2. A password is said to be strong if it satisfies the following criteria:

It contains at least one lowercase English character.

It contains at least one uppercase English character.

It contains at least one special character.

The special characters are: !@#$%^&*()-+

Its length is at least 8.

It contains at least one digit. Given a string, find its strength.

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
// Function to check if a character is a special character
bool isSpecialChar(char ch) {
 char specialChars[] = "!@#$%^&*()-+";
 for (int i = 0; i < strlen(specialChars); i++) {
 if (ch == specialChars[i]) {
 return true;
 }
 }
 return false;
}
// Function to check password strength
void checkPasswordStrength(char password[]) {
 int length = strlen(password);
 bool hasLower = false, hasUpper = false, hasSpecial = false, hasDigit = false;
 for (int i = 0; i < length; i++) {
 if (islower(password[i])) {
 hasLower = true;
 } else if (isupper(password[i])) {
 hasUpper = true;
 } else if (isSpecialChar(password[i])) {
 hasSpecial = true;
```

```
 } else if (isdigit(password[i])) {
 hasDigit = true;
 }
 }
 if (length >= 8 && hasLower && hasUpper && hasSpecial && hasDigit) {
 printf("Password is strong.\n");
 } else {
 printf("Password is not strong.\n");
 }
}
int main() {
 char password[50];
 // Input password from the user
 printf("Enter the password: ");
 scanf("%s", password);
 // Check password strength
 checkPasswordStrength(password);
 return 0;
}
```

Output:

Enter the password: 5263

Password is not strong