COMPUTER SCIENCE

ALGORITHM IN PSEUDOCODE

Validation & Verification

STARTER

-

KNOWING WHAT YOU KNOW

Go to:

https://joinmyquiz.com

- You are to write your real name and grade.
- Example: Thanh 10G5

LESSON OBJECTIVES

Students should be able to:

- Understand what validation and verification are
- Understand pseudocode structure of validation
- Understand pseudocode structure of verification

Validation and Verification

_____________it ensures that the data is reasonable and acceptable.

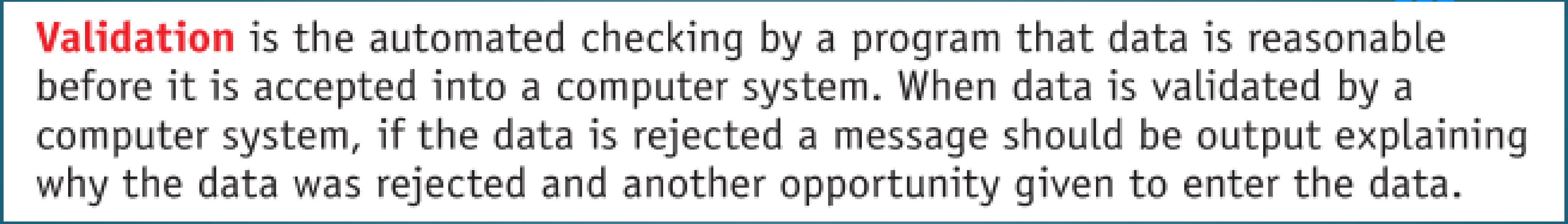
_____________it used to check that the data does not change as it is being entered.

Validation and Verification

Validation ________ it ensures that the data is reasonable and acceptable.

Verification ________ it is used to check that the data does not change as it is being entered.

VALIDATION

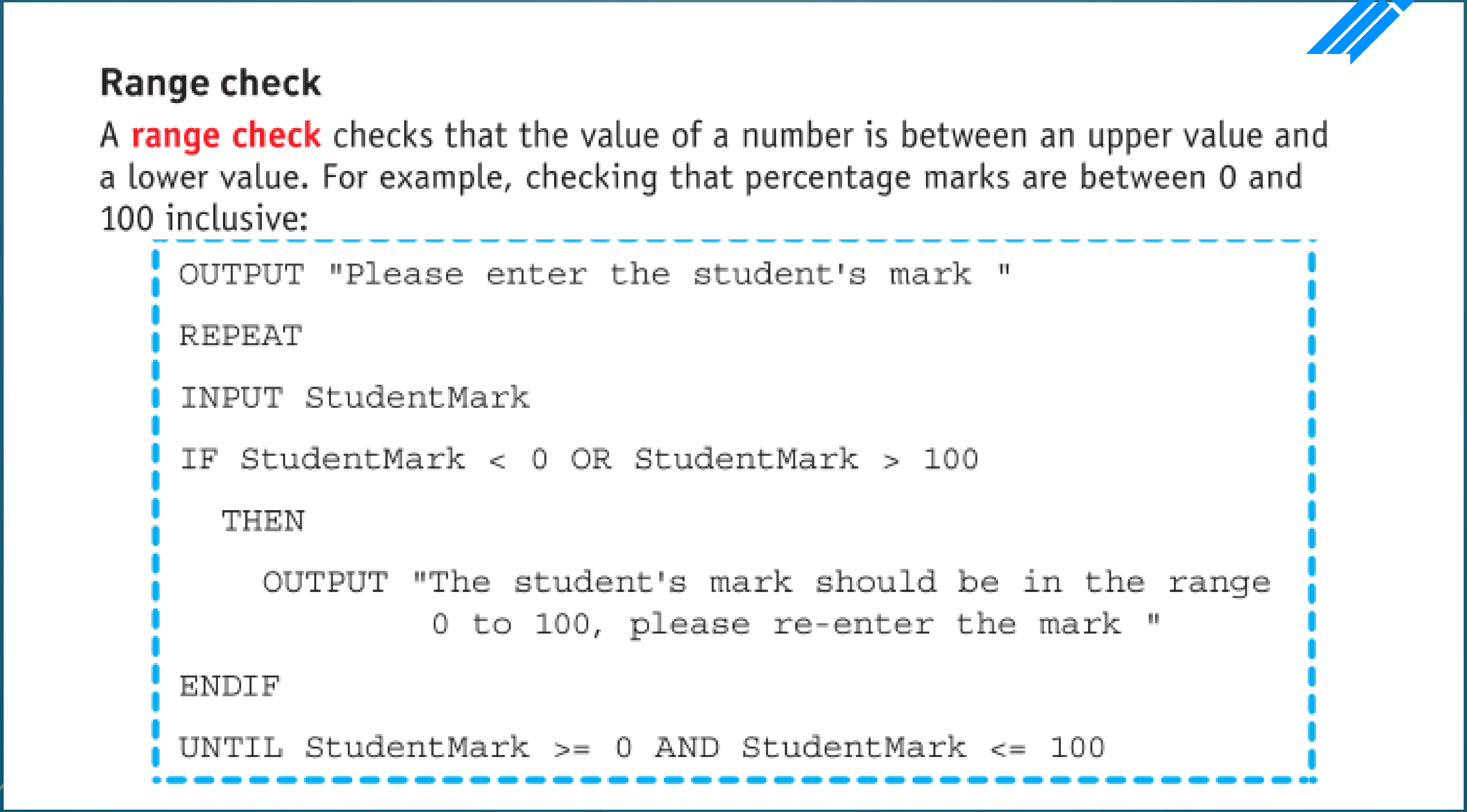

Validation is the automated checking by a program that data is reasonable before it is accepted into a computer system. When data is validated by a computer system, if the data is rejected a message should be output explaining why the data was rejected and another opportunity given to enter the data.

Types of Validation Check

- Range checks
- Length checks
- Type checks
- Presence checks
- Format checks
- Check digits

Note: A range of checks can be combined in a certain programme for a desired validation.

VALIDATION

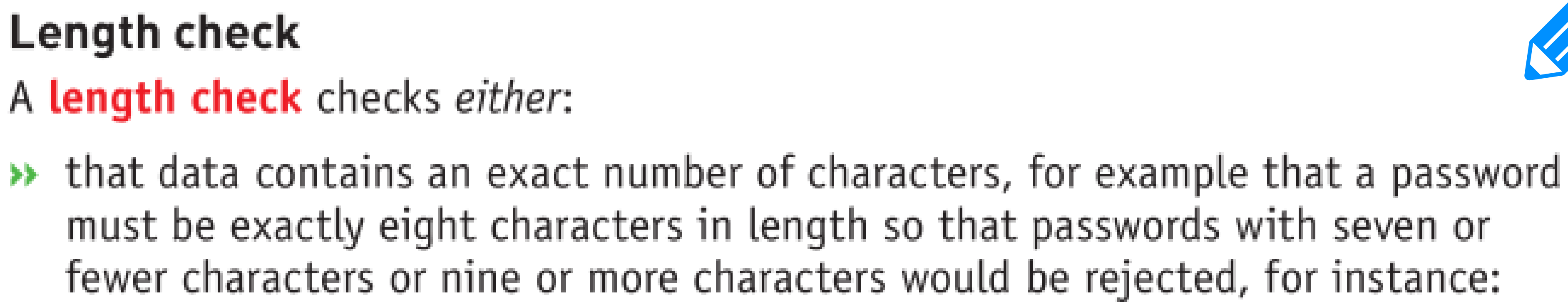


Range check

A **range check** checks that the value of a number is between an upper value and a lower value. For example, checking that percentage marks are between 0 and 100 inclusive:

```
OUTPUT "Please enter the student's mark "
REPEAT
INPUT StudentMark
IF StudentMark < 0 OR StudentMark > 100
  THEN
    OUTPUT "The student's mark should be in the range
            0 to 100, please re-enter the mark "
ENDIF
UNTIL StudentMark >= 0 AND StudentMark <= 100
```

VALIDATION



Length check

A **length check** checks *either*:

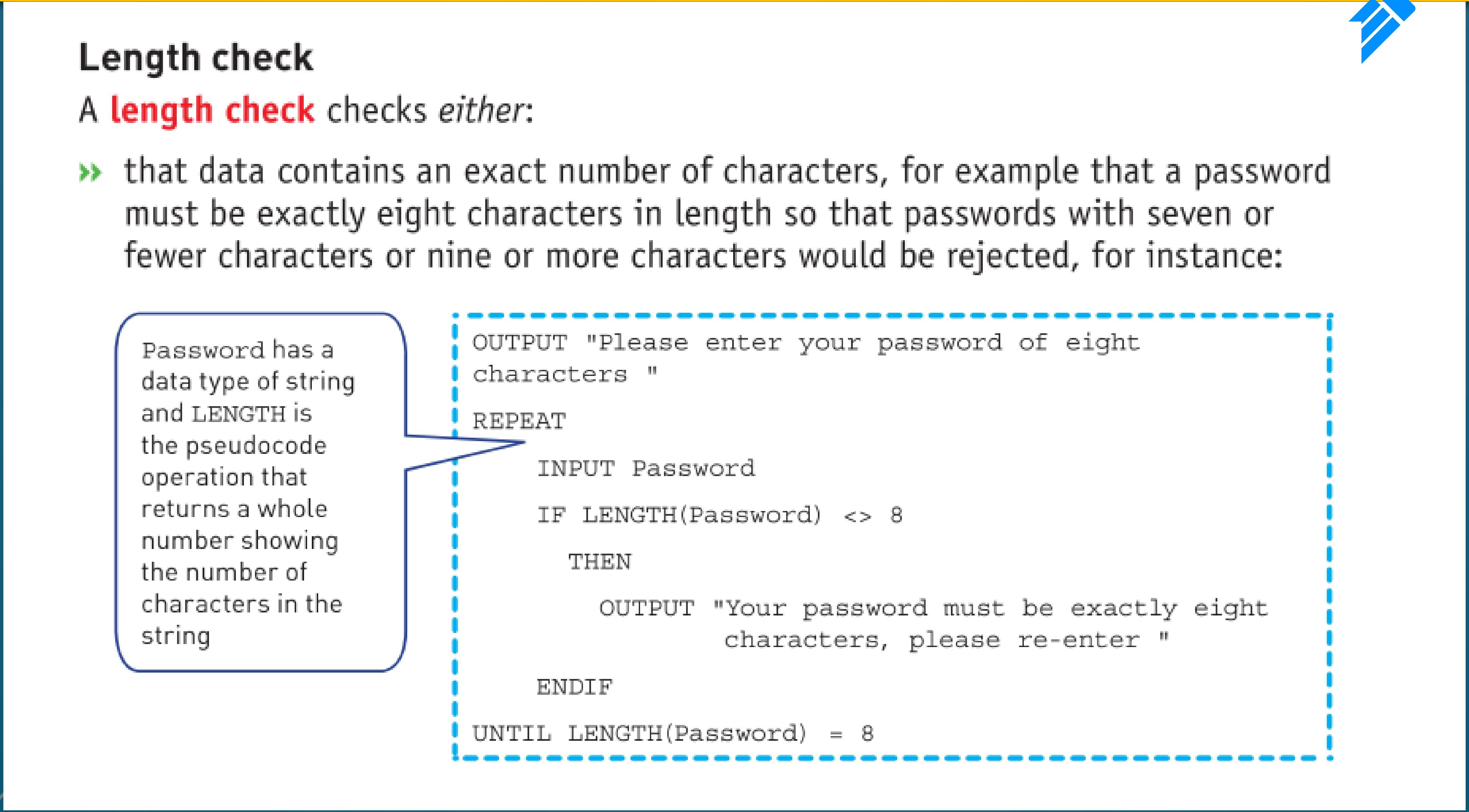
- that data contains an exact number of characters, for example that a password must be exactly eight characters in length so that passwords with seven or fewer characters or nine or more characters would be rejected, for instance:

VALIDATION

Length check

```
OUTPUT "Please enter your password of eight
characters "
REPEAT
    INPUT Password
    IF LENGTH(Password) <> 8
      THEN
        OUTPUT "Your password must be exactly eight
                characters, please re-enter "
    ENDIF
UNTIL LENGTH(Password) = 8
```

VALIDATION



Length check

A **length check** checks *either*:

- that data contains an exact number of characters, for example that a password must be exactly eight characters in length so that passwords with seven or fewer characters or nine or more characters would be rejected, for instance:

Password has a data type of string and LENGTH is the pseudocode operation that returns a whole number showing the number of characters in the string

```
OUTPUT "Please enter your password of eight
characters "
REPEAT
    INPUT Password
    IF LENGTH(Password) <> 8
      THEN
        OUTPUT "Your password must be exactly eight
                characters, please re-enter "
    ENDIF
UNTIL LENGTH(Password) = 8
```

VALIDATION

Length check

A **length check** checks *either*:

» that data contains an exact number of characters, for example that a password must be exactly eight characters in length so that passwords with seven or fewer characters or nine or more characters would be rejected, for instance:

» *or* that the data entered is a reasonable number of characters, for example, a family name could be between two and thirty characters inclusive so that names with one character or thirty-one or more characters would be rejected.

VALIDATION

-

FamilyName has a data type of string and LENGTH is the pseudocode operation that returns a whole number showing the number of characters in the string

```
OUTPUT "Please enter your family name "
REPEAT
  INPUT FamilyName
  IF LENGTH(FamilyName) > 30 OR LENGTH(FamilyName) < 2
    THEN
      OUTPUT "Too short or too long,
      please re-enter "
  ENDIF
UNTIL LENGTH(FamilyName) <= 30 AND LENGTH(FamilyName) >= 2
```

VALIDATION

-

Prese

A **pres** and the
value nline
transa

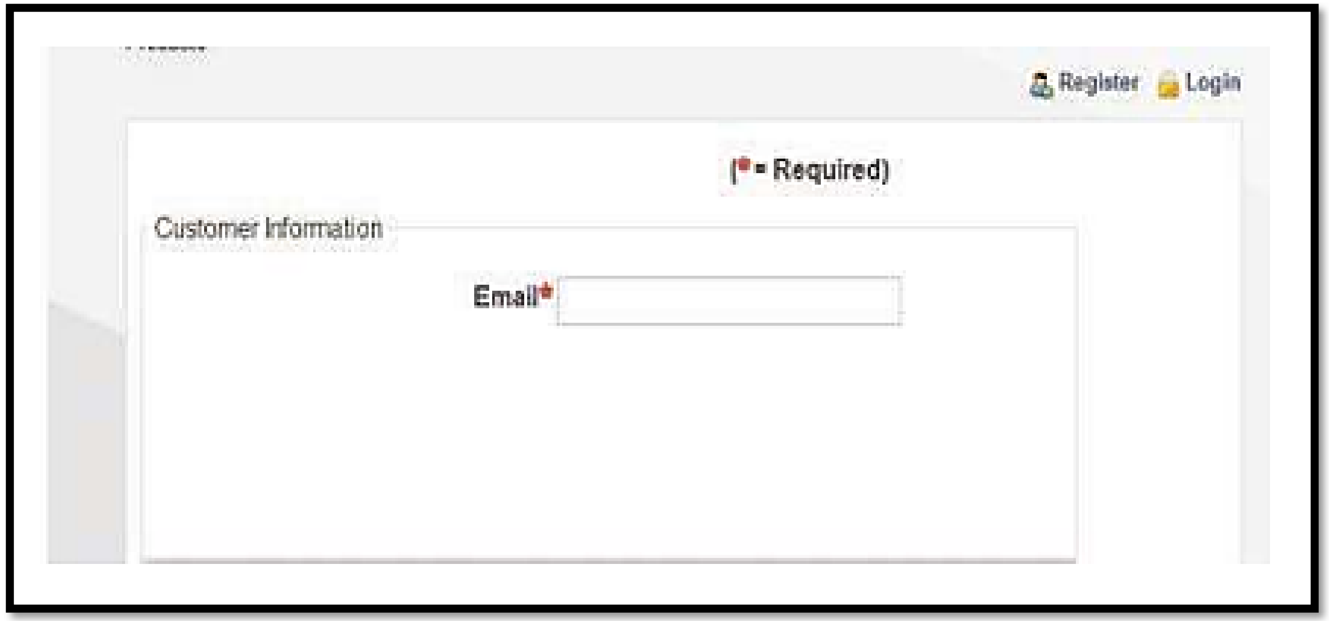
```
OUTPUT "Please enter your email address "
REPEAT
INPUT EmailAddress
IF EmailAddress = ""
   THEN
     OUTPUT "*=Required "
ENDIF
UNTIL EmailAddress <> ""
```

VALIDATION

Presence check

A **presence check** checks to ensure that some data has been entered and the value has not been left blank, for example, an email address for an online transaction must be completed.

```
OUTPUT "Please enter your email address "
REPEAT
INPUT EmailAddress
IF EmailAddress = ""
  THEN
   OUTPUT "*=Required "
ENDIF
UNTIL EmailAddress <> ""
```

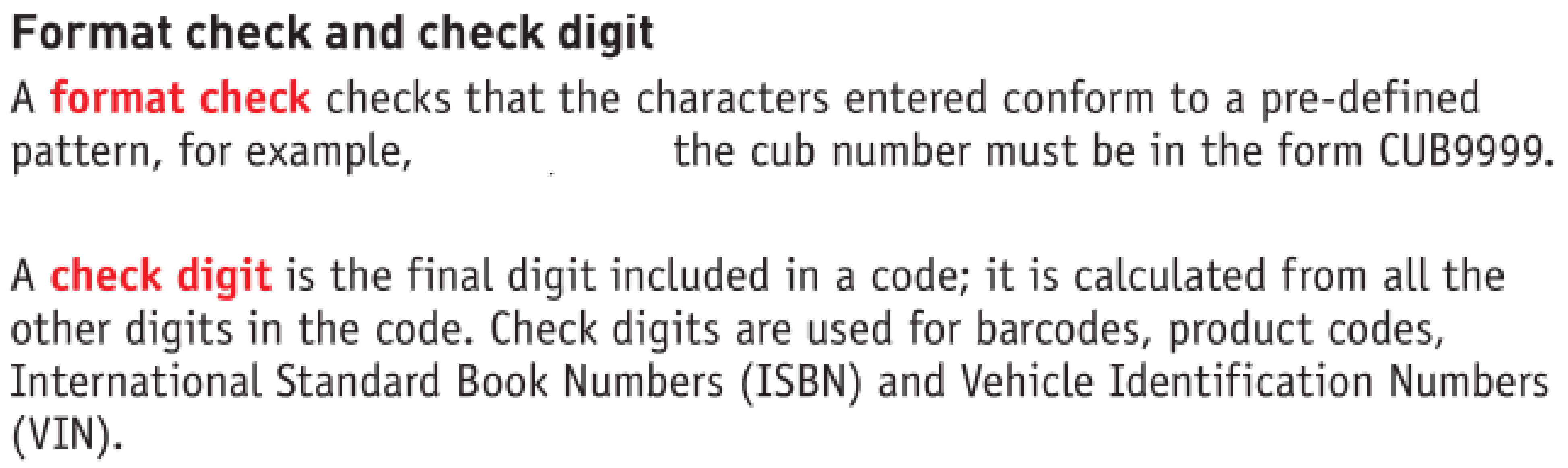


VALIDATION

Format check and check digit

A **format check** checks that the characters entered conform to a pre-defined pattern, for example, the cub number must be in the form CUB9999.

VALIDATION



Format check and check digit

A **format check** checks that the characters entered conform to a pre-defined pattern, for example, the cub number must be in the form CUB9999.

A **check digit** is the final digit included in a code; it is calculated from all the other digits in the code. Check digits are used for barcodes, product codes, International Standard Book Numbers (ISBN) and Vehicle Identification Numbers (VIN).

VALIDATION

Format check and check digit

A **format check** checks that the characters entered conform to a pre-defined pattern, for example, the cub number must be in the form CUB9999.

A **check digit** is the final digit included in a code; it is calculated from all the other digits in the code. Check digits are used for barcodes, product codes, International Standard Book Numbers (ISBN) and Vehicle Identification Numbers (VIN).

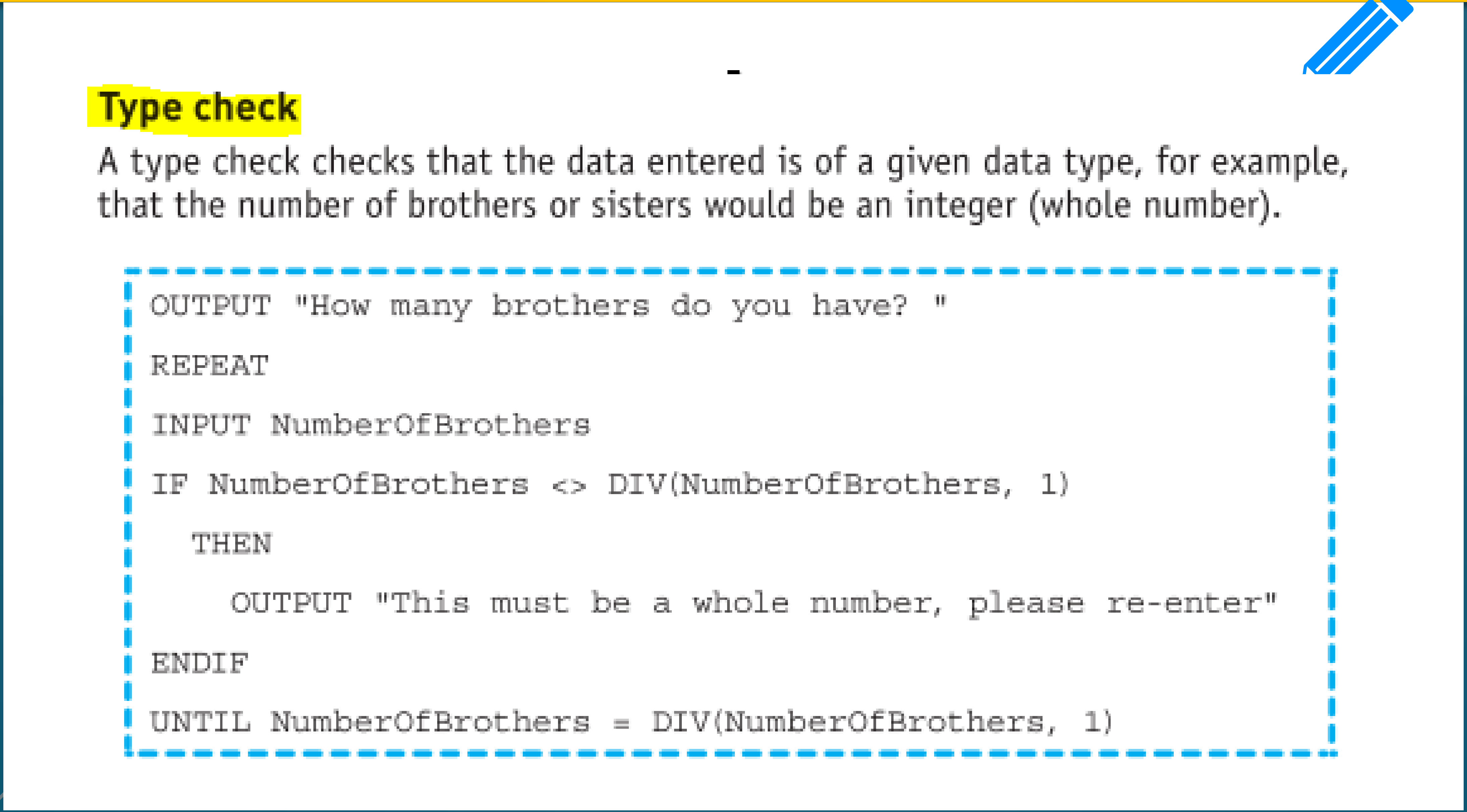
Check digits are used to identify errors in data entry caused by mis-typing or mis-scanning a barcode. They can usually detect the following types of error:

- an incorrect digit entered, for example, 5327 entered instead of 5307
- transposition errors where two numbers have changed order for example 5037 instead of 5307
- omitted or extra digits, for example, 537 instead of 5307 or 53107 instead of 5307
- phonetic errors, for example, 13, thirteen, instead of 30, thirty.

ISBN 978-0-340-98382-9

9 780340 983829

VERIFICATION



Type check

A type check checks that the data entered is of a given data type, for example, that the number of brothers or sisters would be an integer (whole number).

```
OUTPUT "How many brothers do you have? "
REPEAT
INPUT NumberOfBrothers
IF NumberOfBrothers <> DIV(NumberOfBrothers, 1)
  THEN
    OUTPUT "This must be a whole number, please re-enter"
ENDIF
UNTIL NumberOfBrothers = DIV(NumberOfBrothers, 1)
```

VERIFICATION

-

Verification is checking that data has been accurately copied from one source to another – for instance, input into a computer or transferred from one part of a computer system to another.

VERIFICATION

-

Verification is checking that data has been accurately copied from one source to another – for instance, input into a computer or transferred from one part of a computer system to another.

Verification Methods for Input Data

- Double entry
- Screen Visual Check

VERIFICATION

-

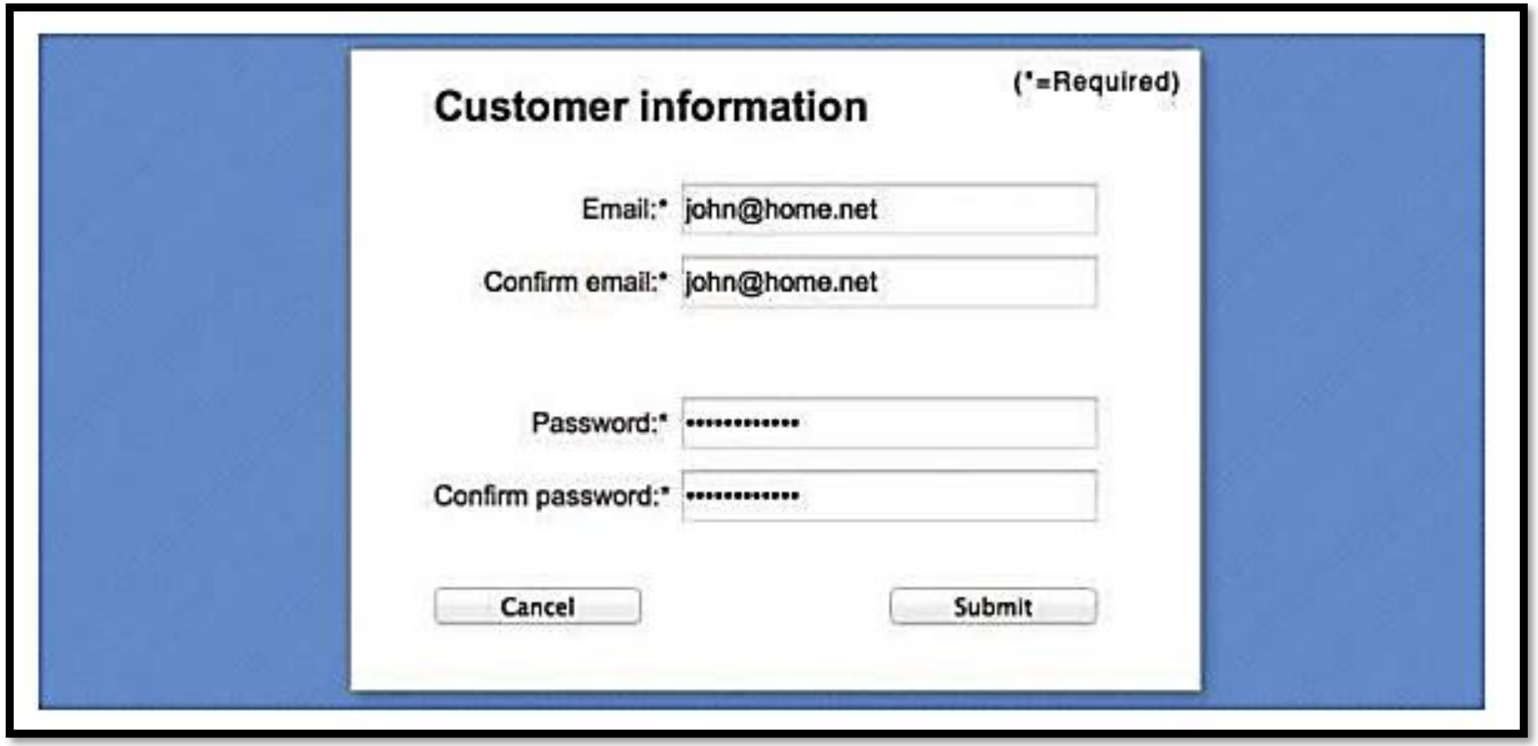
For **double entry** the data is entered twice, sometimes by different operators. The computer system compares both entries and if they are different outputs an error message requesting that the data is entered again.

-

VERIFICATION

-

For **double entry** the data is entered twice, sometimes by different operators. The computer system compares both entries and if they are different outputs an error message requesting that the data is entered again.



Customer information (*=Required)

Email:*	john@home.net
Confirm email:*	john@home.net
Password:*	•••••••••••••
Confirm password:*	•••••••••••••

-

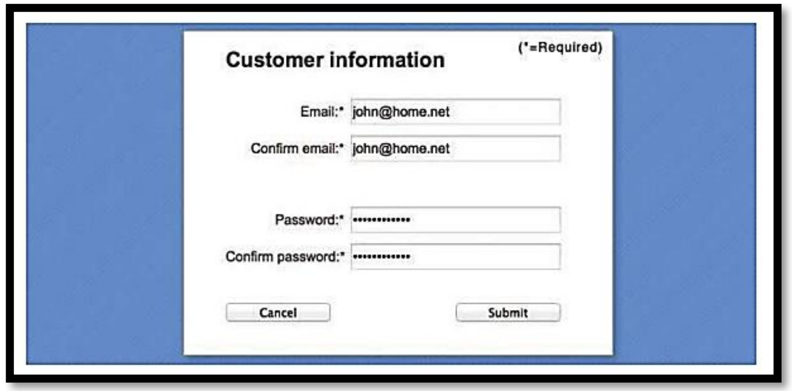
VERIFICATION

-

For **double entry** the data is entered twice, sometimes by different operators. The computer system compares both entries and if they are different outputs an error message requesting that the data is entered again.

A **screen/visual check** is a manual check completed by the user who is entering the data. When the data entry is complete the data is displayed on the screen and the user is asked to confirm that it is correct before continuing. The user either checks the data on the screen against a paper document that is being used as an input form or, confirms whether it is correct from their own knowledge.

-



Customer information (*=Required)

Email:*	john@home.net
Confirm email:*	john@home.net
Password:*	••••••••••••
Confirm password:*	••••••••••••

PRE – DEFINED PYTHON FUNCTIONS

These are some of the pre-defined Python functions that some can be included in the Pseudocode.

Pseudocode	Python
INPUT	input()
PRINT	print()
LENGTH	len()
LIST	list()
STRING	str()
ABSOLUTE	abs()

Python
dict()
divmod()
eval()
float()
int()
isinstance()

Python
max()
min()
next()
pow()
range()
round()

Python
set()
sorted()
sum()
type()
tuple()
ascii()

ACTIVITY

-

Write a Pseudocode to check the length of a password is between 8 to 12 characters inclusive.

ACTIVITY

-

Write a Pseudocode to check the length of a password is 8 characters.

ACTIVITY

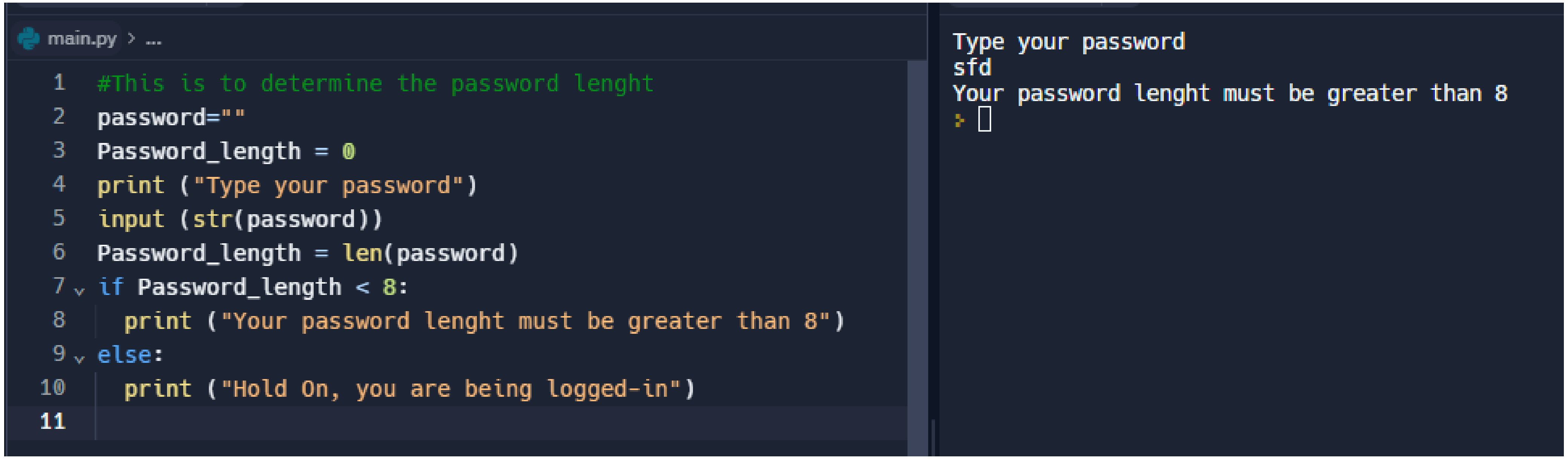
Write a Pseudocode to check the length of a password is 8 characters.

```
Password_length ← 0
INPUT Password
Password_length ← LEN(Password)      // LEN() function is to determine the length
        IF Password_length >=8
                PRINT "Logging you in"
        ELSE
                PRINT "Password must be greater than 8 characters"
        ENDIF
```

PYTHON PROGRAMMING

-

Write a Pseudocode to check the length of a password is 8 characters.



main.py > ...

```python
#This is to determine the password lenght
password=""
Password_length = 0
print ("Type your password")
input (str(password))
Password_length = len(password)
if Password_length < 8:
  print ("Your password lenght must be greater than 8")
else:
  print ("Hold On, you are being logged-in")
```

```
Type your password
sfd
Your password lenght must be greater than 8
› 
```

ACTIVITY

Write a Pseudocode to check the length of a password is 8 and contains a number.

```
# 1. Input the `password` that we plan to validate
password = “Vietn4m2024"
# 2. To keep track of the password length, establish a `pass_length` variable and initially set it to `0`
pass_length = 0
# 3. To keep track of if the password contains a number, establish a `contains_number` variable and initially set it to `False`
contains_number = False
# 4. Has the entire `password` been searched?
while pass_length is not len(password):
  # 5. Iterate to the next character in `password`
  current_character = password[pass_length]
  # 6. Increment `pass_length`
  pass_length = pass_length + 1
  # 7. Is the current character a number?
  if current_character.isdigit():
# If so, set the `contains_number` variable to `True` and then go back to step 4
    contains_number = True
# 8. Is the `pass_length` greater than `8` and is `contain_number` equal to `True`?
if pass_length > 8 and contains_number is True:
  # If so, then the `password` is valid!
  print("Valid Password!")
else:
  # If not, then the `password` is invalid
  print("Invalid Password")
```

PLENARY

-

KNOWING WHAT YOU LEARNED

Go to:

https://joinmyquiz.com

- You are to write your real name and grade.
- Example: Thanh 10G3

COMPUTER SCIENCE

FRITZ EUGENE BANSAG

THANK YOU

- fritz.bansag@vas.edu.vn
- mail@febstar.com