CS 381 Homework 1
Goal: Research and Explore a new programming language

1. Select a programming language you have not used (NOT Haskell or Prolog). Give a brief history of this language. Who developed the language? When? Where?

   The programming language Python was created by Guido van Rossum with Python's first release on February 20, 1991.

2. Give a description of the language. What is its paradigm(s)? How is it categorized? Describe some of the features of the language. What is typing method/discipline?

   Python is a high-level language and has an emphasis on code readability. It consists of garbage collection and is dynamically typed. Moreover, it can support functional, structured, and object oriented programming. Python is one of the most popular languages and this can be seen by its expansive library

3. Is the language compiled or interpreted? On what platforms is the language available? Is the language standardized? Are there different implementations?

   The language is available on all platforms such as Linux, Windows, and macOS. Python is an interpreted language as its source code is converted into bytecode and executed. Its code is not required to be linked and compiled unlike languages such as C or C++. Standardized languages consists of a formal, approved standard written by an organization. Python is an open source language and thus not formally standardized. Finally, the known implementations of Python are IronPython, Jython, and PyPY.

4. Give examples of at least two control structures in the language (ie. for-loop or if-statement). Explain.

   (a) if-statement

```
main.py
1  x = 2
2  if (x < 5):
3    print("x is less than 5")
4
5  elif(x > 5):
6    print("x is more than 5")
7
```

In this case, the program dynamically creates a variable of x setting it equal to 2.
It is ran through the control if statement to assess whether the variable is less than 5.
If it is, print that x is less than 5.
Else, we assess in an else if statement whether the variable x is less than 5
If it is, print that x is more than 5.

(b) for-loop

```
main.py
1  for x in range(5):
2    print(x)
```

For a variable that is within the range of x,
Print the iterating numerical values that x becomes
In this case, it will iterate through 0, 1, 2, 3, 4
As it is a range of 5 includes 0 to 4.

5. To complete questions 6- 8 you will need to find an online complier/interpreter to run code in your language,
for example https://onecompiler.com/.
Alternatively you can install the software or use the school's server.
What will you be using?

   Note: You can use code you find online to answer questions 6-8 just include this information in the reference list for question 9.

6. Write a "Hello World" program written in the language. Describe how it works. Provide a screenshot of the execution of the program.

```
main.py
1  print("Hello World!")
```

Python takes any number of parameters given in the print() funciton and prints them all out in a single line
The items are converted into text and a newline character is placed at the end of the string.

7. Write a program to compute the first n Fibonacci numbers where the user is prompted for n (if possible). Describe how the code works. Is the program iterative or recursive? Provide a screenshot of the execution of the program.

main.py + 3z5ay9tw8

```
# Program to display the Fibonacci sequence up to n-th term

nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
# generate fibonacci sequence
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

   (a) Variable n terms is casted as an int for user input for number of terms

   (b) The first two terms are defined and set equal to 0 and 1

   (c) A counter is defined and set to 0

   (d) if the number of terms are less than or equal to 0

      i. The user inputted a negative integer and reprompt

   (e) Else if if the numebr of terms is just 1

      i. Print the first fibonacci sequence

   (f) else

      i. while the counter is less than the number of terms inputted by user

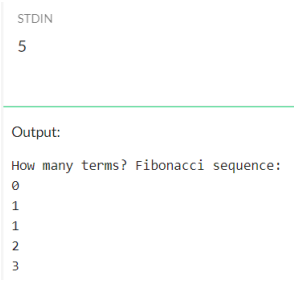
         A. print the first value

B. have a new variable be the sum of the first and second values of the sequence

C. set the first value as the second

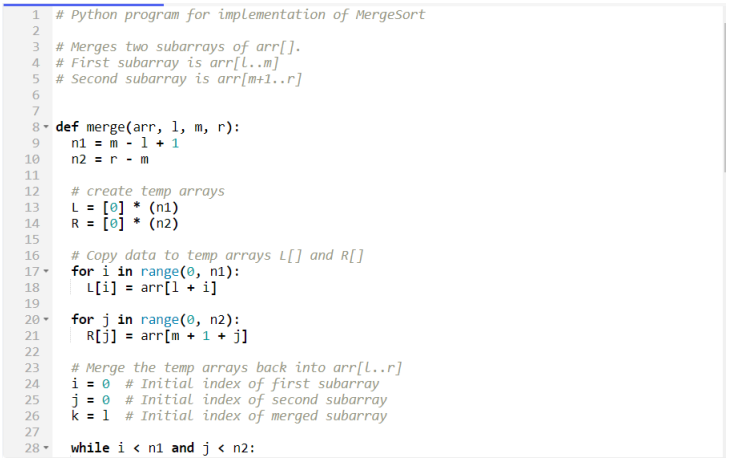
D. set the second as the sum

E. increase the counter



The program shows to be an iterative implementation of the fibonacci sequence.

8. Write a program to sort a list of integers. You can use any sorting algorithm, but do not use library functions. Describe how the code works. Provide a screenshot of the execution of the program.



```python
# Python program for implementation of MergeSort

# Merges two subarrays of arr[].
# First subarray is arr[l..m]
# Second subarray is arr[m+1..r]


def merge(arr, l, m, r):
  n1 = m - l + 1
  n2 = r - m

  # create temp arrays
  L = [0] * (n1)
  R = [0] * (n2)

  # Copy data to temp arrays L[] and R[]
  for i in range(0, n1):
    L[i] = arr[l + i]

  for j in range(0, n2):
    R[j] = arr[m + 1 + j]

  # Merge the temp arrays back into arr[l..r]
  i = 0  # Initial index of first subarray
  j = 0  # Initial index of second subarray
  k = l  # Initial index of merged subarray

  while i < n1 and j < n2:
```

main.py + 3z5ay9tw8

```
  while i < n1 and j < n2:
    if L[i] <= R[j]:
      arr[k] = L[i]
      i += 1
    else:
      arr[k] = R[j]
      j += 1
    k += 1

  # Copy the remaining elements of L[], if there
  # are any
  while i < n1:
    arr[k] = L[i]
    i += 1
    k += 1

  # Copy the remaining elements of R[], if there
  # are any
  while j < n2:
    arr[k] = R[j]
    j += 1
    k += 1

# l is for left index and r is right index of the
# sub-array of arr to be sorted
def mergeSort(arr, l, r):
  if l < r:


    # Same as (l+r)//2, but avoids overflow for
    # large l and h
    m = l+(r-l)//2

    # Sort first and second halves
    mergeSort(arr, l, m)
    mergeSort(arr, m+1, r)
    merge(arr, l, m, r)


# Driver code to test above
arr = [12, 11, 13, 5, 6, 7]
n = len(arr)
print("Given array is")
for i in range(n):
  print("%d" % arr[i],end=" ")

mergeSort(arr, 0, n-1)
print("\n\nSorted array is")
for i in range(n):
  print("%d" % arr[i],end=" ")
```

The implementation uses a merge sort algorithm in Python. Merge sort is a divide and conquer algorithm in which case
The array that is to be sorted is havled into two different sub-arrays.
This process is repeated for those two halves until all sub-arrays are sorted.
Ultimately, all sub-arrays are merged. This algorithm has a time complexity of $O(n * \log(n))$

Output:

```
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13
```

9. List at least three references you used for this assignment. Include any sites that you used to obtain code

   Python history:

https://pythoninstitute.org

Python control statements:
https://www.w3schools.com/python/python$_{f}or_{l}oops.asp$

Python fibonacci sequence code:
https://www.programiz.com/python-programming/examples/fibonacci-sequence

Python merge sort algorithm:
https://www.geeksforgeeks.org/python-program-for-merge-sort/

10. Would you like to learn more about this language? Would anticipate using this language in the future? Explain

    Yes, python is a popular language and I would recommend it to people who would like to get their toes into coding or computer science. It is a high-level language and it is easy to understand with its simply readability.