

# ISP\_AP\_performance\_task

```

import random
import string
import inquirer
import os

lowercase_letters = list(string.ascii_lowercase)
uppercase_letters = list(string.ascii_uppercase)
digits = list(string.digits)
special_characters = ['!', '#', '$', '%', '&', '(', ')', '*', '+', ',', '-', '=', '?', '@', '[', ']',
    '^', '_', '`', '{', '}', '~']

all_characters = {"Lowercase Letters": lowercase_letters, "Uppercase Letters": uppercase_letters, "Numbers": digits, "Special Characters": special_characters}
#print(all_characters)

def generate(password_length: int):
    questions = [
        inquirer.Checkbox('selection',
            message="Make your selection of what types of characters youd like in your password",
            choices=["Lowercase Letters", "Uppercase Letters", "Numbers", "Symbols"]
        ),
    ]
    answers = inquirer.prompt(questions)
    #(answers["interests"])
    user_selection = []
    char_list = []
    for i in answers["selection"]:
        for x in all_characters:
            if i == x:
                user_selection.extend(all_characters[x])
                #print(user_selection)

    for u in range(password_length):
        char_list.append(random.choice(user_selection))
    final_password = "".join(char_list)
    print("Your password is: " + final_password)
    input("Press enter to continue. . . ")
    os.system('cls')

def get_length():
    global length
    length = int(input("Input the desired length of your password: "))

if __name__ == "__main__":
    print(
        """
    _____
    | _ | _ _ _ _ _ _ _ _ _ _ _ _ _ _ | | | _ | _ _ _ _ _ _ _ _ _ _ | | _ _ _
    | _ | . | _ - | _ - | | | . | _ | . | | | | _ | _ - | | _ - | _ | . | _ | . | _
        """
    )

```

```
53 | _ | _ , _ | _ | _ | _ | _ | _ | _ | _ | _ | _ , _ | _ | _ |  
54 |  
55 | ""  
56 | )  
57 | while True:  
58 |     get_length()  
59 |     generate(length)  
60 |     if input('Generate another password?[y/n] ') != 'y':  
61 |         os.system('cls')  
        break
```

---

PDF document made with CodePrint using [Prism](https://bakerfranke.github.io/codePrint/)