# Placeholder functions to simulate the missing functions from v2\_Password\_Manager module  
def creating\_file():  
 pass  
  
def decrypting\_file():  
 pass  
  
SYMBOLIC = "!@#$%^&\*()\_+=-[]{}|\\;:'\",.<>/?"  
# End placeholder functions  
  
def running\_program():  
 print('\nWelcome to Prototype Password Manager.\nHere you can create passwords which will be encrypted in a file saved to your device.\nFurther functionality will be added for user-specific save files with another layer of password protection.\nEnjoy!')  
 while True:  
 mode = input('\n----------------------\n- Create Password[c] -\n----------------------\n- Open File[o] -\n----------------------\n Input:')  
 if mode in 'cC':  
 initial\_loop = True  
 while initial\_loop:  
 file\_name = input('\nName of file encrypted password will be saved to [No use of special characters]: ')  
 if any(char in SYMBOLIC for char in file\_name):  
 print('Invalid Character Detected')  
 else:  
 password = input(  
 '\nDo you want a specific password to be saved and encrypted to your device [type in your password if yes, type \'NO\' if not]: ')  
 if password.upper() == 'NO':  
 while True:  
 password\_amount = input('\nHow many passwords do you want generated? ')  
 if password\_amount.isdigit():  
 password\_amount = int(password\_amount)  
 break  
 else:  
 print('Not a number')  
 initial\_loop = False  
 else:  
 initial\_loop = False  
  
 elif mode in 'oO':  
 # Code for opening files  
 pass  
 else:  
 print('\nInvalid input. Try Again.')  
 continue  
  
running\_program()