```python
# I declare that my work contains no examples of misconduct, such as plagiarism, or collusion.
# Any code taken from other sources is referenced within my code solution.
# All the reference to my course work are include in my test case file.
# Student ID: w2052917
# Date: 2023.12.13

from graphics import *

#define a function to calculate the outcome
def calculate_progression(pass_credits, defer_credits, fail_credits):
    if pass_credits == 120:
        return "Progress"
    elif pass_credits == 100:
        return "Progress (module trailer)"
    elif fail_credits >= 80 and fail_credits % 20 == 0:
        return "Exclude"
    else:
        return "Module retriever"
#define a fuction to get valid credits
def validate_credits(credits):
    if not credits.isdigit():
        print("\nInteger required")
        return False
    credits = int(credits)
    if credits not in [0, 20, 40, 60, 80, 100, 120]:
        print("\nOut of range")
        return False
    return credits

def main():
    user=0
    progress_count = 0
    trailer_count = 0
    retriever_count = 0
    exclude_count = 0
    total_outcomes = 0
    successful_inputs = []

    #Ask user whether he is a student or a staff member
    print("\nEnter '1' for Student version or '2' for Staff version.")
    try:
        user = int(input("What version do you wish to use ?:"))
    except ValueError:
        print("Please enter a valid input.")

    while user not in [1,2]:
        print("\nInvalid user. Please enter 1 or 2.")
        try:
           user = int(input("What version do you wish to use ?:"))
        except ValueError:
           print("Please enter a valid input.")
        continue

    # if user is a student, calculate the outcome
    while user == 1 :
        while True:
            pass_credits = input("\nEnter the total PASS credits: ")
            pass_credits = validate_credits(pass_credits)
            if pass_credits is False:
                continue
            break
        while True:
            defer_credits = input("Enter the total DEFER credits: ")
            defer_credits = validate_credits(defer_credits)
            if defer_credits is False:
                continue
            break

        while True:
            fail_credits = input("Enter the total FAIL credits: ")
            fail_credits = validate_credits(fail_credits)
            if fail_credits is False:
                continue
            break

        total_credits = pass_credits + defer_credits + fail_credits
        if total_credits != 120:
                print("\nTotal incorrect")
                continue

        outcome = calculate_progression(pass_credits, defer_credits, fail_credits)
        print(f"\n{outcome}")
        print("\nEnd of program")
        break

    while True:
        #if user is a staff member, calculate the outcome of each individu student until he quits the program
        if user == 2 :
            while True:
                pass_credits = input("\nEnter the total PASS credits: ")
                pass_credits = validate_credits(pass_credits)
                if pass_credits is False:
                    continue
                break
            while True:
                defer_credits = input("Enter the total DEFER credits: ")
                defer_credits = validate_credits(defer_credits)
                if defer_credits is False:
                    continue
                break

            while True:
                fail_credits = input("Enter the total FAIL credits: ")
                fail_credits = validate_credits(fail_credits)
                if fail_credits is False:
                    continue
                break

            total_credits = pass_credits + defer_credits + fail_credits
            if total_credits != 120:
                print("\nTotal incorrect")
                continue

            outcome = calculate_progression(pass_credits, defer_credits, fail_credits)

            # Update count variables
            if outcome == "Progress":
                progress_count += 1
            elif outcome == "Progress (module trailer)":
                trailer_count += 1
            elif outcome == "Exclude":
                exclude_count += 1
            elif outcome == "Module retriever":
                retriever_count += 1

            total_outcomes = progress_count+trailer_count+exclude_count+retriever_count
            #append information to null list
            successful_inputs.append({"PASS credits": pass_credits, "DEFER credits": defer_credits,"FAIL credits": fail_credits,"Outcome":
outcome})

            print(f"\n{outcome}")

            #asking staff member to continue or quit the program
            string1 = str(input("\nWould you like to enter another set of data? \nEnter 'y' for yes or 'q' to quit and view result: ")).
lower().replace(" ", "")

            if string1.lower() == 'q':
                break
            if string1.lower() == 'y':
                 continue

            while string1 not in ['y','q']:
                print("\nInvalid letter. Please enter 'y' or 'q'.")
                string1 = str(input("\nWould you like to enter another set of data?\nEnter 'y' for yes or 'q' to quit and view results:")).
lower().replace(" ", "")
                continue

    #insert window and make background white size and window heading
    win = GraphWin("Histogram", 600, 550)
    win.setBackground("White")

    #add topic to bar chart
    text0 = Text(Point(150,50),"Histogram Results")
    text0.setFace("arial")
    text0.setSize(22)
    text0.setStyle("bold")
    text0.setTextColor("black")
    text0.draw(win)

    #add perment line of chart
    aLine = Line(Point(50,450), Point(550,450))
    aLine.draw(win)

    #frist bar to represent outcome of progress
    aRectangle1 = Rectangle(Point(70,450), Point(170,450-(progress_count*10)))
    aRectangle1.setFill("pale green")
    aRectangle1.draw(win)

    #add title to frist bar as progress blowe it
    text1 = Text(Point(120,475),"Progress")
    text1.setSize(18)
    text1.setStyle("bold")
    text1.setTextColor("slate gray")
    text1.draw(win)

    #add the count of progress top of the 1st bar
    text6 = Text(Point(120,((450-(progress_count*10))-20)),f"{progress_count}")
    text6.setSize(18)
    text6.setStyle("bold")
    text6.setTextColor("slate gray")
    text6.draw(win)

    # second bar to represent outcome of trailer
    aRectangle2 = Rectangle(Point(190,450), Point(290,450-(trailer_count*10)))
    aRectangle2.setFill("dark sea green")
    aRectangle2.draw(win)

    #add title to 2nd bar as Trailer blowe it
    text2 = Text(Point(240,475),"Trailer")
    text2.setSize(18)
    text2.setStyle("bold")
    text2.setTextColor("slate gray")
    text2.draw(win)

    #add the count of trailer top of the 2nd  bar
    text7 = Text(Point(240,((450-(trailer_count*10))-20)),f"{trailer_count}")
    text7.setSize(18)
    text7.setStyle("bold")
    text7.setTextColor("slate gray")
    text7.draw(win)

    #third bar to represent outcome of retriever
    aRectangle3 = Rectangle(Point(310,450), Point(410,450-(retriever_count*10)))
    aRectangle3.setFill("DarkOliveGreen3")
    aRectangle3.draw(win)

    #add title to 3rd bar as retriever blowe it
    text3 = Text(Point(360,475),"Retriever")
    text3.setSize(18)
    text3.setStyle("bold")
    text3.setTextColor("slate gray")
    text3.draw(win)

    #add the count of retriever top of the 3rd bar
    text8 = Text(Point(360,((450-(retriever_count*10))-20)),f"{retriever_count}")
    text8.setSize(18)
    text8.setStyle("bold")
    text8.setTextColor("slate gray")
    text8.draw(win)

    #4th bar to represent outcome of exclude
    aRectangle4 = Rectangle(Point(430,450), Point(530,450-(exclude_count*10)))
    aRectangle4.setFill(color_rgb(210,182,181))
    aRectangle4.draw(win)

    #add title to 4th bar as exclude blowe it
    text4 = Text(Point(480,475),"Excluded")
    text4.setSize(18)
    text4.setStyle("bold")
    text4.setTextColor("slate gray")
    text4.draw(win)

    #add the count of exclude top of the 4tg bar
    text9 = Text(Point(480,((450-(exclude_count*10))-20)),f"{exclude_count}")
    text9.setSize(18)
    text9.setStyle("bold")
    text9.setTextColor("slate gray")
    text9.draw(win)

    #add total outcomes in the bottem
    text5 = Text(Point(175,525),f"{total_outcomes} Outcomes in total.")
    text5.setSize(22)
    text5.setStyle("bold")
    text5.setTextColor("slate gray")
    text5.draw(win)

    print("\n")
    # display the outcomes and marks of each individual student
    print("Part 2 :")
    for info in successful_inputs:
          print(info["Outcome"],"-", info["PASS credits"],",",info["DEFER credits"],",",info["FAIL credits"])

    print("\nEnd of program")
    #store list extension data to text file
    with open("w2052917_part_3.txt","+w") as file:
        file.write("Part 3 :\n")
        for info in successful_inputs:
            file.write(str(info["Outcome"]) + " - " +str(info["PASS credits"]) + " , " +str(info["DEFER credits"]) + " , " +str(info["FAIL
credits"])+"\n")
```

```
main()
```