####

#name: Thomas Williamson

#Student Id: 588206

#CMPT360 spring 2021

#assignment #1

#title: calculate triangle on complex plane

#####

Assignment 1

This assignment fulfills the following goals:

- A group I language(python)
- A group II language(Fortran)
- Implemented on PC

Assign#	Tate due	Group	Language	Language	Platform
1	Mon Week III	I & II	Python	Fortran	PC
2	Mon Week V				
3	Mon Week VII				
4	Mon Week IX				
5	Fri Week X				
6	Fri Week XII				
7	Mon Week XIV				

Restatement of problem

Wright a program in a language that I know and one I don't know that accepts the coordinates of three vertices of a triangle on a complex plane calculate the length of the edges and the angles in the triangle. Type checks the input. And finally, Make a summary of the differences of the two programing languages.

Sample I/O

Program asks for input for 3 vertices

Receives input for three vertices

Calculates the length of the sides of a triangle made of these three vertices

Calculates the angle at each vertex in the triangle

Prints the length of each side and the angle of each vertices

Operation

Runn program type vertex as in the form a+bi as in the complex plane when prompted for each vertex

Press enter

--- results

Error handling

if the coordinate is not put in in the correct formatting it will say so and you will have the opportunity to re enter the last coordinate that was entered on the chance that does not work restart the program and re enter the coordinates in the correct form

Python pseudo code

```
point1 = getinput(user input))

point2 = getinput(user input))

point3 = getinput(user input))


getinput(user input) – recursive

    try:

        if( last char of user input == i):

            point = split user input into just the numbers as an array minus the i

        else:

            rases error

        # convert user input to int

        point[0] = point[0] converted to int

        point[1] = point[1] converted to int

    except:

        states error and asks for new input

        puts new user input back through get input

    returns (point)


length12 = getLength(point1, point2)

length32 = getLength(point3, point2)
```

```
length31 = getLength(point3, point1)


getLength(point1, point2):  #calculates length form each point to point
    if statements: calculate which of the two points is higher on x and on y
    templeng1 = subtract the lower one from the higher on x axis
    templeng2= subtract the lower one from the higher on y axis
    returns ($\sqrt{\mathrm{templeng1}^{\;2} + \mathrm{templeng2}^{2}}$)



angle1 = #the angle of point 1in degrees
arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees
angle2 =  #the angle of point 2in degrees
arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees


angle3 =  #the angle of point 3in degrees
arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees


print results:
length12
length31
length32
angle1
angle2
angle3


all variables
```

point, point1, point2, point3, templeng1, templeng2, length12, length31, length32, angle1, angle2, angle3

all imports

math

all defined functions

getinput(point) returns point as num array

getlength(point1,point2) returns length

fortran pseudo code

```
prompt user
point = user input
pint1V= get input(point1)
prompt user
point2 = user inpurt
pint2V = getinput(point2)
prompt user
point3 = user input
pint3V = getinput(point3)
```

```
getinput(point) – recursive
    if( last char of user input == i) and valueCheck(point):
        point = split user input into just the numbers as an array minus the i
        # convert user input to int
        point[0] = point[0] converted to int
        point[1] = point[1] converted to int
    else:
        states error and asks for new input
        puts new user input back through get input
    returns (point)
```

```
valueCheck(point)
    makes sure point does not contain unwanted characters
    do for each charachte
        if (character is not between “0” and “9” or is not “+” or “ “):
            return false
        else:
            return true
getLength(point1, point2):  #calculates length form each point to point
    if statements: calculate which of the two points is higher on x and on y
    templeng1 = subtract the lower one from the higher on x axis
    templeng2= subtract the lower one from the higher on y axis
    returns ($\sqrt{\mathrm{templeng1}\ ^2 + \mathrm{templeng2}^2}$)
```

angle1 = #the angle of point 1in degrees

arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees

angle2 = #the angle of point 2in degrees

arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees

angle3 = #the angle of point 3in degrees

arccos of (length12**2) + (length31**2) - (length32**2)) / (2*length12*length31) converted into degrees

list of variables:

point1, point2, point3, pint1v, pint2v, pint3v, length12, lenght32, length31, angle1, angle2, angle3, torf, point, c, pintv, pintv4, templeng1, templeng2, templeng3,

no imports

list of defined functions

valueCheck(point)

```
getinput(point)

getLength(pintv, pintv4)
```

notable differences between python and Fortran

python does not require variable types to be stated however Fortran does. Fortran does not have a try catch statement for errors, however python does make type detection easier in python than Fortran. Finally, Fortran’s variables where not case sensitive but pythons is this however did not play a large roll in my experience. Overall Fortran is a much complicated and fickle language to learn than python but was not actually as difficult as I was led to believe.

Trials

Fortran: inputs 0+0i, 10+0, 10+0i, 0+10i

```
!

!name: Thomas Williamson

!Student Id: 588206

!CMPT360 spring 2021

!assignment #1

!title: calculate triangle on complex plane

!

input point 1 as "a + bi"

0+0i

 input point 2 as "a + bi"

10+0

 Error incorrect type\ntry again?\ninput point as "a + bi"

10+0i

 input point 3 as "a + bi"

0+10i

 point1 angle:    90.0000000

 point2 angle:    45.0000000

 point3 angle:    45.0000000
 length point1 - point2:    10.0000000
 length point2 - point3:    14.1421356
 length point1 - point3:    10.0000000
```

inputs 10000+55423i, 153+89i, 87913146+1567961318421i

```
input point 1 as "a + bi"
10000+55423i
 input point 2 as "a + bi"
153+89i
 input point 3 as "a + bi"
87913146+1567961318421i
 point1 angle:    90.0000000
 point2 angle:    90.0000000
 point3 angle:    0.00000000
 length point1 - point2:    56203.3359
 length point2 - point3:    1.56796138E+12
 length point1 - point3:    1.56796138E+12
```

input 0i

```
input point 1 as "a + bi"
0i
At line 86 of file .\hello.f90
Fortran runtime error: End of file

Error termination. Backtrace:

Could not print backtrace: libbacktrace could not find executable to open
```

(did not have a catch for out of range)

python: inputs 0+0i, 10+0, 10+0i, 0+10i

```
####
#name: Thomas Williamson
```

```
        #Student Id: 588206
        #CMPT360 spring 2021
        #assignment #1
        #title: calculate triangle on complex plane
        #####
        input point 1 as "a + bi"0+0i
        input point 2 as "a + bi"10+0
        Error incorrect type
        try again?
        input point as "a + bi"10+0i
        input point 3 as "a + bi"0+10i
        point1 angle: 90.00
        point2 angle: 45.00
        point3 angle: 45.00
        length point1 - point2: 10.00
length point2 - point3: 14.14
length point1 - point3: 10.00
inputs 10000+55423i, 153+89i, 87913146+1567961318421i
        input point 1 as "a + bi"10000+55423i
        input point 2 as "a + bi"153+89i
        input point 3 as "a + bi"87913146+1567961318421i
        point1 angle: 169.91
        point2 angle: 10.09
        point3 angle: 0.00
        length point1 - point2: 56203.34
        length point2 - point3: 1567961320796.57
        length point1 - point3: 1567961265462.02
inputs 0i, 102.12+0.89i, 10+10i, 10+10i, 10+10i
        input point 1 as "a + bi"0i
        Error incorrect  type

        try again?

        input point as "a + bi"102.12+0.89i

        Error incorrect  type

        try again?

        input point as "a + bi"10+10i

        input point 2 as "a + bi"10+10i

        input point 3 as "a + bi"10+10i

        Traceback (most recent call last):
```

(does not except float)

```
  File "C:\Users\thelo\Google Drive\semester 8\CMPT 360\1\assignment 1python.py", line 52, in <module>

    angle1 = math.degrees(math.acos(((length12**2)+(length31**2)-(length32**2))/(2*length12*length31)))

ZeroDivisionError: float division by zero

py

input point 1 as "a + bi"0i

Traceback (most recent call last):

  File "C:\Users\thelo\Google Drive\semester 8\CMPT 360\1\assignment 1python.py", line 43, in <module>

    point1 = getinput(input('input point 1 as "a + bi"'))

  File "C:\Users\thelo\Google Drive\semester 8\CMPT 360\1\assignment 1python.py", line 21, in getinput

    point[1] = int(point[1])

IndexError: list index out of range
```

(I

FORTRAN code:

```
!
!name: Thomas Williamson
!Student Id: 588206
!CMPT360 spring 2021
!assignment #1
!title: calculate triangle on complex plane
!

program triangleCalc

implicit none

!gets inputs
character*100 :: point1
real :: pint1V(2)
real :: pint2V(2)
real :: pint3V(2)
character*100 :: point2
character*100 :: point3
REAL, PARAMETER :: Pi = 3.1415927

!gets lengths
real :: length12
real :: length32
real :: length31
!gets angle of each point
real :: angle1
real :: angle2
real :: angle3
```

```
print *, 'input point 1 as "a + bi"'
read *, point1
pint1V = getinput(point1)
print *, 'input point 2 as "a + bi"'
read *, point2
pint2V = getinput(point2)
print *,'input point 3 as "a + bi"'
read *, point3
pint3V = getinput(point3)

length12 = getLength(pint1V, pint2V)
length32 = getLength(pint3V, pint2V)
length31 = getLength(pint3V, pint1V)
!gets angle of each point
angle1 = (acos(((length12**2)+(length31**2)-(length32**2))/(2*length12*length31)))*(180/pi)
angle2 = (acos(((length12**2)+(length32**2)-(length31**2))/(2*length12*length32)))*(180/pi)
angle3 = (acos(((length32**2)+(length31**2)-(length12**2))/(2*length32*length31)))*(180/pi)

!prints results
print *, "point1 angle: ", angle1
print *, "point2 angle: ", angle2
print *, "point3 angle: ", angle3
print *, "length point1 - point2: ", length12
print *, "length point2 - point3: ", length32
print *, "length point1 - point3: ", length31


contains
```

```
    !check valididty
    function valueCheck(point) result(torf)
       implicit none
       logical :: torf
       character*100 :: point
       integer :: c
       torf = .true.
       !print *, point
       do c=1, len(trim(point))-1, 1
         ! print*, point(c:c)
          if (.not. ((point(c:c) >= "0") .and. (point(c:c) <= '9') .or. (point(c:c) == "+") .or. (point(c:c) == " ")) ) then
             torf = .false.
             exit
          end if
       end do
    end function valueCheck

   ! check valididty of input and change input to int
    recursive function getinput(point) result(pointV)
       implicit none
       character*100 :: point
       real :: pointV(2)
       !print*, point
       !print*, point(len(trim(point)):len(trim(point)))
       if ((point(len(trim(point)):len(trim(point))) == 'i') .and. valueCheck(point)) then
          read(point(1:(index(point, "+")-1)) ,*) pointV(1)
          read(point((index(trim(point), "+")+1):len(trim(point))-1) ,*) pointV(2)
       else
```

```
         print *, 'Error incorect type\ntry again?\ninput point as "a + bi"'
         read *, point
         pointV = getinput(point)
      end if
   end function getinput

   !calculates length form each point to point
   function getLength(pintV, pintV4) result(length)
      implicit none
      real :: pintV(2)
      real :: pintV4(2)
      real :: templeng1, templeng2, length
      if (pintV(1) <= pintV4(1)) then
         templeng1 = (pintV4(1)-pintV(1))
         if(pintV(2) <= pintV4(2)) then
            templeng2 = pintV4(2)-pintV(2)
         else
            templeng2 = pintV(2)-pintV4(2)
         end if
      else
         templeng1 = pintV(1)-pintV4(1)
         if (pintV(2) <= pintV4(2)) then
            templeng2 = pintV4(2)-pintV(2)
         else
            templeng2 = pintV(2)-pintV4(2)
         end if
      end if
      length = (sqrt((templeng1**2)+(templeng2**2)))
   end function getLength
end program triangleCalc
```

PYTHON code

```
####
#name: Thomas Williamson
#Student Id: 588206
#CMPT360 spring 2021
#assignment #1
#title: calcualte triangle on complex plane
#####
import math

#gets point
#check valididty of input and change input to int
def getinput(point):
   try:
      if point[-1] == 'i':
         point = point[:-1].split("+")
      else:
         raise ValueError()
      #convert user input to int
      point[0] = int(point[0])
      point[1] = int(point[1])
   except:
      point = getinput(input('Error incorrect  type\ntry again?\ninput point as "a + bi"'))
   return(point)

#calculates length form each point to point
def getLength(point1, point2):
   if point1[0] <= point2[0]:
      templeng1 = point2[0]-point1[0]
      if point1[1] <= point2[1]:
            templeng2 = point2[1]-point1[1]
        else:
            templeng2 = point1[1]-point2[1]
    else:
        templeng1 = point1[0]-point2[0]
        if point1[1] <= point2[1]:
            templeng2 = point2[1]-point1[1]
        else:
            templeng2 = point1[1]-point2[1]
    return(math.sqrt((templeng1**2)+(templeng2**2)))


#gets inputs
point1 = getinput(input('input point 1 as "a + bi"'))
point2 = getinput(input('input point 2 as "a + bi"'))
point3 = getinput(input('input point 3 as "a + bi"'))


#gets lengths
length12 = getLength(point1, point2)
length32 = getLength(point3, point2)
length31 = getLength(point3, point1)
#gets angle of each point
angle1 = math.degrees(math.acos(((length12**2)+(length31**2)-(length32**2))/(2*length12*length31)))
angle2 = math.degrees(math.acos(((length12**2)+(length32**2)-(length31**2))/(2*length12*length32)))
angle3 = math.degrees(math.acos(((length32**2)+(length31**2)-(length12**2))/(2*length32*length31)))


#prints results
print("point1 angle: %.2f" % angle1)
print("point2 angle: %.2f" % angle2)
print("point3 angle: %.2f" % angle3)
print("length point1 - point2: %.2f" % length12)
print("length point2 - point3: %.2f" % length32)
print("length point1 - point3: %.2f" % length31)
```