C++ language report

Deadline 2019

[201601639 컴퓨터공학부 홍승현] e-mail : cx\_nf@naver.com Contact : 010 – 4019 - 2514

목차

[클래스 다이어 그램 1](#_Toc24722286)

[Character.h 2](#_Toc24722287)

[protected 2](#_Toc24722288)

[int hp 2](#_Toc24722289)

[int mp 2](#_Toc24722290)

[int armor 2](#_Toc24722291)

[int damage 2](#_Toc24722292)

[솔루션 내 코드 2](#_Toc24722293)

[PUBLIC 2](#_Toc24722294)

[Character(const int hp, const int mp, const int armor, const int damage) : hp(hp), mp(mp), armor(armor), damage(damage) 2](#_Toc24722295)

[Virtual ~Character() 2](#_Toc24722296)

[bool HP\_isEmpty() const 2](#_Toc24722297)

[bool MP\_isUsable() const 3](#_Toc24722298)

[virtual void Attack(Character\* c) = 0; 3](#_Toc24722299)

[virtual void Skill(Character\* c) = 0; 3](#_Toc24722300)

[Get set 함수 3](#_Toc24722301)

[솔루션 내 코드 3](#_Toc24722302)

[정리 4](#_Toc24722303)

[Character 상속받는 Thives.h, Wizards.h, Warriors.h, Enemies.h 5](#_Toc24722304)

[public 5](#_Toc24722305)

[Thieves() : Character(120, 60, 7, 16) 또는 Warriors() : Character(150, 60, 10, 13) 또는 Wizards() : Character(120 80 7 10) 5](#_Toc24722306)

[~Thieves() 또는 ~Warriors() 또는 ~Wizards() 5](#_Toc24722307)

[void Attack(Character\* enemies) 또는 void Attack(Character\* main\_hero) 5](#_Toc24722308)

[void Skill(Character\* enemies) 또는 void Skill(Character\* trash) 5](#_Toc24722309)

[thieves.h 코드 6](#_Toc24722310)

[warriors.h 코드 7](#_Toc24722311)

[wizards.h코드 8](#_Toc24722312)

[Enemies.h 코드 9](#_Toc24722313)

[정리 9](#_Toc24722314)

[GameMain.h 10](#_Toc24722315)

[private 10](#_Toc24722316)

[int difficulty 10](#_Toc24722317)

[Character\* enemies 10](#_Toc24722318)

[Character\* main\_hero 10](#_Toc24722319)

[int Set\_Difficulty() 10](#_Toc24722320)

[int Set\_Job() 10](#_Toc24722321)

[int Set\_Menu() 10](#_Toc24722322)

[public 10](#_Toc24722323)

[GameMain() 10](#_Toc24722324)

[~GameMain() 11](#_Toc24722325)

[bool Error(const string& errorcode, const int size) 11](#_Toc24722326)

[void Menu() 11](#_Toc24722327)

[void Start() 11](#_Toc24722328)

[솔루션 내 코드 11](#_Toc24722329)

[정리 12](#_Toc24722330)

[GameMain.cpp 알고리즘, 코드 설명 13](#_Toc24722331)

[gamemain\_cpp 코드 13](#_Toc24722332)

[정리 14](#_Toc24722333)

[프로그램 간략도 15](#_Toc24722334)

[GameMain 코드 간략화 15](#_Toc24722335)

[게임시작(Start() 메소드) 코드 설명 및 간략화 16](#_Toc24722336)

[입력(Set\_Difficulty, Set\_Job, Set\_Menu) 코드 설명 및 간략화 16](#_Toc24722337)

[올바른 입력인지 판별하는 오류판별(Error 메소드)코드 설명 및 간략화 17](#_Toc24722338)

[출력결과 18](#_Toc24722339)

[이 과제를 함으로써 느꼈던 점 20](#_Toc24722340)

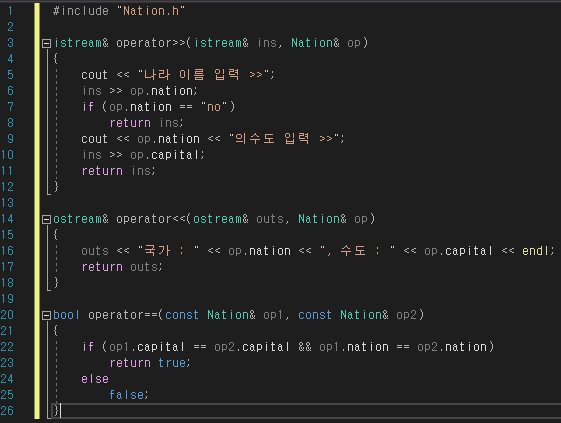
[성장했다? 20](#_Toc24722341)

# Nation.h

## 솔루션 내 코드



### 정리

nation과 capital을 멤버로 가지는 클래스이며, get, set 함수와 함께 연산자 중복정의 3개를 추가하였다. 이는 다음 사진과 같다.

연산자 중복정의는 해당 멤버 변수를 입력 , 출력, 그리고 동등한지의 여부를 참, 거짓으로 반환하게끔 만들었으며 이는 Manager 클래스에서 사용하게 될 것임을 미리 알린다.  
또한 istream 연산자 내부에 조건문으로 no를 입력받았을 경우 수도의 입력을 받지 않고 바로 반환하는 모습을 볼 수 있다. 이 역시 Manager에서 사용하게 될 예정이다.

# Manager.h

## private

### vector<Nation> v

Nation 클래스형 벡터이다. 여러 Nation 객체들을 통해 많은 입력과 수정, 제거를 할 수 있다.

### void Upload()

현 저장 폴더에 있는 Nation.txt의 값을 벡터에다 저장하게 해주는 함수이다.

### void Rename\_Capital()

수도의 이름을 수정하게 해주는 함수이다.

### void Delete\_Nation()

국가(데이터)를 삭제해주는 함수이다.

### void BackUp()

벡터 내에 있는 정보들을 그대로 txt 파일에 업로드한다.

### void Individual\_Search()

나라 이름을 입력받아 해당 나라의 수도를 출력해주는 함수이다.

### void All\_Print()

입력과 상관없이 현재 벡터에 있는 모든 국가의 이름과 수도를 출력해준다.

### void Quiz()

해당 벡터의 데이터를 가지고 수도를 맞추는 게임을 진행한다.

### vector<Nation>::iterator Find\_Overlap(const Nation& nation)

Nation 객체가 벡터 내의 값이랑 중복되는지(같은지) 확인 후 해당 위치를 반환함.  
중복되지 않는다면 벡터의 끝을 가리키는 v.end()를 리턴함.

### vector<Nation>::iterator Find\_Overlap(const string& nation\_name)

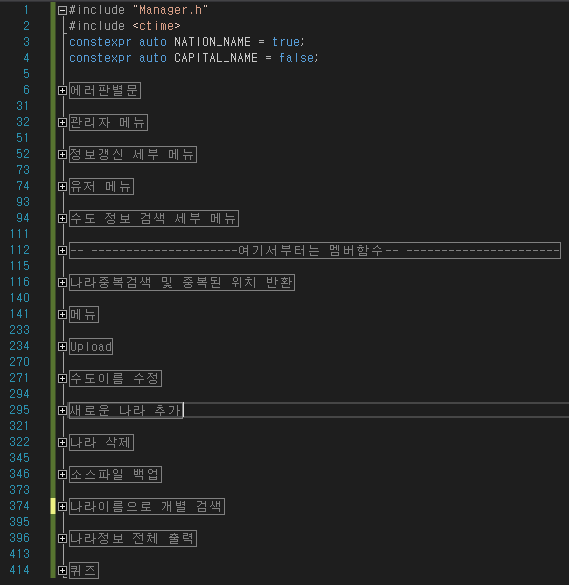
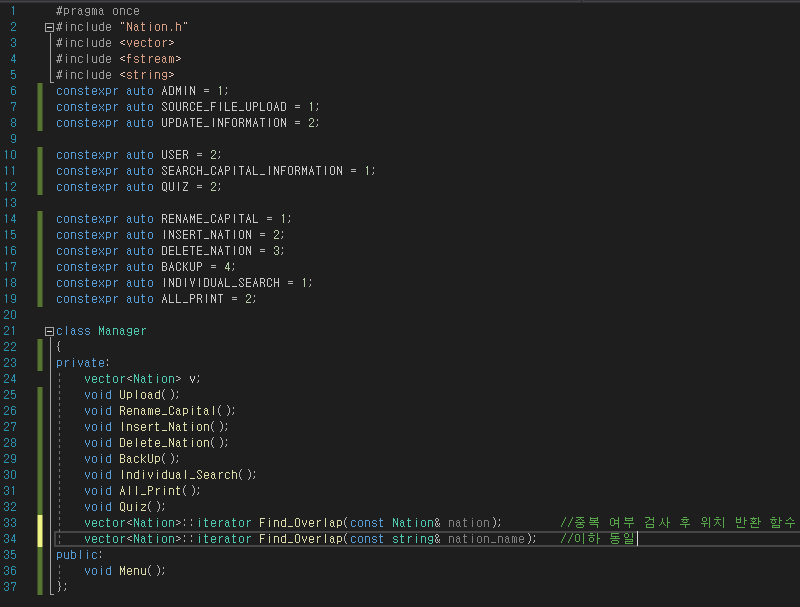
위와 동일하나, Nation 객체가 아닌 국가의 이름으로 중복검색하는 것과 전자에 비해 차이가 있다.

## public

### void Menu()

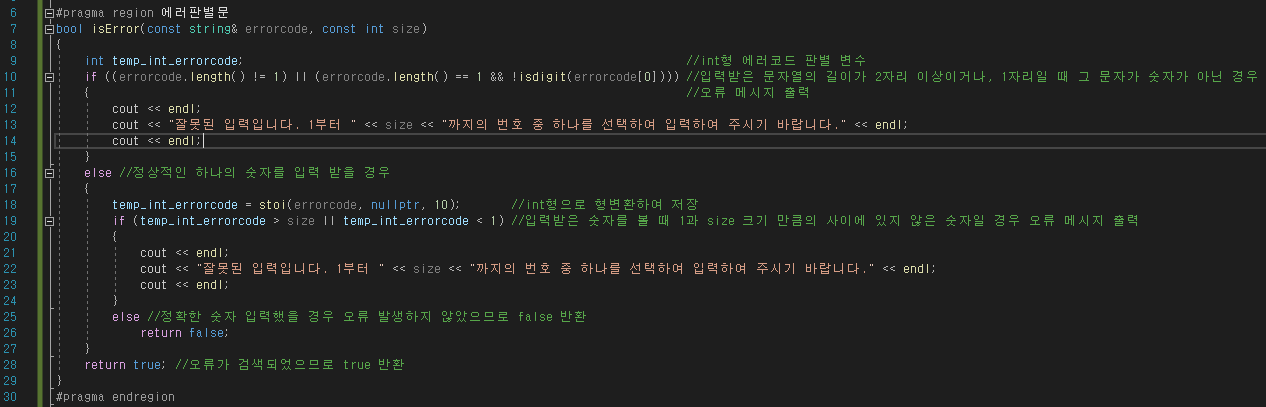
메뉴를 출력해주고 메뉴 번호에 따라 맞는 함수를 실행시키게 하는 전반적인 부분을 맡는 함수이다.

## 솔루션 내 코드



# Manager.cpp

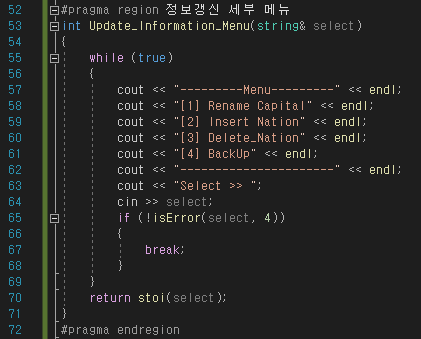
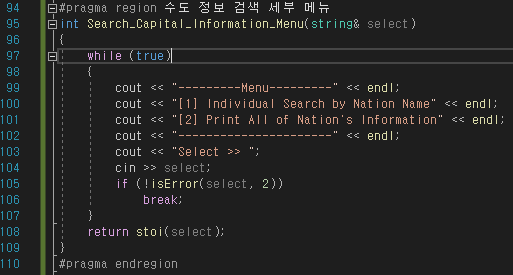
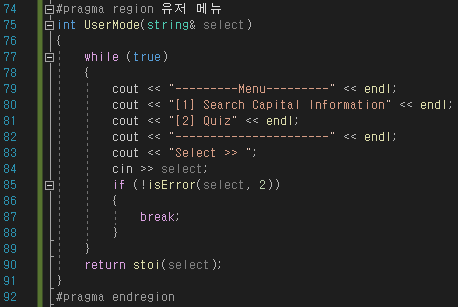
## 에러판별문



### 정리

저번 과제 때 사용했던 것을 그대로 가져왔다. 파라미터로 입력받은 문자열 값과 size까지 입력받기 위한 int값도 받아서 그 사이가 아닐 경우 무조건 적으로 에러를 출력해준다.

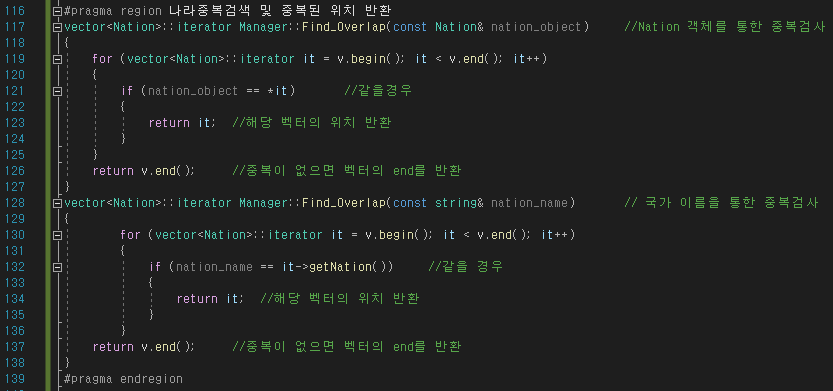
## 메뉴



### 정리

서로의 메뉴에 대해 출력하여 메뉴 선택한 값을 반환해주는 함수들이다. 이는 전반적으로 똑같다고 보면 된다.

## 나라중복 검색 및 위치 반환

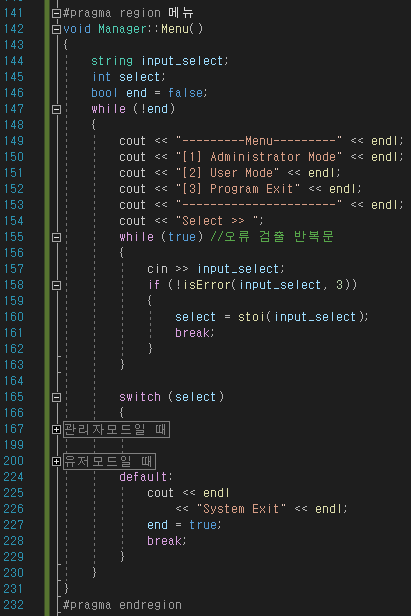


### 정리

해당 멤버인 벡터의 iterator을 반환하는 것으로써, 벡터의 처음부터 끝까지 하나씩 증가해가면서 파라미터로 받아온 참조형 객체나 문자열을 비교하여 같을경우에는 그 위치(iterator) 값을 반환하고, 없을경우에는 벡터의 끝 다음을 가리키는 v.end()를 반환한다.

여기서 121줄의 ==은 연산자 중복정의를 하여 구분하는 것으로 보고서 2page에 기술하였다.

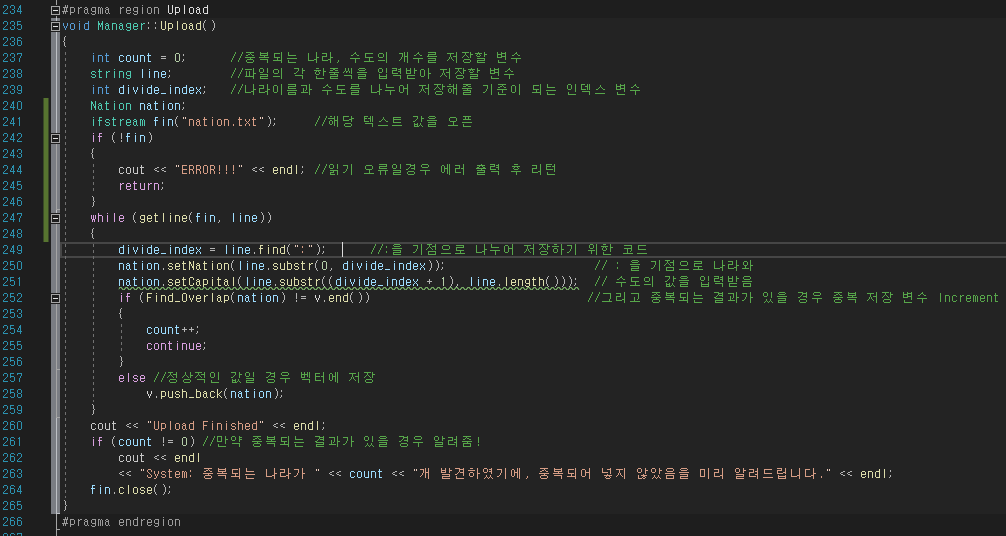
## 메뉴



### 정리

첫 메뉴 화면에는 관리자모드와 유저모드 그리고 종료가 있다. 관리자 모드일 때는 관리자에 대한 세부메뉴가 출력 및 입력되고, 유저모드일 경우에는 관리자모드일 때를 제외한 세부메뉴들을 선택하게 되어있다. 이러한 모드와 세부메뉴들은 위쪽에 기술해두었다.

## 업로드

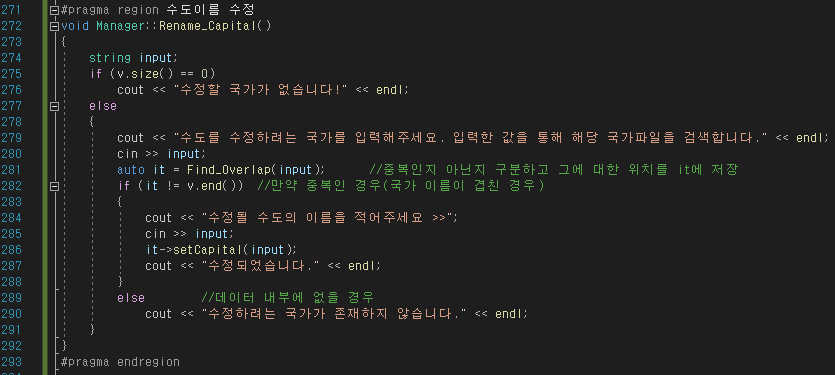


### 정리

txt파일을 읽어 Vector 변수에 저장하는 함수로, fin을 사용하여 문자열이 끝날 때 까지 읽어오고 문자열의 함수인 find를 통해 ‘:’을 기점을 왼쪽은 국가 이름, 오른쪽은 국가의 수도로 나누어 저장하는 것을 볼 수 있다.

또한, 또 한 번 업로드를 한다던가, 이미 데이터를 추가한 상태에서 업로드를 하게 될 경우 중복이 발생될 수 있다. 그럴 때 조건문으로 나누어 중복일 경우 count의 값을 1 증가시키고 값을 저장하지 않는식으로 하여 효율적으로 관리해주고, 후에 중복되는 수를 출력해준다.

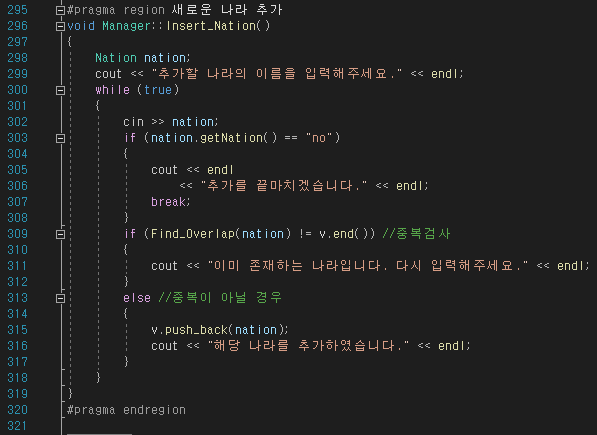
## 수도이름 수정



### 정리

수도를 수정할 때, 데이터가 들어있는지 확인을 하고, 하나의 string형 input 변수를 두 번 사용하여 국가와 수도를 입력받는다. 처음으로 국가를 입력받을 때는 수도를 수정하려는 목적으로 국가를 입력받고, 중복검사 후, 수도를 수정하기위해 수도를 입력받는다.

## 나라 추가



### 정리

Nation 객체를 선언하여 연산자 중복정의에 의해 cin을 통해 객체로부터 멤버변수를 모두 입력받아 이를 중복검사 후 새로운 데이터면 저장을 한다. 위 if문을 통해 no라는 문자열을 입력 받았을 경우 추가를 종료하는 대목임을 알 수 있다.

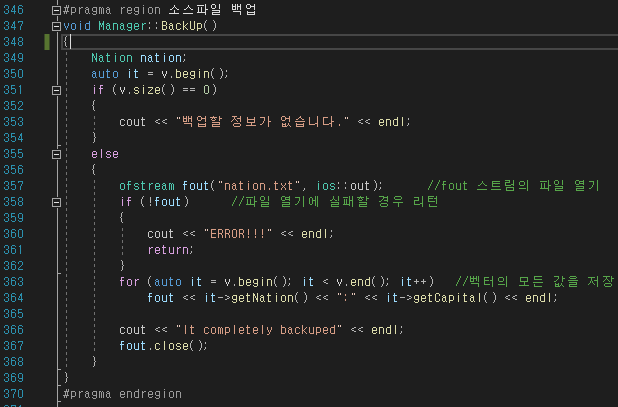
## 나라 삭제



### 정리

해당 국가의 이름을 입력받아 중복검사 후 해당 국가의 이름이 존재할 경우 그 위치를 반환받았으므로 erase 함수를 통해 벡터의 해당 부분을 제거한다.

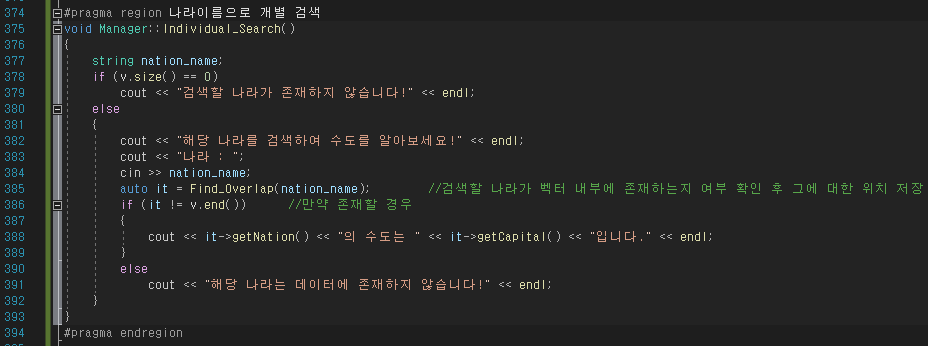
## 백업



### 정리

업로드의 반대로, 벡터의 데이터들을 txt파일에 덮어 씌워주는 함수이다. fout 스트림으로 파일을 열어 벡터가 가지고 있는 모든 값을 전부 열어서 변경하고자 하는 nation.txt 파일에 덮어씌운다.

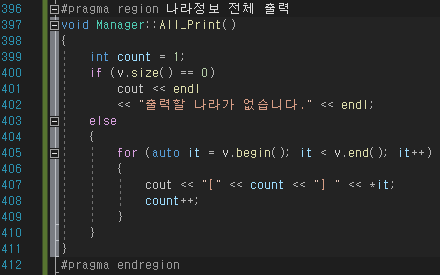
## 개별 검색



### 정리

하나의 국가 이름을 입력 받아 그 국가에 대한 수도 이름을 출력해주는 함수이다.

## 전체 출력



### 정리

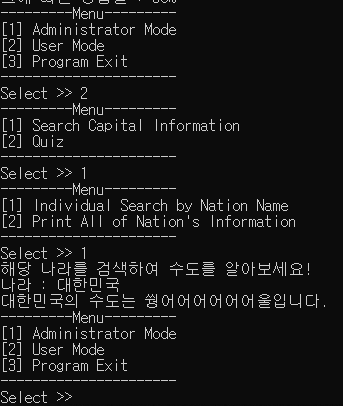
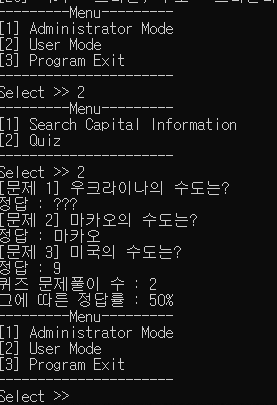
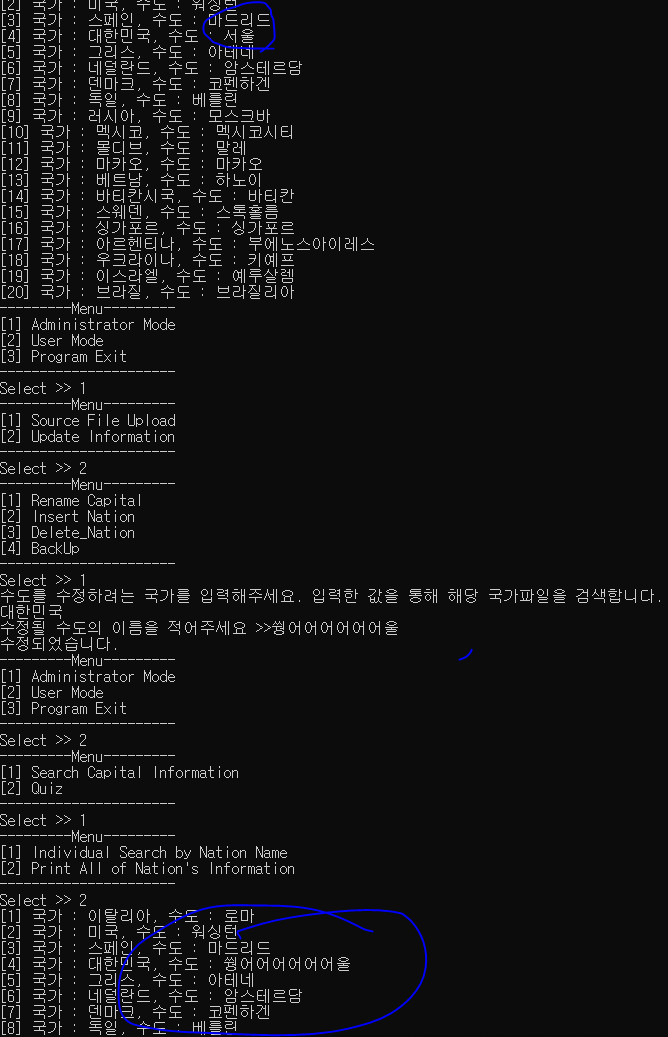
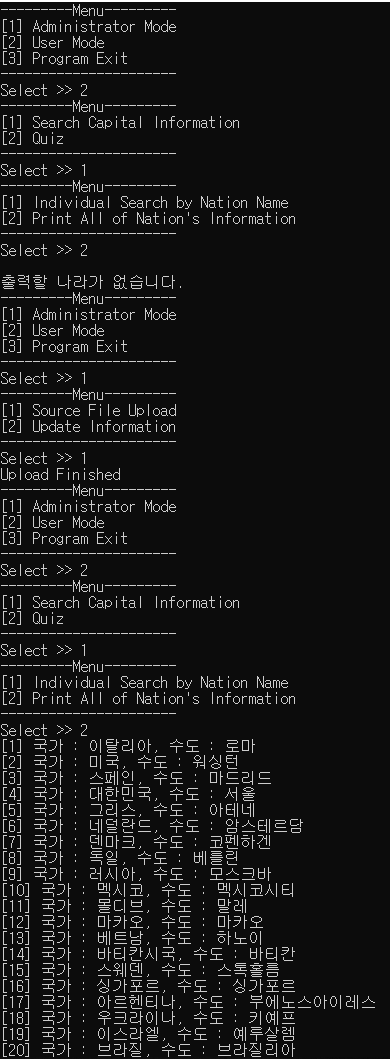
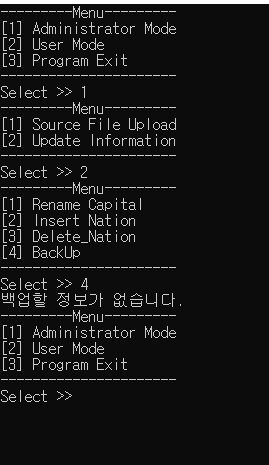
말 그대로 모든 벡터에 있는 데이터 값을 출력해주는 함수이다. 연산자 중복정의를 통해 cout << \*it가 되는 것을 확인할 수 있다.

## 퀴즈



해당 데이터의 수 사이에서 난수로 입력받아 국가의 이름을 출력하고 수도를 맞추어 정답률을 출력해주는 함수로 9라는 입력을 받을 경우 퀴즈 풀이를 마치고 그 전까지 했던 퀴즈에 대한 정보들을 결산하여 정보를 출력해준다.

## 출력결과



## 이 과제를 함으로써 느꼈던 점

### 과제중 가장 어려웠던 느낌? 이랄까?

지금 껏 했던 과제중에서 가장 어려운 축에 속한 느낌이였다. 다른 과제들은 수월하게 했었다만, 이번 마지막 과제는 파일입출력에 대해 완벽하게 숙지 못한 상태에서 한게 크게 작용해서인지 ppt자료를 들춰보면서 하기도 했다. 타 수업에서도 여러 과제들이 많기에 압박감에 의해 어렵다고 느껴진 것일수도 있겠다만, 확실한건 어렵게 느껴진 것이다.  
마지막 과제라고도 생각해서 그런 것일지도 모르겠다는 생각도 든다. 이렇게 마지막 과제까지 마무리를 지으며 종강을 향해 달려나가는 시간을 보니 벌써? 라는 생각이 든다. 참 시간이란 빠른 것 같다..!!!

프로젝트 보고서 종료

작성자 : 201601639 홍승현