Data structure and algorithms

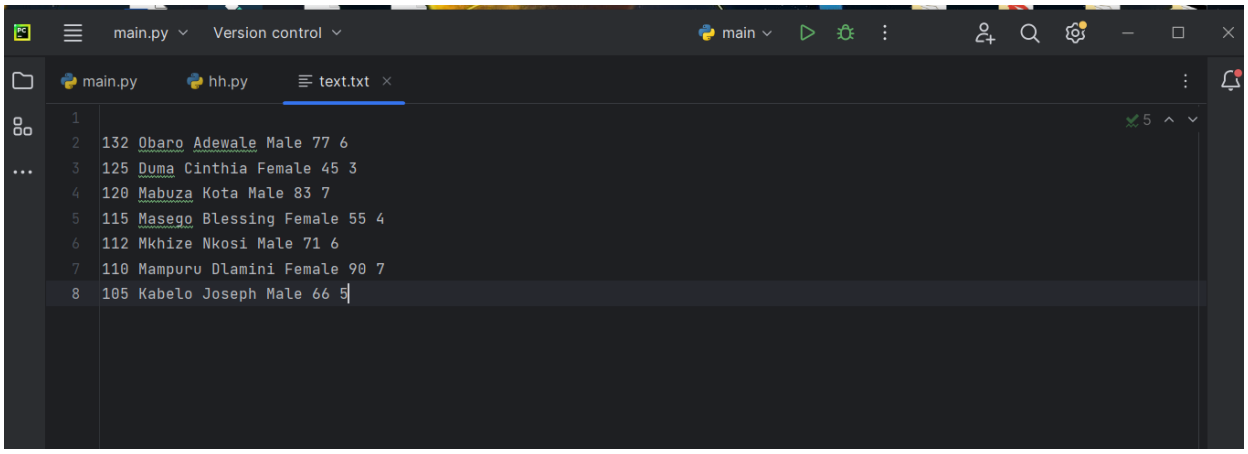
Name:	***Glodi Mukadi***
Surname:	***Kalombo***
Student number :	***D1LXT16D9***
Campus:	***Tyger valley campus***
Module:	***ITDPA2-44***
Due date:	***14 November 2023***
Marks :	***100***

QUESTION 1

1.1)

(Anon., 2023)

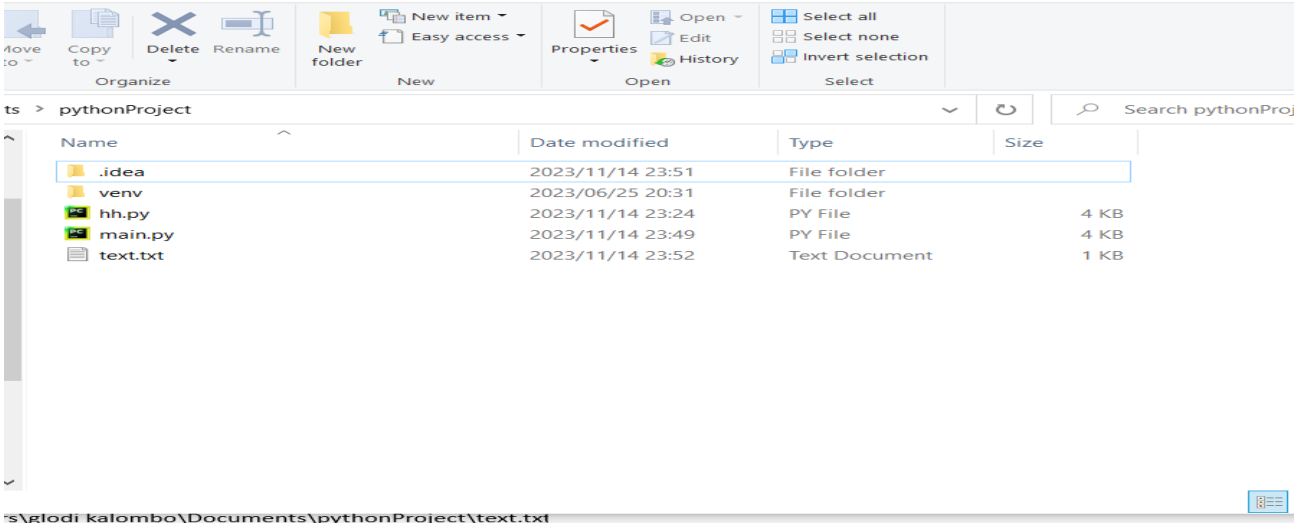
TEXT FILE



FILE PATH OF THE TEXT FILE IS HAS FOLLOW

: C:\Users\glodi kalombo\Documents\pythonProject\text.txt

HERE THE TEXT FILE IN EXPLORA IN THE SAME DIRECTORY HA THE MAIN.PY FILE




```
#1.6 Create a class "StudentData ()" to store all the
individual student data.
class Student:
```

```
    def __init__(self, num, last, first, gender, mark, level):
        self.num = num
        self.last = last
        self.first = first
        self.gender = gender
        self.mark = mark
        self.level = level

# 1.1 Create a class "StudentFileReader ()" for retrieving the
student data from the given text file.

class StudentReader:
    def __init__(self, filename):
        self.filename = filename
        self.file = None

    # 1.4 Create a function "getAll ()" that will retrieve all
the student data and store them in a list.

    def read_all(self):
        students = []
        self.open_file()  # tis  will Open the text file

        if self.file is not None:
            for line in self.file:
                data = line.split()
                if len(data) == 6:  # this will  Ensure there
is enough elements in the line
                    mark = ''.join(c for c in data[4] if
c.isdigit())
                    level = ''.join(c for c in data[5] if
c.isdigit())
                    students.append(Student(data[0], data[1],
data[2], data[3], int(mark), int(level)))
                else:
                    print(f"Invalid data format in line:
{line}")

            self.close_file()  # this   close   the
connection  to  the file

        return students
    # 1.2 Create a function "open ()" to open a connection to
the text file.


    def open_file(self):
        try:
            self.file = open(self.filename, 'r')
        except FileNotFoundError:
            print(f"File '{self.filename}' not found.")
```

```
        except Exception as e:
            print(f"Error opening the file: {e}") # error
handling
            self.file = None  # Set file to None to handle the
error


    # 1.3 Create a function "close ()" to close a connection
to the text file.

    def close_file(self):
        if self.file is not None and not self.file.closed:
            self.file.close()

    # 1.5 Create a function "getData ()" that will retrieve
the next student data from the text file.
    # The function  will  return None when there is no
additional data to be retrieved from  the  text  file
    def read_data(self):
        line = self.file.readline()
        if line:
            data = line.split()
            return Student(data[0], data[1], data[2], data[3],
int(data[4]), int(data[5]))
        else:
            return None

# 1.7 Create a function "

def display_data(students):

print(f"{'studNum':<10}{'Lastname':<10}{'Firstname':<10}{'Gend
er':<10}{'Mark':<10}{'Level':<10}")
    print("-----------------------------------------------------
------------")
    for student in students:

print(f"{student.num:<10}{student.last:<10}{student.first:<10}
{student.gender:<10}"
              f"{student.mark:<10}level {student.level:<10}")
        print("-----------------------------------------------
----------------")

 #1.8 Create a function "main ()" to execute the entire
program.
def execute_program():
    filename = "text.txt"  #this  is  the  file  path  of
my  text filee
    reader = StudentReader(filename)

    student_list = reader.read_all()  # this  fetches  all
```

```
the  data  from  the  text file

    # this   sort  the  list   by  student  number
    student_list.sort(key=lambda x: x.num)

    # this Print the student data
    display_data(student_list)

# this  Call the function to execute the program
if __name__ == "__main__":
    execute_program()
```

(Anon., 2023)

PROOF OF output :

```
"C:\Users\glodi kalombo\Documents\pythonProject\venv\Scripts\python.exe" "C:\Users\glodi kalombo\Documents\pythonPr
Invalid data format in line:

studNum   Lastname  Firstname Gender    Mark      Level
-------------------------------------------------------------
105       Kabelo    Joseph    Male      66        level 5
-------------------------------------------------------------
110       Mampuru   Dlamini   Female    90        level 7
-------------------------------------------------------------
112       Mkhize    Nkosi     Male      71        level 6
-------------------------------------------------------------
115       Masego    Blessing  Female    55        level 4
-------------------------------------------------------------
120       Mabuza    Kota      Male      83        level 7
-------------------------------------------------------------
125       Duma      Cinthia   Female    45        level 3
-------------------------------------------------------------
132       Obaro     Adewale   Male      77        level 6
-------------------------------------------------------------
Could not find platform independent libraries <prefix>

Process finished with exit code 0
```

QUESTION 2

(Anon., 2023)

```
#QUESTION   2.1==========================================
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

def create_array(*args):
    return list(args)


#first  row  of  array  1  to  12
array1 = create_array(2, 4, 6, 8, 10, 12)

#  the second array 20 to  40
array2 = create_array(23, 29, 31, 37)


# QUESTION  2.2)
==================================================

def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left = arr[:mid]
    right = arr[mid:]

    left = merge_sort(left)
    right = merge_sort(right)

    return merge(left, right)

def merge(left, right):
    result = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1

    result.extend(left[i:])
```

```
    result.extend(right[j:])
    return result

# Merge and sort the arrays
Merged_Arrays = merge_sort(array1 + array2)
print(" Link list ")
print(Merged_Arrays)
#QUESTION  2.3) A)-------------------------------------------
------
#==========================================================
=====
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

def Arr_LL(arr):
    if not arr:
        return None

    head = Node(arr[0])
    current = head

    for i in range(1, len(arr)):
        current.next = Node(arr[i])
        current = current.next

    return head

def display_LL(node):
    while node:
        print(node.data, end=" -> ")
        node = node.next
    print("None")

def Reverse_LL(node):
    prev = None
    current = node

    while current:
        next_node = current.next
        current.next = prev
        prev = current
        current = next_node

    return prev

# QUESTION 2.3) A) ---------------------------------------------
-------------------
# Creating a linked list from the merged and sorted array
linked_list = Arr_LL(Merged_Arrays)
```

```
print("Linked List in  same  order  same order  as  the merge
sorted  arrays:")
display_LL(linked_list)

# QUESTION 2.3) B) ----------------------------------------------
--------

# Reversing the linked list
Reversed_LL = Reverse_LL(linked_list)
print("Reversed Linked List:")
display_LL(Reversed_LL)

#QUESTION 2.3 C ------------------------------------------------
-----

def Insertion(node, value_to_insert, reference_value):
    current = node

    while current:
        if current.data == reference_value:
            new_node = Node(value_to_insert)
            new_node.next = current.next
            current.next = new_node
            break

        current = current.next

    return node

# Inserting 45 after the last value in the reversed linked
list
List_Insertion = Insertion(Reversed_LL, 45, Reversed_LL.data)
print("Linked List after Insertion:")
display_LL(List_Insertion)
```

(Anon., 2023)

PROOF , OF THE OUTPUT

```
Run    main ×
"C:\Users\glodi kalombo\Documents\classof15august\pythonProject5\Scripts\python.exe" "C:\Users\
Could not find platform independent libraries <prefix>
 Link list
[2, 4, 6, 8, 10, 12, 23, 29, 31, 37]
Linked List in  same  order  same order  as  the merge sorted  arrays:
2 -> 4 -> 6 -> 8 -> 10 -> 12 -> 23 -> 29 -> 31 -> 37 -> None
Reversed Linked List:
37 -> 31 -> 29 -> 23 -> 12 -> 10 -> 8 -> 6 -> 4 -> 2 -> None
Linked List after Insertion:
37 -> 45 -> 31 -> 29 -> 23 -> 12 -> 10 -> 8 -> 6 -> 4 -> 2 -> None

Process finished with exit code 0
```

QUESTION 3

3.1) a)

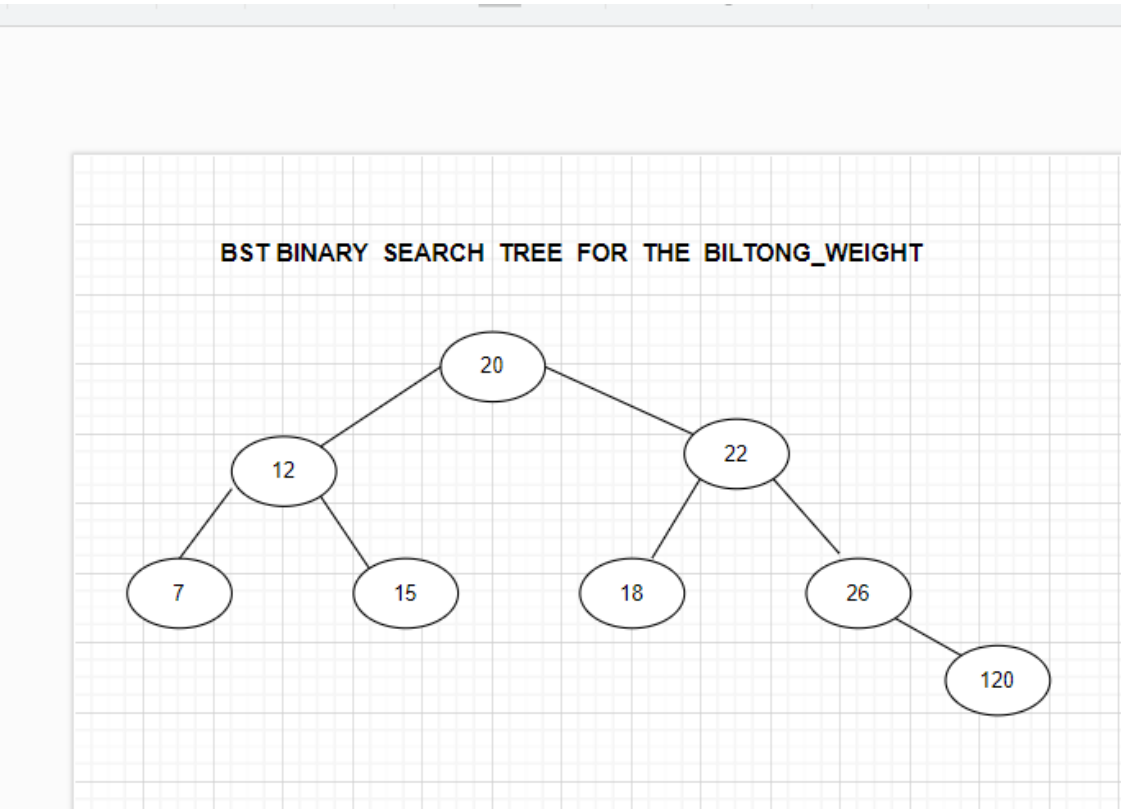
To represent the array it will be

Biltong_weight = [20, 12, 7, 22, 15, 26, 18, 120]

B)

(Anon., 2023)

THE BST (**BINARY SEARCH TREE**) IS AS FOLLOW :



3.2)

(Anon., 2023)

```
#QUESTION  3.2) A) ----------------------------------------
---
class Node:
    def __init__(self, data):
        self.left = self.right = None
        self.val = data


def insert(root, data):
    if not root:
        return Node(data)
    if data > root.val:
        root.right = insert(root.right, data)
    else:
        root.left = insert(root.left, data)
    return root
```

```
#QUESTION  3.2 B) ---------------------------------------------
--------
def inorder(root):
    return inorder(root.left) + [root.val] +
inorder(root.right) \
        if root else []

#QUESTION 3.2) C)---------------------------------------------
----------
def calculations(root):
    if not root:
        return None, None, None

    min_value, max_value = root.val, root.val

    while root.left:
        min_value, root = root.left.val, root.left

    while root.right:
        max_value, root = root.right.val, root.right

    division_result = max_value / min_value if min_value != 0
else None
    return min_value, max_value, division_result


# drawing  the BST
root = None
biltong_weights = [20, 12, 7, 22, 15, 26, 18, 120]

for weight in biltong_weights:
    root = insert(root, weight)
#  OUTPUT   FOR       THE   3.2  QUESTION --------------------
---
# sorting  the  date  Inorder traversal
sorted_weight = inorder(root)
print("Biltong Weights Inorder Traversal :", sorted_weight)

# displayying   of   the    min, max,  and  the  division
calculations
min_value, max_value, division_result = calculations(root)
print("Minimum Value =", min_value)
print("Maximum Value =", max_value)
print("Division Result (Max/Min)= ",
max_value,'/',min_value,'=  ', division_result)
```

PROOF OF OUTPUT :

```
Run    bst ×
"C:\Users\glodi kalombo\Documents\classof15august\pythonProject5\Scripts\python.exe" "C:\Use
Could not find platform independent libraries <prefix>
Biltong Weights Inorder Traversal : [7, 12, 15, 18, 20, 22, 26, 120]
Minimum Value = 7
Maximum Value = 20
Division Result (Max/Min)=  20 / 7 =   2.857142857142857

Process finished with exit code 0
```

Bibliography

Anon., 2023. *codeacademy.* [Online]
Available at: https://www.codecademy.com/resources/docs/general/binary-search-tree/inorder-traversal
[Accessed 12 11 2023].

Anon., 2023. *freecodecamp.* [Online]
Available at: https://www.freecodecamp.org/news/file-handling-in-python/
[Accessed 10 11 2023].

Anon., 2023. *freecodecamp.* [Online]
Available at: https://www.freecodecamp.org/news/introduction-to-linked-lists-in-python/#:~:text=More%20memory%20is%20required%20when,the%20option%20of%20random%20access.
[Accessed 11 11 2023].

Anon., 2023. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/python-program-for-merge-sort/
[Accessed 09 11 2023].

Anon., 2023. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/python-program-for-binary-search/
[Accessed 12 11 2023].

Anon., 2023. *pythontutorial.* [Online]
Available at: https://www.pythontutorial.net/python-basics/python-read-text-file/
[Accessed 10 11 2023].