Section A:

Question 1:

1.1

```
class Mytriangle():
    def _init_(self, base, height):
        self.base = base
        self.height = height
```

Class missed brackets (), def missed self and the underscore was removed

1.2

```
class Triangle():
    def _init_(self, base, altitude):
        self.base = base
        self.altitude = altitude
```

Class missed an :, def _init missed an : and underscores was removed

1.3

```
def main():
    creature = Vertebrate()
    print(creature.msg())
    print(isinstance(creature, Animal))

class Animal:
    def msg(self):
        return("Can Move.")
class Vertebrate:
    def msg(self):
        return("Has a backbone.")
main()
```

```
Has a backbone.
False
```

1.4

```
#what will this print
def sum(x, y=10):
    return x+y
print(sum(10))
```

```
20
```

1.5

```
import copy
a = ['Ndai', 'Tairo', ['Eunice', 'Doreen']]
b = copy.deepcopy(a)
a is b
```

```
False
```

Question 2:

```
Index = [] #Create an Index for the names

Number = int(input("How many names do you want to insert?")) #Get the total name
X = 0
while X != Number: #While the total names are not inserted, the program will not stop
    Name = input("Please enter names") #Ask the user for a name
    Index = [(X)] #Insert the name into
    X = X + 1
Search = input("Please Enter a name to search") #Ask the user for a name to search
for Y in Index:
        if Search == Index[Y]: #Goes through the index to search for the name
            print("The name exists")
            print("The index of the name is:", Y) #Prints the index of the name
        else:
            print("The Name is not found")


Change = input("Please enter the name to change") #Gets the name from the user that will be changed
for Z in Index:
    if Change == Index[(Z)]: #If the name is found the code will execute
        New_Name = input("Please enter a new name") #Ask the user for a new name
        Index[(Z)] = New_Name #Replaces the old name with the new one
        print("Name has been changed")
        for X in Index:
            print(Index[(X)]) #print all the names
    else:
        print("Name was not found")
        for X in Index:
            print(Index[(X)]) #print all the names
```

```
How many names do you want to insert?5
Please enter namesEddie
Please enter namesKyle
Please enter namesTiaan
Please enter namesBill
Please enter namesSteve
Please Enter a name to searchTiaan
```

Question 3:

```
ColourList = [] #Create index for the colors
def Display(): #Create the display function to display the names of the colors
    for X in ColourList:
        print(ColourList[X]) #prints all of the colors in the index
def Letter(): #Create the letter function to ask the user for a letter to be entered
    Letter = input("Please enter an uppercase letter")
def FillList(): #Create the filllist function to fill the index with color names
    for X in ColorFile: #The program goes through the whole colors.txt file
        if X == Letter: #If a color that begins with the letter that was enter by the user is found the code will execute
            print(ColorFile.readline()) #print the name of the color
            ColourList[F] = ColorFile.readline() #Add the name of the color to the index
def Main(): #create the main function for the whole program to be started
    ColorFile = open("Colors.txt","r") #open the color.txt file for reading
    Letter() #calls the letter function to be used
    FillList() #calls the filllist function to be used
    Display() #calls the display function to be used


Main() #Calls the main function to be used
```

The colors.txt file was missing, so the program could not be completed

Section B:

Question 4:

```
from math import gcd #import the gcd method to use it
class Fraction(): #create the class
    def __init__(self, numerator, denominator): #initialize the class
        numerator = int(input("Please enter a numerator")) #ask for a numerator
        denominator = int(input("Please enter a denominator")) #ask for a denominator
        self.numerator = numerator #make self.numerator = to numerator entered by the user
        self.denominator = denominator #make self.denominator = to denominator entered by the user
    def Greatest_DIV(self): #Create the Greatest Div function to get the greatest common divisor
        GCD = gcd(self.numerator, self.denominator) #Accquire the Greatest common divisor between the numerator and denominator
        self.numerator = self.numerator/GCD #Divide the numerator by the greatest common divisor
        self.denominator = self.denominator/GCD #Divide the denominator by the greatest common divisor
        print("The Lowest term of the fraction is:", self.numerator, "and", self.denominator)
    def Equivalent(self): #Create the Equivalent function to get the equal fraction
        EqualNumerator = self.numerator * 2 # Multiply the Numerator by 2 to get the equal numerator
        EqualDenominator = self.denominator * 2 # Multiply the denominator by 2 to get the equal denominator
        print("The equal fraction for each is: " , EqualNumerator , "and under it", EqualDenominator)
fraction = Fraction() #create the fraction class
fraction.Greatest_DIV() #Call the Greatest_Div function
fraction.Equivalent() #Call the Equivalent function
```

Question 5:

```
from tkinter import * #import tkinter to be used
EduvosCampus = Tk()
Campuslabel = Label(EduvosCampus, text = "Campus") #Create campus label
Personlabel = Label(EduvosCampus, text = "Person") #Create Person label
EduCampuslabel = Label(EduvosCampus, text = "Eduvos Campuses") #Create Eduvos campuses label
AcademicPerson = Listbox() #Create a listbox for all the academicpersons
EduCampus = Listbox() #Create a listbox for all the Eduvos campuses
AcademicPerson.insert(1, "Annemie Parkin") #Assign all the people to the listboxes
AcademicPerson.insert(2, "Anri Pienaar")
AcademicPerson.insert(3, "Claris Mhike")
AcademicPerson.insert(4, "Karin Soderlund")
AcademicPerson.insert(5, "Karin Soderlund")
AcademicPerson.insert(6, "Kyle Knickelbein")
AcademicPerson.insert(7, "Lance Krasner")
AcademicPerson.insert(8, "Olika Saikoolal")
AcademicPerson.insert(9, "Rafiq Mana")
AcademicPerson.insert(10, "Tendai Muzanenhamo")
AcademicPerson.insert(11, "Tess Biscombe")
EduCampus.insert(1, "Bedfordview") #Assign all the campuses to the listboxes
EduCampus.insert(2, "Bloemfontein")
EduCampus.insert(3, "Cape Town Claremont")
EduCampus.insert(4, "Cape Town Tygervalley")
EduCampus.insert(5, "Durban")
EduCampus.insert(6, "East London")
EduCampus.insert(7, "Mbombela")
EduCampus.insert(8, "Midrand")
EduCampus.insert(9, "Nelson Mandela")
EduCampus.insert(10, "Potchefstroom")
EduCampus.insert(11, "Pretoria")
MatchButton = Button(EduvosCampus, text = "Determine if Match is Correct") #Create the match button
AnswerLabel = Label(EduvosCampus, text = "Answer:") #Create a label for answer
AnswerEntry = Entry(EduvosCampus) #Create an entry for answer
Campuslabel.grid(row=1, column=2) #Plot all the GUI objects on the GUI
Personlabel.grid(row=2, column=1)
EduCampuslabel.grid(row=2, column=3)
AcademicPerson.grid(row=3, column=1)
EduCampus.grid(row=3, column=3)
MatchButton.grid(row=4, column=2)
AnswerLabel.grid(row=5, column=1)
AnswerEntry.grid(row=5, column=2)
EduvosCampus.mainloop() #Keep the GUI from stopping
```