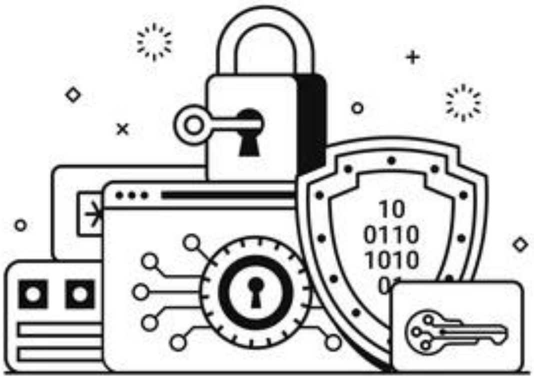
PASSWORD GENERATOR/ CHECKER

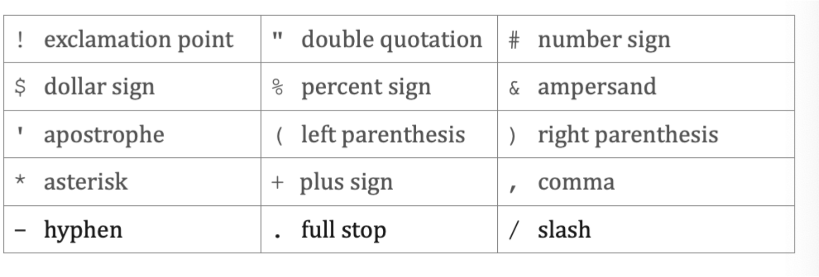
Assessment 3
Sarina Saiyed



Requirements

- To ask the user for information

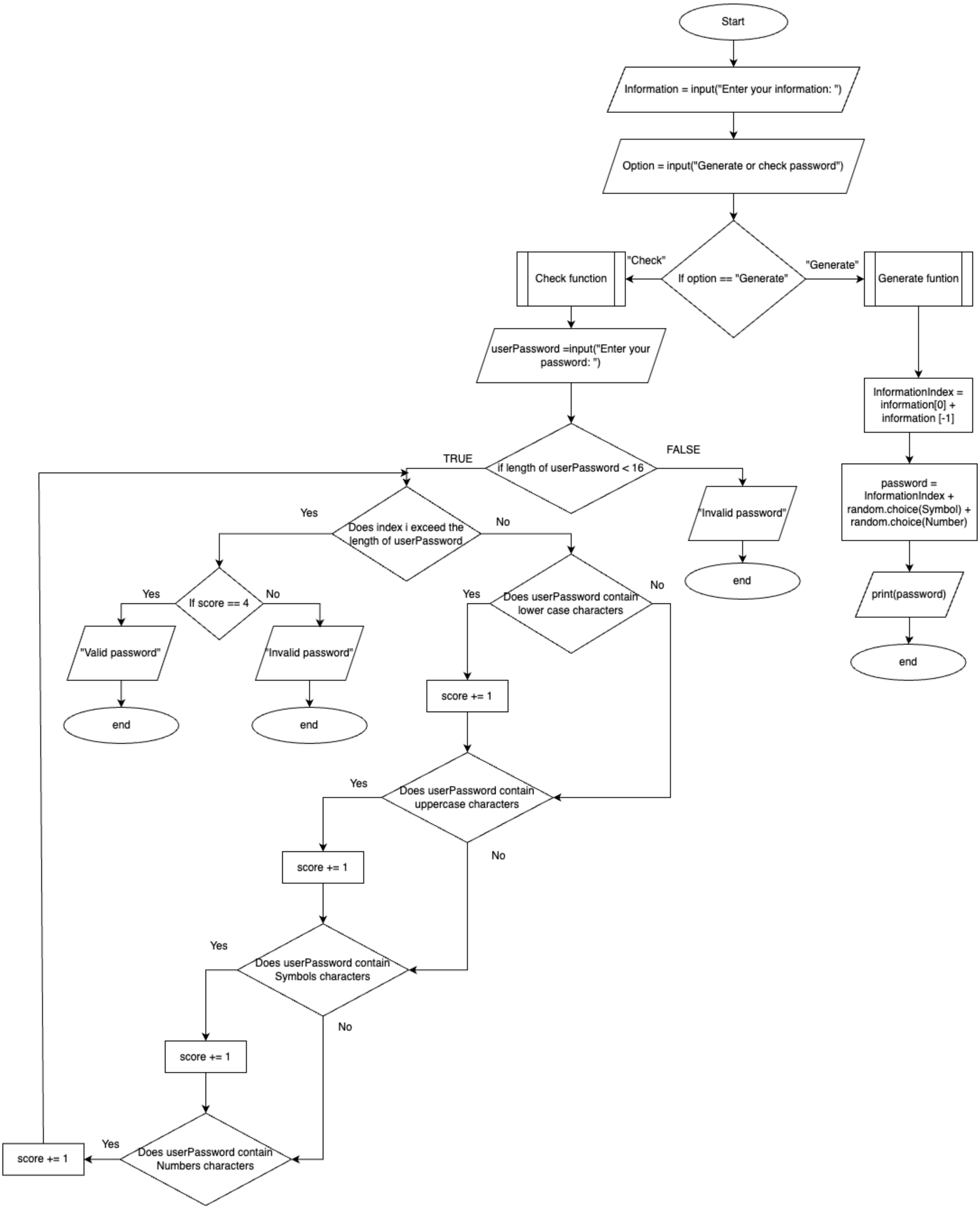
- Give user the choice to generate a password or to check an entered password is valid or not
- If user picks to generate a password
  - Use the information given by the user to create a password base
  - Password must include:
    - A set length 15 character long
    - Include upper- and lower-case characters (letters)
    - Number between 0-9
    - A random generated symbol from:



! exclamation point	" double quotation	# number sign
$ dollar sign	% percent sign	& ampersand
' apostrophe	(left parenthesis	) right parenthesis
* asterisk	+ plus sign	, comma
- hyphen	. full stop	/ slash

- If the user chooses to check the validity of their password:
  - Check if password:
    - Is within the set length
    - Contain upper and lowercase characters
    - Contains a number between 0-9
    - Contains a symbol
  - If the user's password contains all the requirements, output that it is a valid password
  - If the user's password contains two to four of the requirements, output that the password is invalid and output how to improve their password
  - If the user’s password contains
  - one of the requirements, output that that the password is invalid and output all the requirements needed to improve their password

Flow chart


Start
Information = input("Enter your information: ")
Option = input("Generate or check password")
If option == "Generate"
"Check"
"Generate"
Check function
Generate funtion
userPassword =input("Enter your password: ")
InformationIndex = information[0] + information [-1]
password = InformationIndex + random.choice(Symbol) + random.choice(Number)
print(password)
end
if length of userPassword < 16
TRUE
FALSE
"Invalid password"
end
Does index i exceed the length of userPassword
Yes
No
If score == 4
Yes
No
"Valid password"
"Invalid password"
end
end
Does userPassword contain lower case characters
Yes
No
score += 1
Does userPassword contain uppercase characters
Yes
No
score += 1
Does userPassword contain Symbols characters
Yes
No
score += 1
Does userPassword contain Numbers characters
Yes
score += 1

Pseudocode

```
IMPORT random
```

```
DEF passwordGenerator(Information):
InformationIndex = information[0].capitalize() + information[-1]
    Password = InformationIndex + random.choice(Symbol) +
random.choice(Symbol) + STR(random.choice(Number))
    RETURN Password


DEF passwordChecker(Information):
    userPassword = INPUT("Enter your password: ")
    score = 0
    IF LEN(userPassword) < 15:
        For I in userPassword:
            IF (i.islower()):
                Score += 1
            IF (i.isupper()):
                Score += 1
            IF (i.isdigit()):
                Score += 1
            IF (I in Symbol):
                Score += 1
ENDIF
END FOR LOOP
    ELSE:
        Print("Password is too long, Password characters must be equal
or less than 15")
END IF
    IF Score == 5:
        PRINT("Valid password")
    ELIF 1 <= Score <= 4:
PRINT("Invalid password")
END IF
```

```
Information = INPUT(“Enter your information: ”)

Number = [0, 1, 2, 3,4, 5, 6, 7, 8, 9]

Symbol = [“!”,””””,”#”,“$”,”%”,”&”,“’”,”(”,”)”,“*”,”+”,”,”,“-”,”.”,”/”]


Option = INPUT(“Would you like to ‘generate’ a password or ‘Check’ your
password? \n”)

WHILE Option =! ‘generate’ or Option =! ‘check’:

      IF Option == ‘generate’:

            passwordGenerator(Information)

            PRINT(passwordGenerator(Information))

      ELIF Option == ‘check’:

            passwordChecker(Information)

      ELSE:

            PRINT(“Please write a valid input.”)

END IF

END WHILE LOOP
```

Code:

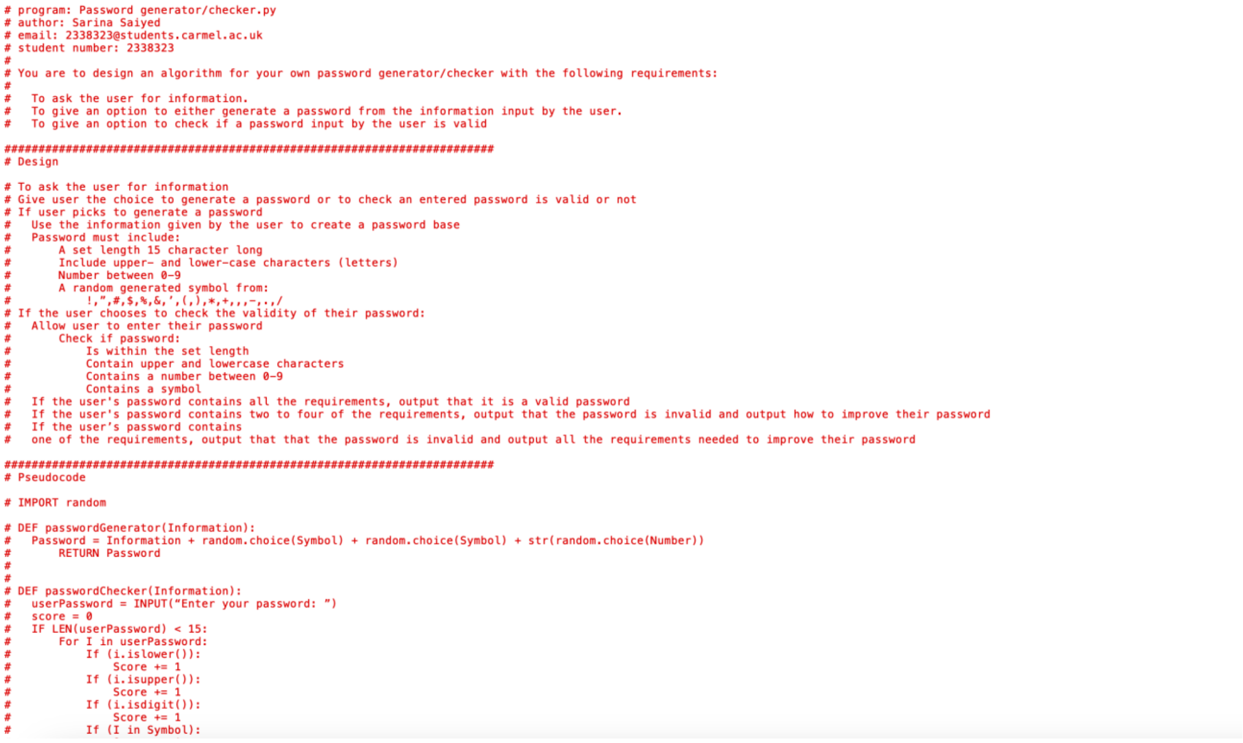
```
# program: Password generator/checker.py

# author: Sarina Saiyed
# email: 2338323@students.carmel.ac.uk
# student number: 2338323
#
# You are to design an algorithm for your own password generator/checker with the following requirements:
#
#   To ask the user for information.
#   To give an option to either generate a password from the information input by the user.
#   To give an option to check if a password input by the user is valid

########################################################################
# Design

# To ask the user for information
# Give user the choice to generate a password or to check an entered password is valid or not
# If user picks to generate a password
#   Use the information given by the user to create a password base
#   Password must include:
#       A set length 15 character long
#       Include upper- and lower-case characters (letters)
#       Number between 0-9
#       A random generated symbol from:
#           !,”,#,$,%,&,’,(,),*,+,,,-,.,/
# If the user chooses to check the validity of their password:
#   Allow user to enter their password
#       Check if password:
#           Is within the set length
#           Contain upper and lowercase characters
#           Contains a number between 0-9
#           Contains a symbol
#   If the user's password contains all the requirements, output that it
is a valid password

#   If the user's password contains two to four of the requirements,
output that the password is invalid and output how to improve their
password

#   If the user’s password contains

#   one of the requirements, output that that the password is invalid and
output all the requirements needed to improve their password


########################################################################

# Pseudocode


# IMPORT random


# DEF passwordGenerator(Information):

# informationIndex = information[0].capitalize() + information[-1]

# Password = InformationIndex + random.choice(Symbol) +
random.choice(Symbol) + STR(random.choice(Number))

#       RETURN Password

#

#

# DEF passwordChecker(Information):

#   userPassword = INPUT(“Enter your password: ”)

#   score = 0

#   IF LEN(userPassword) < 15:

#       For I in userPassword:

#           IF (i.islower()):

#               Score += 1

#           IF (i.isupper()):

#               Score += 1

#           IF (i.isdigit()):

#               Score += 1

#           IF (I in Symbol):

#               Score += 1
#           ENDIF
#     END FOR LOOP
#   ELSE:
#       Print(“Password is too long, Password characters must be equal or
less than 15”)
#   END IF
#
#   IF Score == 5:
#       PRINT(“Valid password”)
#   ELIF 1 <= Score <= 4:
#     PRINT(“Invalid password”)
#   END IF
#
#
#
# Information = INPUT(“Enter your information: ”)
# Number = [0, 1, 2, 3,4, 5, 6, 7, 8, 9]
# Symbol = [“!”,”””,”#”,“$”,”%”,”&”,“’”,”(”,”)”,“*”,”+”,”,”,“-”,”.”,”/”]

# Option = INPUT(“Would you like to ‘generate’ a password or ‘Check’ your
password? \n”)
# WHILE Option =! ‘generate’’ or Option =! ‘check’’:
#   IF Option == ‘generate’:
#       passwordGenerator(Information)
#       PRINT(passwordGenerator(Information))
#   ELIF Option == ‘check’:
#       passwordChecker(Information)
#   ELSE:
#       PRINT(“Please write a valid input.”)
#     END IF
# END WHILE LOOP

##########################################################################
```

```
# Variables

# STR AnimalIndex
# STR ColourIndex
# STR password
# STR userPassword
# INT lower
# INT upper
# INT digit
# INT symbol
# STR InformationAnimal
# STR InformationColour
# LIST Number
# LIST Symbol
# STR option

########################################################################
# Functions

# passwordGenerator
# passwordChecker

########################################################################
# Main

import random # imports the libary random



def passwordGenerator(InformationAnimal,InformationColour): # funtion for generating a password for the user
  animalIndex = InformationAnimal[0].capitalize() + InformationAnimal[-1] #takes the first and last character of animal information and joins them
  colourIndex = InformationColour[0].capitalize() + InformationColour[1]
#takes first two characters of colour information

  password =  animalIndex + random.choice(Symbol) + colourIndex +
random.choice(Symbol) + str(random.choice(Number)) # variable that

  # takes the input information and capitalises it as well as joining the
random chosen numbers/ symbols to create a valid password

  return password # return the variable password for the user




def passwordChecker(): # funtion for checking an input password

    userPassword = input("Enter your password: ") # input variable

    lower, upper, digit, symbol = 0, 0 ,0 , 0 # four different variables
set to 0

    if len(userPassword) < 16: # checks that the users password is not
exceeding 15 characters

        for i in userPassword: # for the index i of the characters in
userPassword; checks each character and goes to the next

            # the loop ends when it reaches the last character

            if (i.islower()): # checks if current index is a lower case
character

                lower += 1 # if there is, variable lower is incremented by
1

            if (i.isupper()): # checks if current index is a upper case
character

                upper += 1 # if there is, variable upper is incremented by
1

            if (i.isdigit()): # checks if current index is a digit
character

                digit += 1 # if there is, variable digit is incremented by
1

            if (i in Symbol): # checks if current index is a symbol
character

                symbol += 1 # if there is, variable symbol is incremented
by 1

            # does everthing again with the next index in userPassword
    if (lower>=1 and upper >= 1 and digit >= 1 and symbol>=1 and lower+upper+digit+symbol==len(userPassword)): # and when all variables have run through the length of userPassword
        print("Valid input") # states valid input only if the all the requirements are more than or equal to one ^
    else:
        print("""Invalid input, make sure your password includes:
- Upper and lowercase characters
- digits
- symbols
- is less than 15 characters""") # if any requrirements = 0, states invalid password and states how to imporve the password




InformationAnimal = input("Enter an animal (Up to 5 characters long): ") # input variable of first information for the generator function
InformationColour = input("Enter a colour (Up to 5 characters long): ") # input variable of the second information for the generator function

while len(InformationAnimal) > 5 or (len(InformationColour) > 5): # while loop that repeats if length of input varibles are more than 5 characters
    if len(InformationAnimal) > 5: # only repeats the animal information input
        print("Animal information should be 5 characters or less")
        InformationAnimal = input("Enter an animal (Up to 5 characters long): ")

    elif len(InformationColour) > 5: # only repeats the colour information input
        print("Colour information should be 6 characters or less")
        InformationColour = input("Enter a colour (Up to 5 characters long): ")
    # outputs both if both exceed 5 characters
```

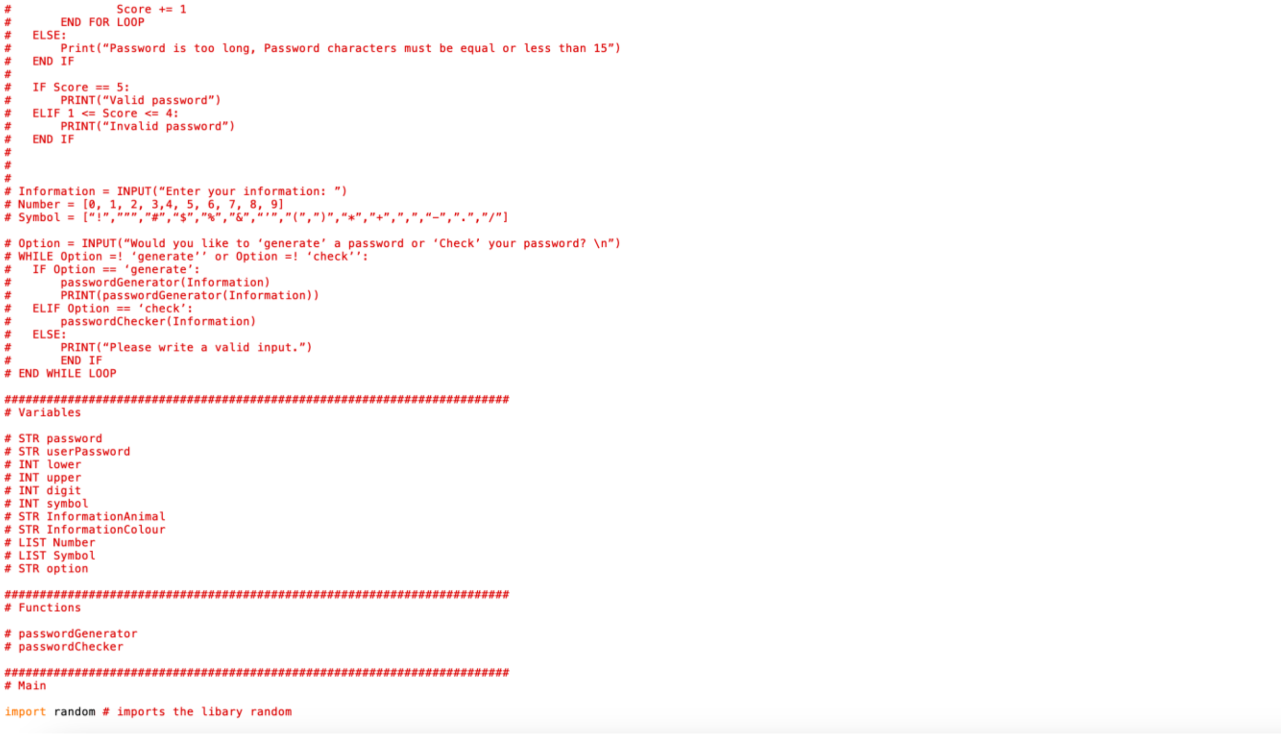
```
Number = [0, 1, 2, 3,4, 5, 6, 7, 8, 9]  # stored list of numbers used for
the generator function

Symbol = ["!","”","#","$","%","&","’","(",")","*","+",",","-",".","/"]  #
stored list of valid symbols used for the generator function
```

```
option = input("Would you like to ‘generate’ a password or ‘check’ your
password? \n") # allows user to either generate or check their password

while (option != 'generate' or option != 'check'): # while loop that
allows the user to re-enter their options if they have chosen wrong

    if option == 'generate':

        passwordGenerator(InformationAnimal,InformationColour) # sends
user to the passwordGenerator function

        print(passwordGenerator(InformationAnimal,InformationColour)) #
prints out the generated password using the users information

        break # ends whiile loop

    elif option == 'check':

        passwordChecker() #send user to the passwordChecker funtion

        break

    else:

        print("Please enter a valid input") #states that the input is
invalid

        option = input("Would you like to ‘generate’ a password or ‘check’
your password? \n") # allows user to re-enter their option
```

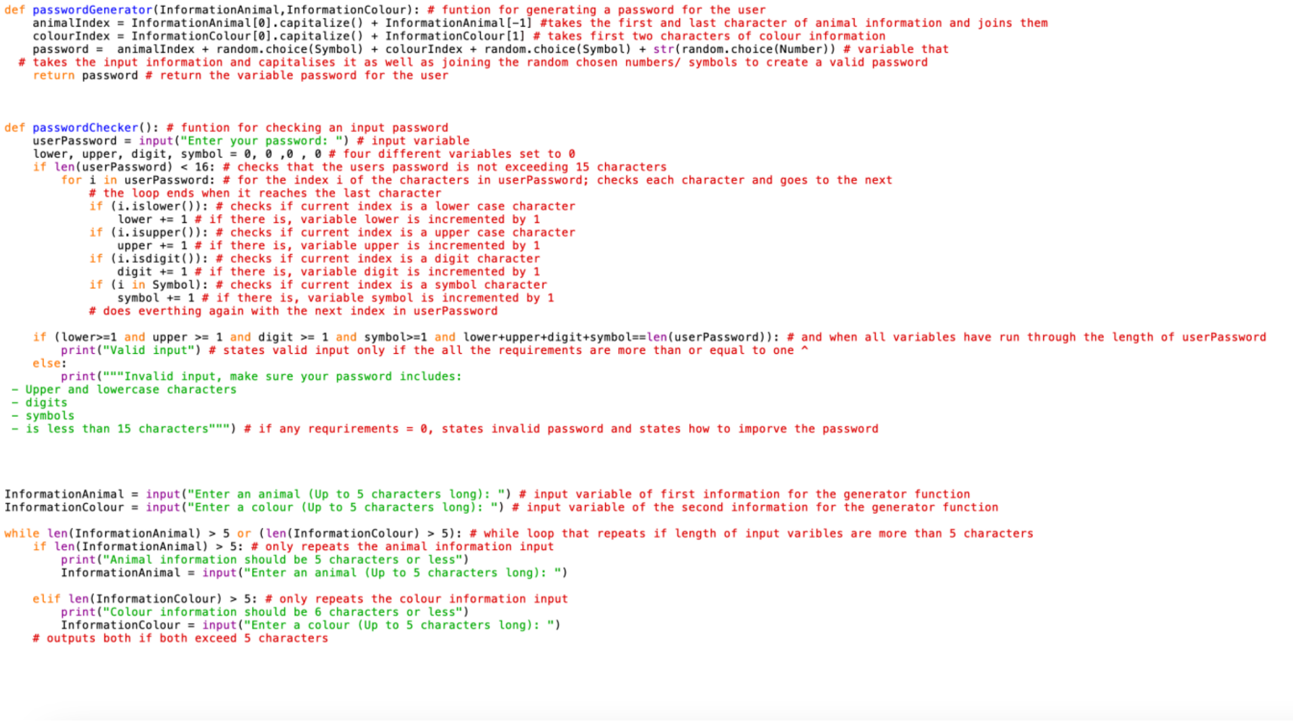
Code Screenshot



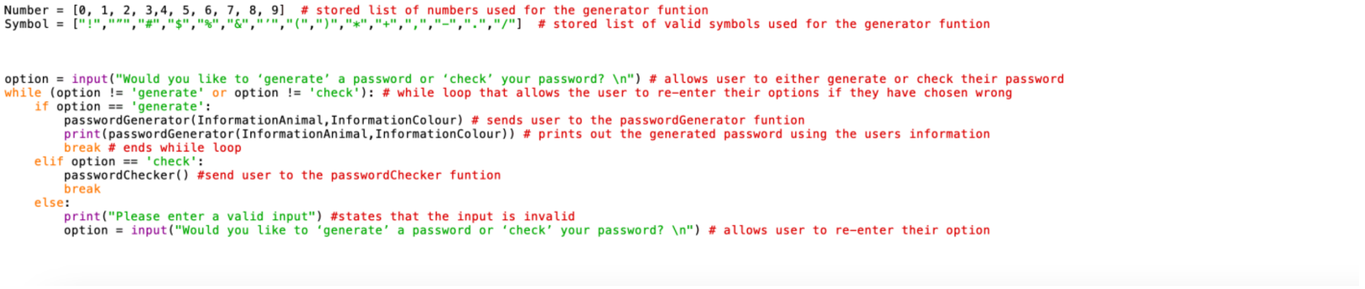
```
# program: Password generator/checker.py
# author: Sarina Saiyed
# email: 2338323@students.carmel.ac.uk
# student number: 2338323
#
# You are to design an algorithm for your own password generator/checker with the following requirements:
#
#   To ask the user for information.
#   To give an option to either generate a password from the information input by the user.
#   To give an option to check if a password input by the user is valid

########################################################################
# Design

# To ask the user for information
# Give user the choice to generate a password or to check an entered password is valid or not
# If user picks to generate a password
#   Use the information given by the user to create a password base
#   Password must include:
#       A set length 15 character long
#       Include upper- and lower-case characters (letters)
#       Number between 0-9
#       A random generated symbol from:
#           !,”,#,$,%,&,’,(,),*,+,,,-,.,/
# If the user chooses to check the validity of their password:
#   Allow user to enter their password
#       Check if password:
#           Is within the set length
#           Contain upper and lowercase characters
#           Contains a number between 0-9
#           Contains a symbol
#   If the user's password contains all the requirements, output that it is a valid password
#   If the user's password contains two to four of the requirements, output that the password is invalid and output how to improve their password
#   If the user’s password contains
#   one of the requirements, output that that the password is invalid and output all the requirements needed to improve their password

########################################################################
# Pseudocode

# IMPORT random

# DEF passwordGenerator(Information):
#   Password = Information + random.choice(Symbol) + random.choice(Symbol) + str(random.choice(Number))
#       RETURN Password
#
#
# DEF passwordChecker(Information):
#   userPassword = INPUT(“Enter your password: ”)
#   score = 0
#   IF LEN(userPassword) < 15:
#       For I in userPassword:
#           If (i.islower()):
#               Score += 1
#           If (i.isupper()):
#               Score += 1
#           If (i.isdigit()):
#               Score += 1
#           If (I in Symbol):
```



```
#                Score += 1
#        END FOR LOOP
#    ELSE:
#        Print(“Password is too long, Password characters must be equal or less than 15”)
#    END IF
#
#    IF Score == 5:
#        PRINT(“Valid password”)
#    ELIF 1 <= Score <= 4:
#        PRINT(“Invalid password”)
#    END IF
#
#
#
# Information = INPUT(“Enter your information: ”)
# Number = [0, 1, 2, 3,4, 5, 6, 7, 8, 9]
# Symbol = [“!”,”””,”#”,“$”,”%”,”&”,“'”,”(“,”)”,“*”,”+”,”,”,“-”,”.”,”/”]

# Option = INPUT(“Would you like to ‘generate’ a password or ‘Check’ your password? \n”)
# WHILE Option =! ‘generate’’ or Option =! ‘check’’:
#    IF Option == ‘generate’:
#        passwordGenerator(Information)
#        PRINT(passwordGenerator(Information))
#    ELIF Option == ‘check’:
#        passwordChecker(Information)
#    ELSE:
#        PRINT(“Please write a valid input.”)
#        END IF
# END WHILE LOOP

#########################################################################
# Variables

# STR password
# STR userPassword
# INT lower
# INT upper
# INT digit
# INT symbol
# STR InformationAnimal
# STR InformationColour
# LIST Number
# LIST Symbol
# STR option

#########################################################################
# Functions

# passwordGenerator
# passwordChecker

#########################################################################
# Main

import random # imports the libary random
```



```
def passwordGenerator(InformationAnimal,InformationColour): # funtion for generating a password for the user
    animalIndex = InformationAnimal[0].capitalize() + InformationAnimal[-1] #takes the first and last character of animal information and joins them
    colourIndex = InformationColour[0].capitalize() + InformationColour[1] # takes first two characters of colour information
    password =  animalIndex + random.choice(Symbol) + colourIndex + random.choice(Symbol) + str(random.choice(Number)) # variable that
  # takes the input information and capitalises it as well as joining the random chosen numbers/ symbols to create a valid password
    return password # return the variable password for the user


def passwordChecker(): # funtion for checking an input password
    userPassword = input("Enter your password: ") # input variable
    lower, upper, digit, symbol = 0, 0 ,0 , 0 # four different variables set to 0
    if len(userPassword) < 16: # checks that the users password is not exceeding 15 characters
        for i in userPassword: # for the index i of the characters in userPassword; checks each character and goes to the next
            # the loop ends when it reaches the last character
            if (i.islower()): # checks if current index is a lower case character
                lower += 1 # if there is, variable lower is incremented by 1
            if (i.isupper()): # checks if current index is a upper case character
                upper += 1 # if there is, variable upper is incremented by 1
            if (i.isdigit()): # checks if current index is a digit character
                digit += 1 # if there is, variable digit is incremented by 1
            if (i in Symbol): # checks if current index is a symbol character
                symbol += 1 # if there is, variable symbol is incremented by 1
            # does everthing again with the next index in userPassword

    if (lower>=1 and upper >= 1 and digit >= 1 and symbol>=1 and lower+upper+digit+symbol==len(userPassword)): # and when all variables have run through the length of userPassword
        print("Valid input") # states valid input only if the all the requirements are more than or equal to one ^
    else:
        print("""Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters""") # if any requriements = 0, states invalid password and states how to imporve the password



InformationAnimal = input("Enter an animal (Up to 5 characters long): ") # input variable of first information for the generator function
InformationColour = input("Enter a colour (Up to 5 characters long): ") # input variable of the second information for the generator function

while len(InformationAnimal) > 5 or (len(InformationColour) > 5): # while loop that repeats if length of input varibles are more than 5 characters
    if len(InformationAnimal) > 5: # only repeats the animal information input
        print("Animal information should be 5 characters or less")
        InformationAnimal = input("Enter an animal (Up to 5 characters long): ")

    elif len(InformationColour) > 5: # only repeats the colour information input
        print("Colour information should be 6 characters or less")
        InformationColour = input("Enter a colour (Up to 5 characters long): ")
    # outputs both if both exceed 5 characters
```



```
Number = [0, 1, 2, 3,4, 5, 6, 7, 8, 9]  # stored list of numbers used for the generator funtion
Symbol = ["!","”","#","$","%","&","'","(",")","*","+",",","-",".","/"]  # stored list of valid symbols used for the generator funtion


option = input("Would you like to ‘generate’ a password or ‘check’ your password? \n") # allows user to either generate or check their password
while (option != 'generate' or option != 'check'): # while loop that allows the user to re-enter their options if they have chosen wrong
    if option == 'generate':
        passwordGenerator(InformationAnimal,InformationColour) # sends user to the passwordGenerator funtion
        print(passwordGenerator(InformationAnimal,InformationColour)) # prints out the generated password using the users information
        break # ends whiile loop
    elif option == 'check':
        passwordChecker() #send user to the passwordChecker funtion
        break
    else:
        print("Please enter a valid input") #states that the input is invalid
        option = input("Would you like to ‘generate’ a password or ‘check’ your password? \n") # allows user to re-enter their option
```

Testing:

*All screenshots at bottom of test table

Test Number	Description of test	Test Data	Expected outcome	Actual Outcome
1.0	Generating a password	- cat - pink - generate	Ct(random symbol) Pi (random symbol) (random number)	Ct(Pi%8
1.1	Generate a password when the first information exceeds 5 characters (6 characters)	- kitten - pink - generate	Input should be 5 characters or less Enter an animal (Up to 5 characters long):	Kn%Pi$8
1.2	Generate a password when the first information exceeds 5 characters (6 characters)	- kitten - pink - cat - generate	Input should be 5 characters or less Enter an animal (Up to 5 characters long):	Animal information should be 5 characters or less Enter an animal (Up to 5 characters long): cat Would you like to 'generate' a password or 'check' your password? generate Ct+Pi'6
1.3	Generate a password when the second information exceeds 5 characters (6 characters)	- cat - yellow - pink - generate	Input should be 5 characters or less Enter a colour (Up to 5 characters long):	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): yellow Colour information should be 6 characters or less Enter a colour (Up to 5 characters long): pink Would you like to 'generate' a password or 'check' your password?

				generate Ct/Pi)5

1.4	Generate a password when the first information and the second information exceed 5 characters (6 characters)	- kitten
		- yellow
		- cat
		- pink
		- generate
			Input should be 5 characters or less
			Input should be 5 characters or less
			Enter an animal (Up to 5 characters long):
			Enter a colour (Up to 5 characters long):
				Enter an animal (Up to 5 characters long): kitten
				Enter a colour (Up to 5 characters long): yellow
				Animal information should be 5 characters or less
				Enter an animal (Up to 5 characters long): cat
				Colour information should be 6 characters or less
				Enter a colour (Up to 5 characters long): pink
				Would you like to 'generate' a password or 'check' your password?
				generate
				Ct!Pi#5

2.0	Checking a completely valid password	- cat - pink - check - Cat.Pink/9	- Valid input	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to 'generate' a password or 'check' your password? check Enter your password: Cat.Pink/9 Valid input
2.1	Checking a password that does not contains lower case characters	- cat - pink - check - CAT.PINK/9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to 'generate' a password or 'check' your password? check Enter your password: CAT.PINK/9 Invalid input, make sure your password includes: - Upper and lowercase

				characters - digits - symbols - is less than 15 characters
2.2	Checking a password that does not contain uppercase characters	- cat - pink - check - cat.pink/9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? check Enter your password: cat.pink/9 Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters
2.3	Checking a password that does not contain digits	- cat - pink - check - Cat.Pink/	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? check Enter your password: Cat.Pink/ Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters
2.4	Checking a password that does not contain symbols	- cat - pink - check - CatPink9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a

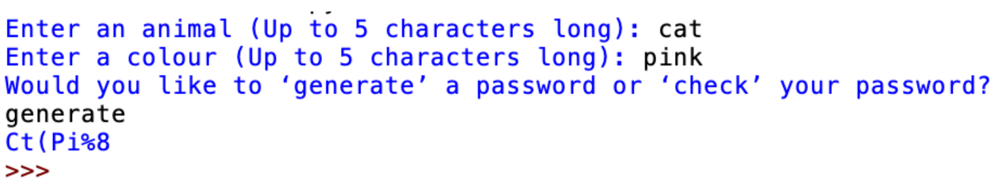
			- is less than 15 characters	password or ‘check’ your password? check Enter your password: CatPink9 Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters
2.5	Checking a password with an invalid symbol	- cat - pink - check - Cat£Pink9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? check Enter your password: Cat£Pink9 Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters
2.6	Checking a password with both a valid and invalid symbol	- cat - pink - check - Cat£Pink.9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? check Enter your password: Cat£Pink.9 Invalid input, make sure your password includes: - Upper and lowercase characters - digits

				- symbols - is less than 15 characters
2.7	Checking a password that exceed 15 characters	- cat - pink - check - Elephant.Yellow/9	Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? check Enter your password: Elephant.Yellow/9 Invalid input, make sure your password includes: - Upper and lowercase characters - digits - symbols - is less than 15 characters
3.0	Enter another input beside ‘generate’ or check	- cat - pink - x - generate	Please enter a valid input Would you like to ‘generate’ a password or ‘check’ your password?	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? x Please enter a valid input Would you like to ‘generate’ a password or ‘check’ your password? generate Ct(Pi,3
3.1	Enter an incorrect input twice	- cat - pink - x - y - generate	Please enter a valid input Would you like to ‘generate’ a password or ‘check’ your password? Please enter a valid input Would you like to ‘generate’ a password or ‘check’ your	Enter an animal (Up to 5 characters long): cat Enter a colour (Up to 5 characters long): pink Would you like to ‘generate’ a password or ‘check’ your password? x Please enter a valid input Would you like to

			password?	‘generate’ a password or ‘check’ your password? y Please enter a valid input Would you like to ‘generate’ a password or ‘check’ your password? generate Ct.Pi+7

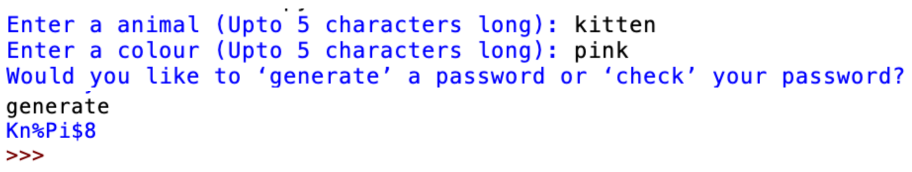
Screen shot testing

1.0



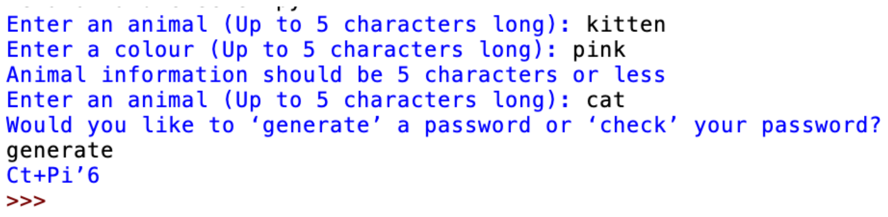
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct(Pi%8
>>>
```

1.1

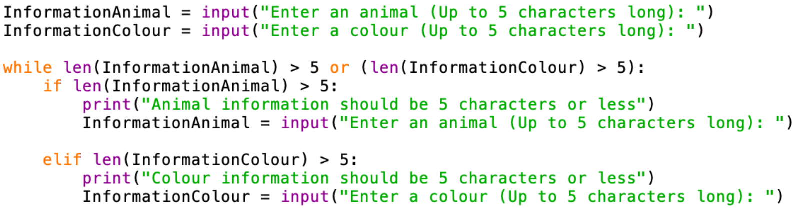


```
Enter a animal (Upto 5 characters long): kitten
Enter a colour (Upto 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Kn%Pi$8
>>>
```

1.2

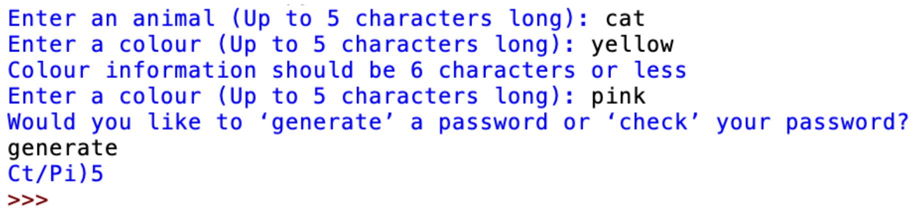


```
Enter an animal (Up to 5 characters long): kitten
Enter a colour (Up to 5 characters long): pink
Animal information should be 5 characters or less
Enter an animal (Up to 5 characters long): cat
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct+Pi’6
>>>
```



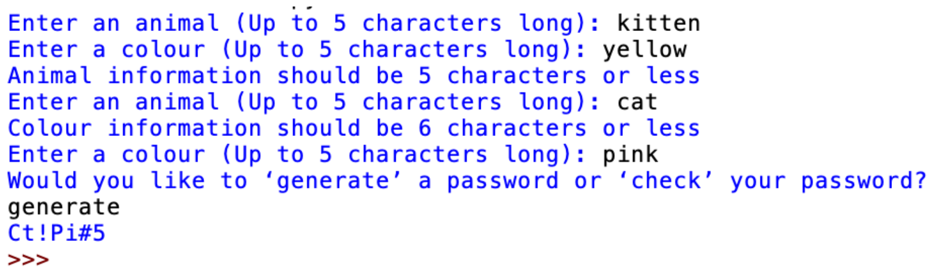
```
InformationAnimal = input("Enter an animal (Up to 5 characters long): ")
InformationColour = input("Enter a colour (Up to 5 characters long): ")

while len(InformationAnimal) > 5 or (len(InformationColour) > 5):
    if len(InformationAnimal) > 5:
        print("Animal information should be 5 characters or less")
        InformationAnimal = input("Enter an animal (Up to 5 characters long): ")

    elif len(InformationColour) > 5:
        print("Colour information should be 5 characters or less")
        InformationColour = input("Enter a colour (Up to 5 characters long): ")
```

1.3



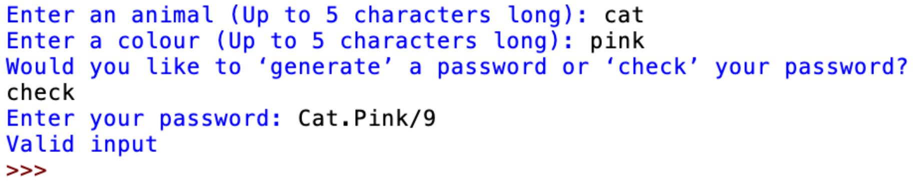
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): yellow
Colour information should be 6 characters or less
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct/Pi)5
>>>
```

1.4



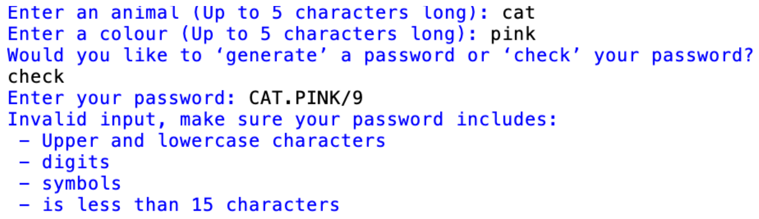
```
Enter an animal (Up to 5 characters long): kitten
Enter a colour (Up to 5 characters long): yellow
Animal information should be 5 characters or less
Enter an animal (Up to 5 characters long): cat
Colour information should be 6 characters or less
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct!Pi#5
>>>
```

2.0



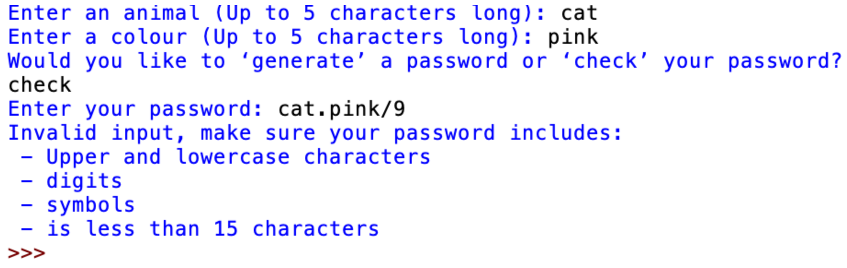
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: Cat.Pink/9
Valid input
>>>
```

2.1



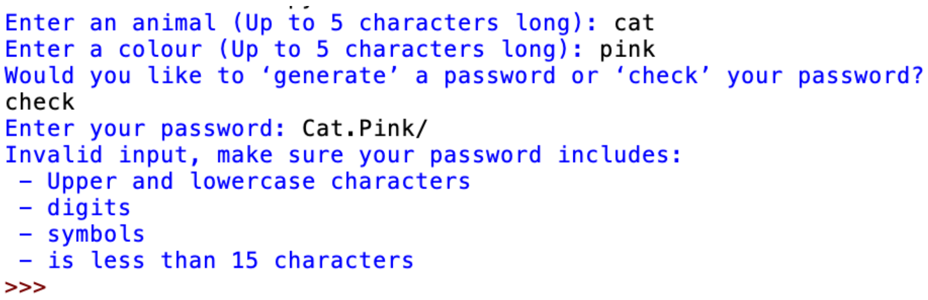
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: CAT.PINK/9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
```

2.2



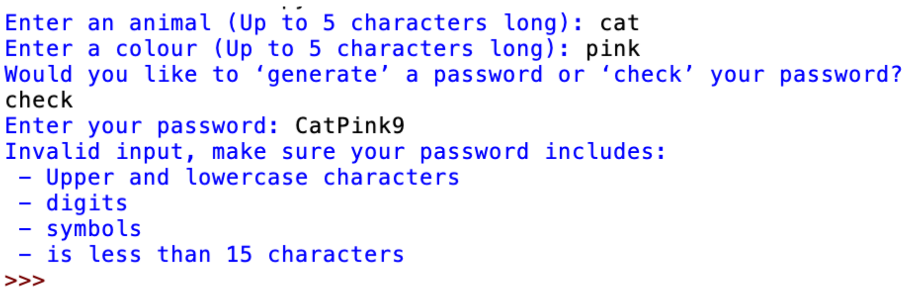
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: cat.pink/9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

2.3



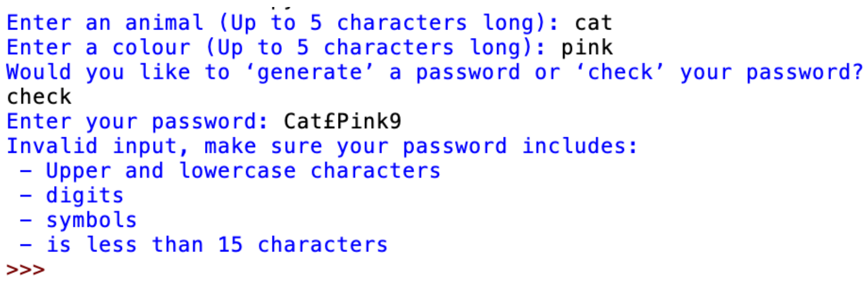
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: Cat.Pink/
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

2.4



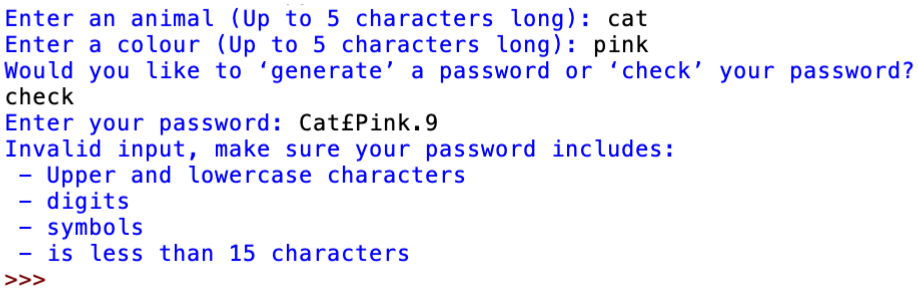
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: CatPink9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

2.5



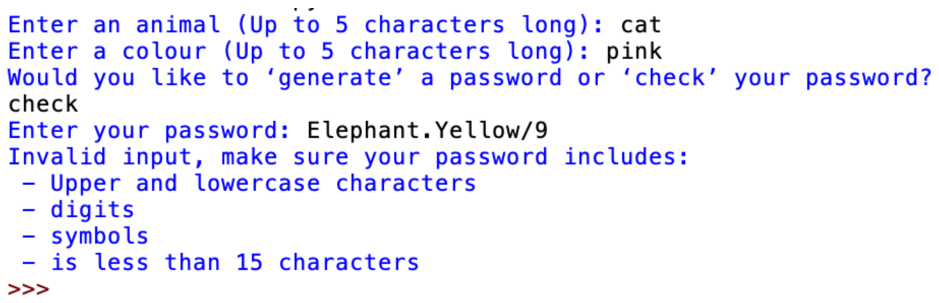
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: Cat£Pink9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

2.6



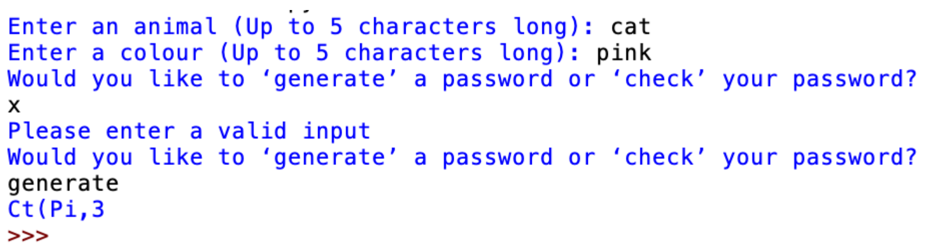
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: Cat£Pink.9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

2.7



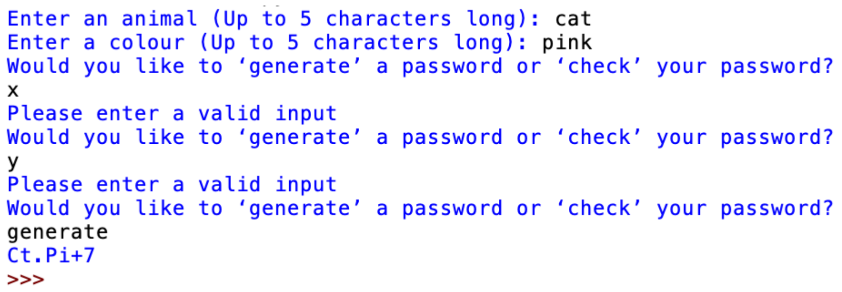
```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
check
Enter your password: Elephant.Yellow/9
Invalid input, make sure your password includes:
 - Upper and lowercase characters
 - digits
 - symbols
 - is less than 15 characters
>>>
```

3.0



```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
x
Please enter a valid input
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct(Pi,3
>>>
```

3.1



```
Enter an animal (Up to 5 characters long): cat
Enter a colour (Up to 5 characters long): pink
Would you like to ‘generate’ a password or ‘check’ your password?
x
Please enter a valid input
Would you like to ‘generate’ a password or ‘check’ your password?
y
Please enter a valid input
Would you like to ‘generate’ a password or ‘check’ your password?
generate
Ct.Pi+7
>>>
```

References

https://www.w3schools.com/python/python_ref_string.asp https://pynative.com/python-random-choice/

https://www.geeksforgeeks.org/python-program-check-validity-password/