Juan Miguel Consebido  
BSCS 1-1  
  
Vowel Count  
A close-up of a computer screen

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.  
  
#include <iostream>

#include <string>

#include <cctype>

int main() {

std::string text;

int vowelCount = 0, consonantCount = 0;

int i = 0;

std::cout << "Enter text: ";

std::getline(std::cin, text);

while (i < text.length()) {

char ch = std::tolower(text[i]);

if (std::isalpha(ch)) {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowelCount++;

} else {

consonantCount++;

}

}

i++;

}

std::cout << "Vowel count: " << vowelCount << std::endl;

std::cout << "Consonant count: " << consonantCount << std::endl;

return 0;

}  
  
  
  
  
  
Password And Username  
A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.  
  
#include <iostream>

#include <string>

#include <cctype>

int main() {

std::string username;

std::string password;

std::string temppass;

int attempts = 3;

while (attempts > 0) {

temppass.clear();

// Prompt the user to enter a username and password

std::cout << "Enter your username: ";

std::getline(std::cin, username);

std::cout << "Enter your password: ";

std::getline(std::cin, password);

for (char c : password) {

if (c != ' ') {

temppass += c;

}

}

password = temppass;

// Perform validations on the username and password

bool containsSpace = (username.find(' ') != std::string::npos);

bool isPasswordLongEnough = (password.length() >= 8);

bool contains123 = (password.find("123") != std::string::npos);

bool isPasswordDifferent = (password != username);

bool hasSpecialChar = false;

for (unsigned char c : password) {

if (!std::isalnum(c)) {

hasSpecialChar = true;

break;

}

}

// Validate all conditions

if (!containsSpace && isPasswordLongEnough && contains123 && isPasswordDifferent && hasSpecialChar) {

std::cout << "Access Granted!" << std::endl;

return 0;

} else {

// Specific error messages

if (containsSpace) {

std::cout << "Error: Username should not contain spaces." << std::endl;

}

if (!isPasswordLongEnough) {

std::cout << "Error: Password should be at least eight characters long." << std::endl;

}

if (!contains123) {

std::cout << "Error: Password must contain the string '123'." << std::endl;

}

if (!isPasswordDifferent) {

std::cout << "Error: Password should not be the same as the username." << std::endl;

}

if (!hasSpecialChar) {

std::cout << "Error: Password must contain a special character." << std::endl;

}

}

attempts--;

if (attempts > 0) {

std::cout << "You have " << attempts << " attempt(s) left. Please try again." << std::endl;

} else {

std::cout << "Access Denied. No attempts left." << std::endl;

}

}

return 0;

}