Strong Password Checker

A password is strong if all are met:

- Between 6 and 20 characters.
- At least 1 lowercase, 1 uppercase and 1 digit
- It must not contain 3 repeating characters in a row
  - aaa WRONG
  - aa...a CORRECT

Write strongPswCheck(String psw) to return minimal change to make the psw strong. If psw is already strong, return 0. Insert, deletion and replacement of any character is considered as a change.

Breakdown

First, we need to check length.

Possible cases:
- $n < 6$ — Reject, too long or too short.
- $n > 20$
- $6 < n < 20$ — OK

So, we have only one case where we need to inspect further $6 < n < 20$. We need to declare an array of integers that will store the id of the conditions the input failed.

Possible fail conditions

length < 6

length > 20

Psw contains at least 1 uppercase

Psw contains at least 1 lowercase

Psw contains at least 1 digit

psw does not contain three repeating characters in a row.

Algorithm 1

```
int strongPswCheck(char *psw){
    int len = strlen(psw)
    if(len < 6){printf("length is less than 6, add more characters");}
    if(len >20){printf("length is more than 20, reduce psw length");}
    if(checkUppercase ==0){printf("No uppercase char detected");}
    if(checkLowercase ==0){printf("No lowercase char detected");}
    if(checkLowercase ==0){printf("No lowercase char detected");}
}

int checkUppercase(char *psw){
    char *check = psw
    int len = strlen(psw);
    int success = 0;   //Will change upon uppercase discovery
    for(int i=0; i<=len; i++){
        if(check[i].isUpper()==1){
            success = 1;
```

```
checkUppercase(psw);
checkLowercase(psw);
checkDigit(psw);
```

```
    break;
}}
```