Tests
print(sum_two_numbers(1,2))
print(sum_two_numbers(-1,2))
print(sum_two_numbers(10,11))
print(sum_two_numbers(-2,-2))
```

```
3
1
21
-4
```

In [2]:

```
# B
def seconds_to_hms_format(seconds):
    hours = seconds // 3600 # find the hours through integer division
    minutes = (seconds % 3600) // 60 # find mintues through remainder of hours and then divide that through int
    seconds = seconds % 60 # the remainder of seconds
    return f"{int(hours):02}:{int(minutes):02}:{int(seconds):02}" # output in the hh:mm:ss format

# B Tests
print(seconds_to_hms_format(1))
print(seconds_to_hms_format(60))
print(seconds_to_hms_format(3660))
print(seconds_to_hms_format(3661))
```

```
00:00:01
00:01:00
01:01:00
01:01:01
```

In [3]:

```
# C
def print_even_index_from_string(string):
    output = "" # create output variable
    for i in range(len(string)): # loop throught the string
        if i % 2 == 0: # find if the index is even
            output+=string[i] # if so, append it to the output string
    print(output) # After the loop print out the output

# C Tests
print_even_index_from_string("Hello")
print_even_index_from_string("World")
print_even_index_from_string("Big Dog")
```

```
Hlo
Wrd
BgDg
```

In [4]:

```
# D
def remove_duplicates_from_list(arr):
    output_arr = [] # create output_arr list
    hash_set = set([]) # Use set for O(1) time operations, changes the remove duplicates from O(n^2) to O(n)
    for element in arr: # loop through all variables in arr
        if element not in hash_set: # if the element is not in the set
            output_arr.append(element) # then add it to the output_arr
            hash_set.add(element) # and the set too
    return output_arr # return the output_arr

# D Tests
print(remove_duplicates_from_list(["a", "a", "ab", "b", "c", "c", "c"]))
print(remove_duplicates_from_list([1]))
print(remove_duplicates_from_list([]))
```

```
['a', 'ab', 'b', 'c']
[1]
[]
```

In [5]:

```
# E
def fibonacci(n):
    fib = [0] * (n+3) # init the fib array which will hold fibanacci numbers in a dynamic programming changing
    fib[0] = 0 # init fib(0)
    fib[1] = 1 # init fib(1)
    fib[2] = 1 # init fib(2)
    for i in range(3,n+1): # loop to the given number n (inclusive) from 3 (because 0 to 2 is calculated)
        fib[i] = fib[i-1] + fib[i-2] # Through the previously calculated values add this to the current value
    return fib[n] # return the nth number of the fibanacci sequence
```

```
# E Tests
print(fibonacci(0))
print(fibonacci(1))
print(fibonacci(6))
print(fibonacci(13))
print(fibonacci(35))
```

```
0
1
8
233
9227465
```

The following code was for assertion testing in a python file and would also read in user input.

This doesn't work in a Jupiter Notebook

In [7]:

```
# Assertion Tests
if __name__ == '__main__':
    # A Tests
    assert sum_two_numbers(1,2) == 3, "Test A1 Failed"
    assert sum_two_numbers(-1,2) == 1, "Test A2 Failed"
    assert sum_two_numbers(10,11) == 21, "Test A3 Failed"
    assert sum_two_numbers(-2,-2) == -4, "Test A4 Failed"

    # B Tests
    assert seconds_to_hms_format(1) == "00:00:01", "Test B1 Failed"
    assert seconds_to_hms_format(60) == "00:01:00", "Test B2 Failed"
    assert seconds_to_hms_format(3660) == "01:01:00", "Test B3 Failed"
    assert seconds_to_hms_format(3661) == "01:01:01", "Test B4 Failed"

    # C Tests
    print_even_index_from_string("Hello")
    print_even_index_from_string("World")
    print_even_index_from_string("Big Dog")
    print()

    # D Tests
    assert remove_duplicates_from_list(["a", "a", "ab", "b", "c", "c", "c"]) == ["a", "ab", "b", "c"], "Test D1
    assert remove_duplicates_from_list([1]) == [1], "Test D2 Failed"
    assert remove_duplicates_from_list([]) == [], "Test D3 Failed"

    # E Tests
    assert fibonacci(0) == 0, "Test E1 Failed"
    assert fibonacci(1) == 1, "Test E2 Failed"
    assert fibonacci(6) == 8, "Test E3 Failed"
    assert fibonacci(13) == 233, "Test E4 Failed"
    assert fibonacci(35) == 9227465, "Test E5 Failed"



    # Get User input and enter them into the previously tested functions

    # # Input for A
    # user_input = input("Problem A)   Enter in 2 numbers (space between each number)\n")
    # n1, n2 = user_input.split()
    # n1 = int(n1)
    # n2 = int(n2)
    # print(f"The sum of {n1} and {n2} numbers is", sum_two_numbers(n1, n2),'\n')

    # # Input for B
    # user_input = input("Problem B)   Enter in number of seconds\n")
    # seconds = int(user_input)
    # print(f"The hh:mm:ss format of {seconds} is", seconds_to_hms_format(seconds),'\n')

    # # Input for C
    # user_input = input("Problem C)   Enter in a string\n")
    # print_even_index_from_string(user_input)
    # print()

    # # Input for D
    # user_input = input("Problem D)   Enter in a list (each value separated by a space)\n")
    # arr = [x for x in user_input.split(" ")]
    # print(f"List content after duplicate removal from {arr} is", remove_duplicates_from_list(arr),'\n')

    # # Input for E
    # user_input = input("Problem E)   Enter in a number for fibonacci\n")
    # fib_num = int(user_input)
    # print(f"Fibonacci sequence for {fib_num} is", fibonacci(fib_num))
```

```
Hlo
Wrd
BgDg
```