Create a flow chart of the login activity with set of questions from SAS 11 to 13
limit the number of login to 5 tries
10 questions Max per SAS
and if the score is 10 to 15 pass
16 to 20 great score
21 to 29 its a up to you
30 perfect

ANSWER:

```
login_attempts = 5

user = "user"
password = "pass"

questions = [
    ("1. A loop enables a particular set of conditions to be executed
repeatedly.", "True", "SAS 11"),
    ("2. Else statement is a way of iterating through each item is by
index offset into the sequence itself.", "False", "SAS 11"),
    ("3. A loop that is never finished is known as an infinite loop.",
"True", "SAS 11"),
    ("4. If a sequence contains an expression list, it is evaluated
first.", "True", "SAS 11"),
    ("5. Loop has the ability to iterate over the items of any sequence,
such as a list or a string.", "True", "SAS 11"),
    ("6. A function that is called loop take a single number and behave
like a list.", "False", "SAS 11"),
    ("7. The following is an example of range: [0, 1, 2, 3, 4, 5, 6, 7].",
"True", "SAS 11"),
    ("8. Nested loop a python programming language allows using one loop
inside another loop.", "True", "SAS 11"),
    ("9. An infinite loop is very useful in embedded systems.", "True",
"SAS 11"),
    ("10. If the else statement is used with a while loop, the else
statement is executed when the condition becomes false.", "True", "SAS
11"),

    ("1. The break statement is used inside the loop to exit out of the
loop.", "True", "SAS 12"),
    ("2. The loop ignores the rest of the loop body statements for the
current iteration and returns program execution to the very first
statement in the loop body.", "False", "SAS 12"),
    ("3. Continue statement allows you to bypass the current iteration of
any loop.", "True", "SAS 12"),
```

```
    ("4. Python supports the following control statements: Loop Control
Statements, Break, Continue, Pass.", "True", "SAS 12"),
    ("5. In Python, the loop statement is considered a no-operation
statement.", "False", "SAS 12"),
    ("6. The pass statement is much like a post.", "False", "SAS 12"),
    ("7. The pass statement is considered a no-operation statement, which
means it consumes the execution cycle like a valid Python statement, but
nothing happens actually when pass is executed.", "True", "SAS 12"),
    ("8. Whenever the controller encountered a break statement, it comes
out of that loop immediately.", "True", "SAS 12"),
    ("9. Break statement increases execution time.", "False", "SAS 12"),
    ("10. When a continuous statement is encountered in the loop, the
Python interpreter ignores the rest of the loop body statements.", "True",
"SAS 12"),

    ("1. Non -logical operators are used on conditional statements.",
"False", "SAS 13"),
    ("2. Python has three logical operators.", "True", "SAS 13"),
    ("3. If both conditions are false, it returns True.", "False", "SAS
13"),
    ("4. AND, OR, and NOT are the three logical operators of Python.",
"True", "SAS 13"),
    ("5. Pythons evaluates logical operators in the order they are listed
when you mix them in an expression, known as operator precedence.",
"True", "SAS 13"),
    ("6. Logical OR: true if the operands is false.", "False", "SAS 13"),
    ("7. Logical Not: true if the operands is false.", "True", "SAS 13"),
    ("8. Logical AND: true if the operands is true.", "False", "SAS 13"),
    ("9. Print it outputs the specified to the screen or another standard
output device.", "True", "SAS 13"),
    ("10. A function is a reusable piece of code that executes a single,
related action is known as Python Built-in Functions.", "True", "SAS 13"),

]

sas11_score = 0
sas12_score = 0
sas13_score = 0

while login_attempts > 0:
    input_user = input("Enter username: ")
    input_pass = input("Enter password: ")
    if input_user != user or input_pass != password:
        print("Wrong credentials. Please try again.")
        login_attempts -= 1
    else:
        print("You have logged in\n")
```

```
        print("True or False")
        for question, answer, sas in questions:
            input_answer = input(f"{question} (SAS {sas})\n")
            if input_answer == answer:
                if sas == "SAS 11":
                    sas11_score += 1
                elif sas == "SAS 12":
                    sas12_score += 1
                else:
                    sas13_score += 1
                print("Correct")
            else:
                print("Incorrect")

        sas_sum = sas11_score + sas12_score + sas13_score

        print("SAS 11 Score:", sas11_score)
        print("SAS 12 Score:", sas12_score)
        print("SAS 13 Score:", sas13_score)
        print("Total Score:", sas_sum, "Average:", str(sas_sum // 3))
        if sas_sum == 30:
            print("Perfect")
        elif 26 <= sas_sum <= 29:
            print("Excellent")
        elif 20 <= sas_sum <= 25:
            print("Great score")
        elif 15 <= sas_sum <= 20:
            print("Pass")
        else:
            print("Fail")
        break
else:
    print("Login fail")
```

FLOWCHART URL

https://docs.google.com/document/d/1rINO-ApgXd-3QBcNW39oIr8W-EnZiFAJNBPfU3SvEbo/edit

Group 1 Flowchart
Demircan, Pacifico, Cabellon