```python
import random
from os import system

def pass_generation(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

def main():
    password = ''
    while (True):
        try:
            pass_length = int(input('Enter the desired password length (at least 8 characters): '))
            if (pass_length < 8):
                system('CLS')
                print('Invalid length, please enter length with at least 8 characters')
                continue
            break
        except:
            system('CLS')
            print('Please enter a valid, positive integer')

    password = ''.join(pass_generation(pass_length))
    print (password)
    print (len(password))

def isoMain():
    password = ''
    while (True):
        try:
            pass_length = int(input('Enter the desired secret length (at least 8 characters): '))
            if (pass_length < 8):
                system('CLS')
                print('Invalid length, please enter length with at least 8 characters')
                continue
            break
        except:
            system('CLS')
            print('Please enter a valid, positive integer')

    password = ''.join(pass_generation(pass_length))
    print (password)
    print (len(password))

def pass_generation2(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

def pass_generation3(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

def pass_generation4(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
```

```
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

def pass_generation5(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

def pass_generation6(length):
    pass_components = []
    # This block ensures that we have a minimum of 1 upper case, 1 lower case, 1 symbol, and 1 number
    pass_components.append(chr(random.randint(65, 90)))  #Uppercase
    pass_components.append(chr(random.randint(97, 112))) #Lowercase
    pass_components.append(chr(random.randint(33, 41)))  #Symbol
    pass_components.append(chr(random.randint(48, 57)))  #Number
    # Fill the rest of the list with random components
    for i in range(length - 4):
        character = random.randint(1 , 4)
        match character:
            case 1 :
                pass_components.append(chr(random.randint(65, 90)))  #Uppercase
            case 2:
                pass_components.append(chr(random.randint(97, 112))) #Lowercase
            case 3:
                pass_components.append(chr(random.randint(33, 41)))  #Symbol
            case 4:
                pass_components.append(chr(random.randint(48, 57)))  #Number
    random.shuffle(pass_components)
    return pass_components

main()
```