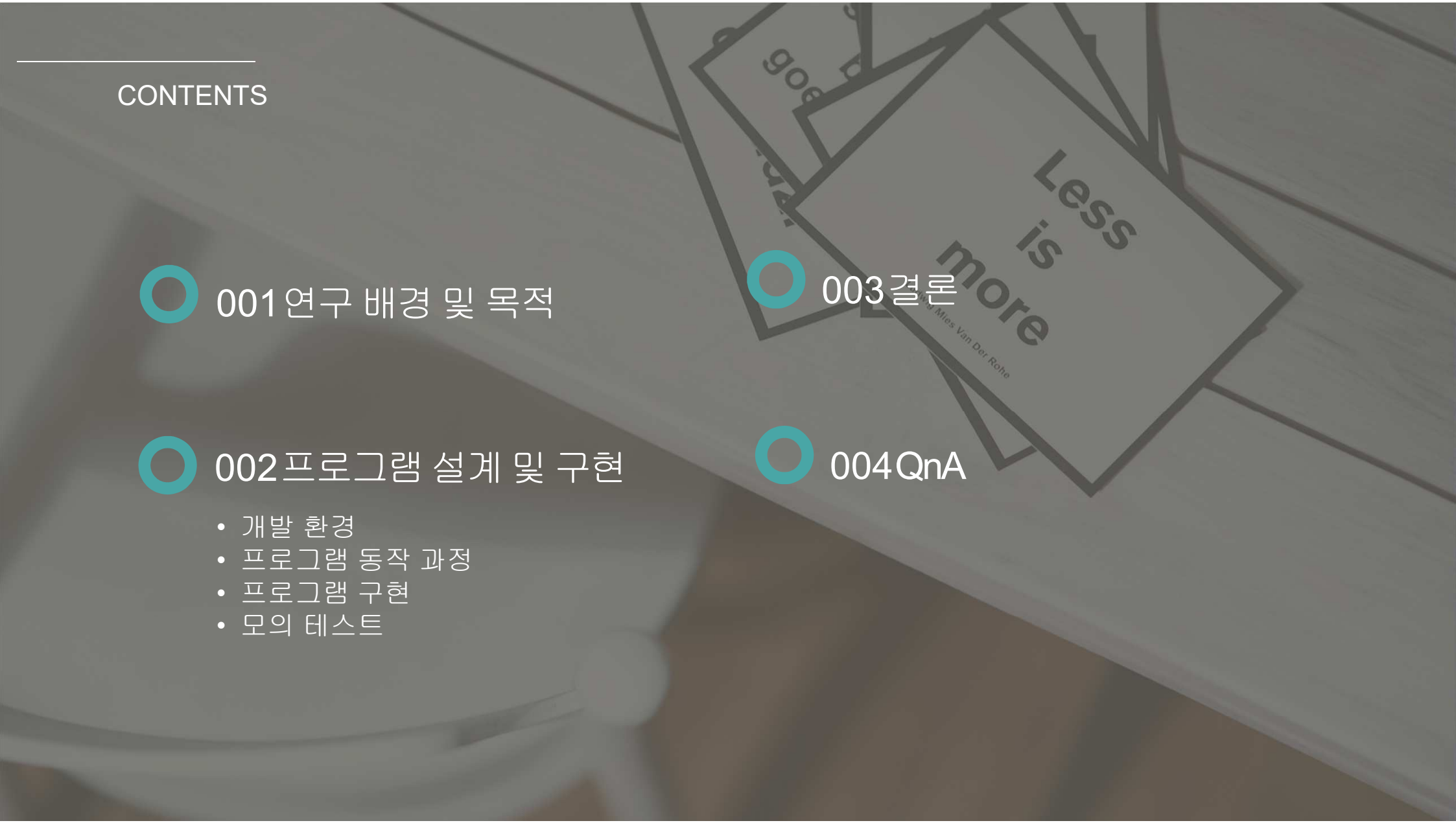
난수를 이용한
암호 생성 프로그램 구축

지도교수 : 신승수
발표자 : 손 * 호
팀명 : CDT

CONTENTS



1.1 연구 배경 및 목적

- 현재 인터넷 통신의 발전과 함께 사용자들의 스마트폰 또는 개인 컴퓨터를 통한 다양한 온라인 서비스 이용이 가속화 되고 있다.
- 불법적 사용을 통제하기 위하여 사용자를 확인하는 사용자 인증 기술이 계속 발달하고 있고 중요성 또한 증가되고 있는 추세이다.
- 온라인 서비스에서 사용되는 가장 일반적인 사용자 인증 기술은 비밀번호 인증 방식이다.
- 하지만 대부분의 사용자들은 비밀번호의 설정 행태가 심하게 좋지 않다.

1.2 연구 배경 및 목적

사람들이 가장 많이 사용한 비밀번호

1.3 연구 배경 및 목적

인터넷 사용자가 피해야 할 비밀번호

1.4 연구 배경 및 목적

안전한 비밀번호 작성 규칙

최소 길이

최소 10자리 이상 : 영어 대·소문자, 숫자, 특수문자 중 2종류 조합

최소 8자리 이상 : 영어 대·소문자, 숫자, 특수문자 중 3종류 조합

주기적 변경

비밀번호에 유효기간 결정하고 최소 6개월마다 변경

동일 비밀번호 사용 제한

2개의 비밀번호를 교대로 사용하지 않음

1.5 연구 배경 및 목적

비밀번호 생성 조건

비밀번호 길이

영어 대·소문자, 특수문자, 숫자로 구현하며 16자리로 생성

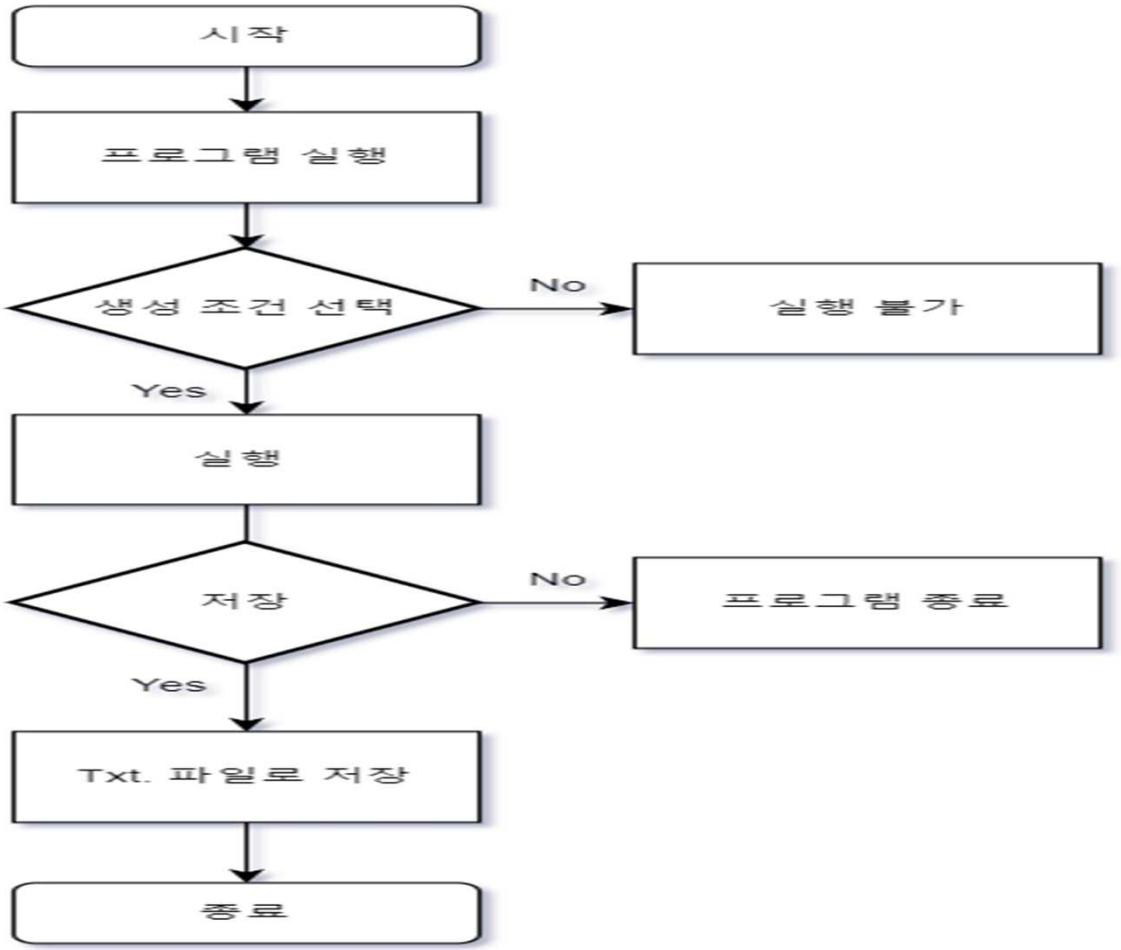
비밀번호 생성시 선택 조건

1. 구별하기 힘든 문자 배제
2. 원하는 문자열

2.1 개발환경

사용 프로그램	운영체제	하드웨어	추가 요구 사항
Visual Studio 2019	Windows 10 버전 1703 이상: Home, Professional, Education 및 Enterprise(LTSC 및 S는 지원되지 않음)	1.8GHz 이상의 프로세서 쿼드 코어 이상 추천 2GB RAM(8GB RAM 추천) 하드 디스크 공간: 설치된 기능에 따라 최소 800MB, 최대 210GB의 사용 가능한 공간이 필요하며, 일반적인 설치에는 20~50GB의 사용 가능한 공간이 필요하다. 최소 디스플레이 해상도 720p(1280x720)를 지원하는 비디오 카드.	C++, JavaScript 또는 .NET 워크로드를 사용하여 모바일 개발을 설치하려면 Windows 7 SP1에 PowerShell 3.0 이상이 필요하다.
IDLE(Python 3.10 64bit)	Windows 7 이상, Mac, Linux 운영체제 지원	1 GHz 이상의 프로세서. 인텔 i5 이상 최소 4기가바이트(GB) 메모리(RAM), 8기가 이상 256GB 하드 디스크 공간	x

2.2 프로그램 동작 과정



2.3 프로그램 구현

헤더파일 코드

```C++
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <qfile.h>
#include <string>
#include <regex>
#include <QFileDialog>
#include <QMessageBox>
#include <time.h>
```

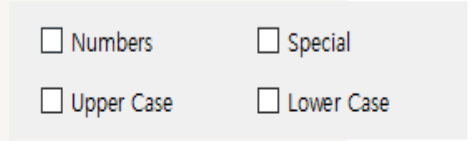
비밀번호 개수 설정 코드

```C++
MainWindow::MainWindow(QWidget *parent) // 메인 윈도우 호출
: QMainWindow(parent)
, ui(new Ui::MainWindow)
{
ui->setupUi(this);
ui->CountBox->setMinimum(1); // 비밀번호 만드는 갯수 최솟값
ui->CountBox->setMaximum(99999999); // 비밀번호 만드는 갯수 최댓값
ui->LengthBox->setMinimum(1); //비밀번호 길이 최솟값
ui->LengthBox->setMaximum(99999999); // 비밀번호 길이 최댓값
}
```

2.4 프로그램 구현

```
srand(time(NULL)+ran*ran2); // 시드를 시간값 + 랜덤값1 * 랜덤값2 으로 설정
for(int i = 0; i<ui->CountBox->value(); ++i) // 비밀번호 만드는 개수만큼 반복
{
    QString value2 = ui->AddText->toPlainText(); //비밀번호 추가값
std::string charSet(""), // 비밀번호 글자 모음 (숫자,특문,대문자,소문자)
            output(""); // 비밀번호
```

seed값 초기화 과정



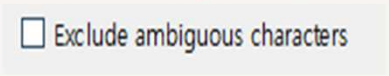

```
    QString value2 = ui->AddText->toPlainText(); //비밀번호 추가값
std::string charSet(""), // 비밀번호 글자 모음 (숫자,특문,대문자,소문자)
            output(""); // 비밀번호

            if(ui->checkNum->isChecked()) // 숫자추가 체크 확인
            {
                charSet+="0123456789"; // 비밀번호 글자 모음에 숫자를 추가
                        }
            if(ui->checkSpe->isChecked()) // 특수문자추가 체크 확인
            {
                charSet+="!@#$%^&*"; // 비밀번호 글자 모음에 특수문자를 추가
            };
            if(ui->checkLow->isChecked()) // 소문자추가 체크 확인
            {
                charSet+="abcdefghijklmnopqrstuvwxyz"; // 비밀번호 글자 모음에 소문자를 추
가
            };
            if(ui->checkUp->isChecked()) // 대문자추가 체크 확인
            {
                charSet+="ABCDEFGHIJKLMNOPQRSTUVWXYZ"; // 비밀번호 글자 모음에 대문자를 추
가
            };
```

비밀번호 선택지 생성 코드

2.5 프로그램 구현

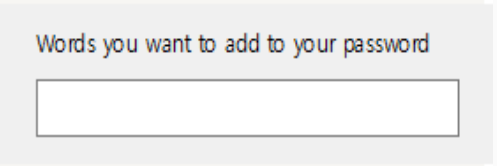
```
};
if(ui->checkExc->isChecked()) // 비슷한 문자 제거 체크 확인
{
    charSet = std::regex_replace(charSet, std::regex("i"), "");
    charSet = std::regex_replace(charSet, std::regex("l"), "");
    charSet = std::regex_replace(charSet, std::regex("1"), "");
    charSet = std::regex_replace(charSet, std::regex("L"), "");
    charSet = std::regex_replace(charSet, std::regex("o"), "");
    charSet = std::regex_replace(charSet, std::regex("0"), "");
    charSet = std::regex_replace(charSet, std::regex("O"), "");
}
```



비슷한 문자 제거 코드

```
    if(value2.length() != 0) //추가 비밀번호값이 있으면 실행
    {
        output+=value2.toStdString(); //추가 비밀번호값 먼저 추가
        for(int i = value2.length(); i<ui->LengthBox->value(); ++i) //비밀번
호 길이만큼 반복
        {
            output+=charSet.at(rand()%(int)charSet.length()); // 비밀번호 글
자 추가
        }
    }
    else // 추가 비밀번호값이 없으면 실행
    {
        for(int i = 0; i<ui->LengthBox->value(); ++i) // 비밀번호 길이만큼 반
복
        {
            output+=charSet.at(rand()%(int)charSet.length()); //비밀번호 글자
추가
        }
    }
        ui->plainTextEdit->appendPlainText(output.c_str()); // 비밀번호 출력

}
```

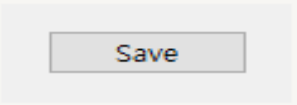


원하는 문자열 입력 코드

2.6 프로그램 구현 저장코드

```
void MainWindow::on_SaveButton_clicked() // 저장버튼이 눌렸을때
{
QString txtFile = QFileDialog::getSaveFileName(this, tr("Save file"), "",
tr("text file (.txt);;C++ File (.cpp *.h)")); // 파일경로 선택, 파일이름 정하기
if (txtFile != "") // 경로가 선택 되었을 경우
{
QFile file(txtFile);
```

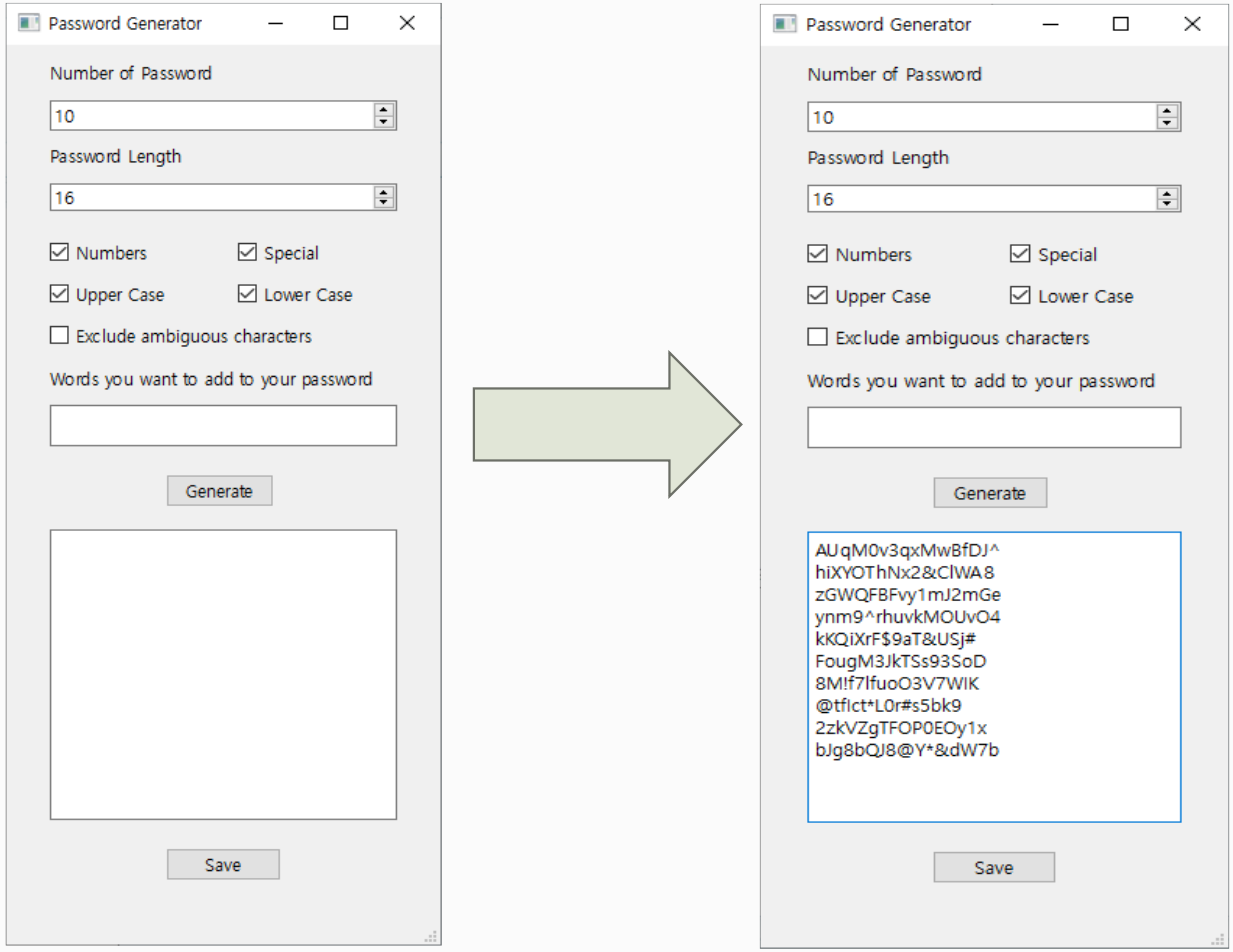
```
        if (file.open(QIODevice::ReadWrite)) // 텍스트 파일 열기 성공시
        {
        QTextStream stream(&file);
            stream << ui->plainTextEdit->toPlainText(); // 나온 비밀번호들을 복사하기
            file.flush(); // 텍스트 파일에 붙여넣기(쓰기)
            file.close(); // 텍스트 파일 닫기
        }
        else // 텍스트 파일 열기 실패시
        {
            QMessageBox::critical(this, tr("Error"), tr("Error")); // 실패시 에러메세지
            return;
        }
    }
```



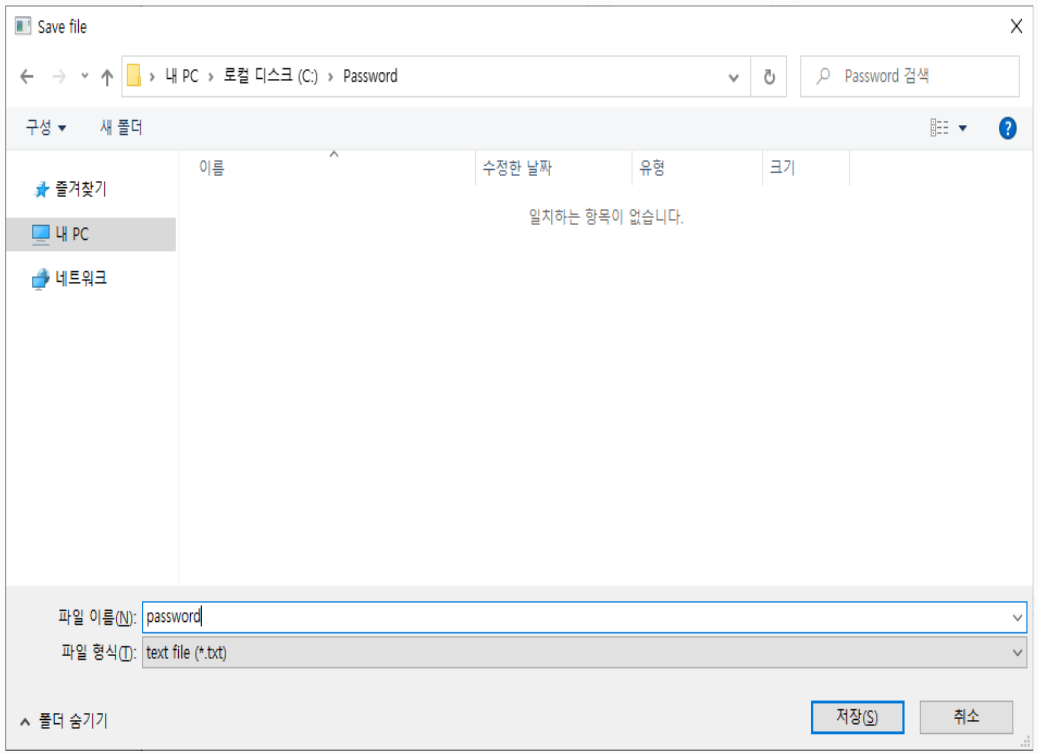
2.7 언어 비교

	Python	c++
속도	느림	빠름
메모리 효율	낮음	높음
용도	웹 어플리케이션 개발	순수 응용프로그램 개발
오류 발생	높음	낮음

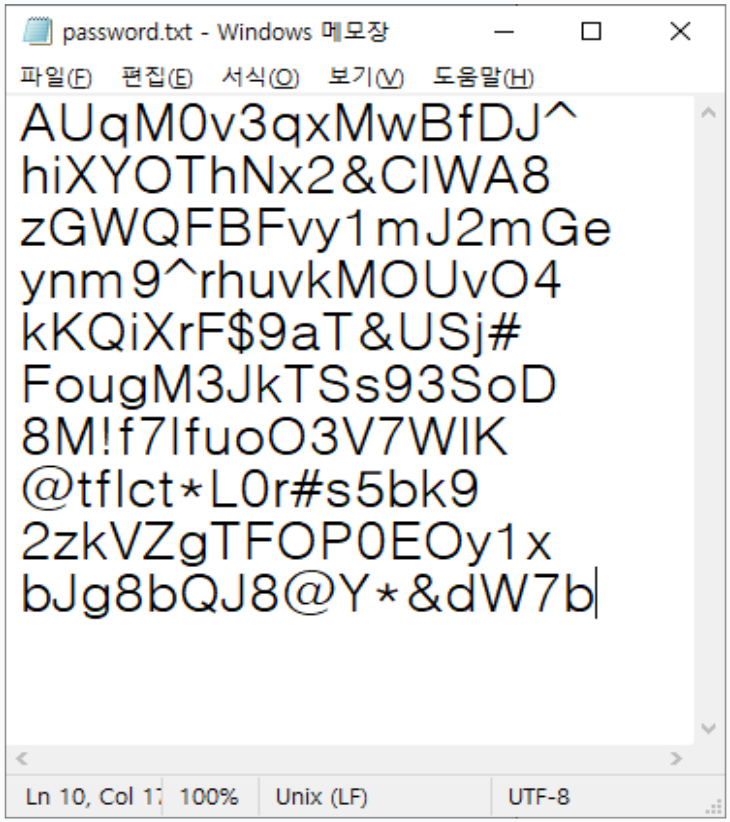
2.8 모의 테스트



2.9 모의 테스트



저장 과정



저장된 .txt파일 확인

3.1 결론

- Python의 GIL(Global interpreter Lock) 정책으로 인하여 Python은 프로그램 구축에 적합한 언어가 아니라고 판단하여 중지함.
- 암호 생성 프로그램으로 인해 회원가입을 하거나 비밀번호를 바꿀 때 추천 비밀번호로서 출력이 된다면 안전한 비밀번호를 생성할 수 있을 것이다.
- 언어의 범용성을 고려하지 않고 비밀번호를 생성할 수 있다면 다른 나라의 언어나 단어들을 조합하여 많은 경우의 수를 만들어 더욱 안전한 비밀번호를 생성할 수 있을 것이다.

QnA