Checking for strength of a password

Due date: 22 July, 2020.

After completing the assignment email your code to me at imranali @um.uob.edu.pk.

Strong Password allow users to prevent attackers from guessing passwords or brute-force attacks. See this wiki article on password strength for details.

Your assignment is to complete the following program, by completing the function **strongpwd** which returns 1 if the password array passed to it contains a strong password, and returns 0 otherwise.

```
#include<stdio.h>
#include<ctype.h>

int length(char array[]);
int strongpwd(char array[]);

int main()
{

    char password[]="123456";
  char password2[]="qwerty";
  char password3[]="qwerty123";
    char password4[]="14July_2020!";

  printf("\n1. Length of password:\t %s is %d", password, length(password));
  printf("\n2. Length of password2:\t %s is %d", password2, length(password2));
  printf("\n3. Length of password3:\t %s is %d", password3, length(password3));
  printf("\n4. Length of password4:\t %s is %d", password4, length(password4));


  printf("\n1. Strength of password:\t %s is %d", password, strongpwd(password));
  printf("\n2. Strength of password2:\t %s is %d", password2, strongpwd(password2));
  printf("\n3. Strength of password3:\t %s is %d", password3, strongpwd(password3));
  printf("\n4. Strength of password4:\t %s is %d", password4, strongpwd(password4));



    printf("\n");
    return 0;
}

int length(char array[])
{
```

```
34      int length=0;
35      int count=0;
36
37      while(array[count] != '\0')
38      {
39          length++;
40          count++;
41      }
42
43      return length;
44  }
45
46
47  int strongpwd(char array[])
48  {
49    int strong=0;
50
51      /* complete the program by
52       * writing this function
53       * you may use the character handling
54       * functions taught in the class
55       */
56
57    return strong;
58  }
```

To save time download the above program from github.

Following are the rules for a valid strong password:

1. It should be at least 8 characters in length, and
2. It should contain a combination of alphabets [a-z], numerics[0-9], and a special symbol e.g. (_, @, !)

You may take help from the function length which I have already provided.