

# 객체지향 프로그래밍 1 차 과제

2018202046 이준휘

## 1. Assignment 1

### - 문제 설명

1 차원 배열을 사용하여 20 미만의 랜덤 수를 저장하고, int \*row[4]에 각 행의 시작점의 좌표를 저장하여 행마다 정렬을 실행한다.

### - 해결 과정

우선 srand()함수를 통해 seed 값을 시간으로 설정한다. 그후 각 행의 시작 주소값을 저장할 int row[4]와 1 차원 배열인 int \*array\_2D 를 생성하고 동적 할당을 통해 int[20]을 할당한다. 그 후 array\_2D 에 0~19 까지의 값을 임의로 할당한 후 Printarr(int\*)를 통해 배열을 4\*5 행렬의 모습으로 출력한다. 그 후 각 행의 첫 번째 값의 주소를 row 배열에 전달하고, 각 행마다 Sortarr(int\*)를 통해 해당 행을 순차정렬을 한다. 그 과정에서 Change(int&, int&)를 통해 각 값이 바뀔 필요가 있을 때 바꾸는 역할을 하게 된다. 정렬이 완료된 후 정렬된 배열을 Printarr(int\*)을 통해 출력하고, 동적할당이 된 array\_2D 를 해제 후 프로그램을 종료한다.

### - 결과 화면

결과

```
선택 Microsoft Visual Studio 디버그 콘솔
11      2      2      12      9
18      10     5      0      3
17      0      3      2      11
9       7      10     8      2

2       2      9      11     12
0       3      5      10     18
0       2      3      11     17
2       7      8      9      10

C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe(프로세스 18956개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 고찰

해당 문제를 푸는 과정에는 크게 어려움은 없었다. 하지만 이 문제를 통하여 1차원 배열을 2차원 배열처럼 사용할 수 있는 것을 알게 되었고, 전달한 주소값을 통해 2차원배열처럼 정렬할 수 있다는 사실이 새로웠다.

## 2. Assignment 2

- 문제 설명

INSERT, FIND, END의 명령어를 받는 단어장 프로그램을 만드는 것이다. 단어는 100개 이하로 저장하고 단어를 알파벳 순과 길이가 짧은 순으로 정렬한다. 그리고 대소문자를 구별하지 않는다. 만약 찾는 단어가 없으면 정해진 출력을 내보낸다. 종료를 입력 받은 경우에만 함수를 종료한다.

- 해결 과정

우선 문자열 배열을 저장할 변수 `char** list`를 생성하고 여기에 `(char*)[100]`의 값을 동적 할당한다. 그리고 `int num`을 생성하여 입력받은 단어의 수를 센다.

그 후 `int Command(char*)` 함수를 통해 명령어를 입력 받는다. 입력 받은 명령어는 `int Mystrcmp(char*, char*)`를 통해 같음을 확인하는데 대소문자를 구분하지 않기 위해 각 단어의 소문자를 비교한다. 만약 단어가 모두 같을 경우 길이를 비교한다. `Mystrcmp()`함수의 반환값이 1일 경우는 앞에 문자가 먼저 나와야 하는 경우, 반환값이 2일 경우는 뒤의 문자가 먼저 와야 하는 경우, 반환값이 0일 경우는 문자가 일치하는 경우다. 만약 입력받은 명령어와 "INSERT"가 일치하는 경우에는 뒤에 입력받은 문자열 `input`을 `words`에 복사하고, 1을 반환한다. "FIND"와 일치하는 경우에는 동일하게 `input`을 `words`로 복사하고, 2를 반환한다. 만약 "END"와 일치하는 경우에는 0을 반환한다.

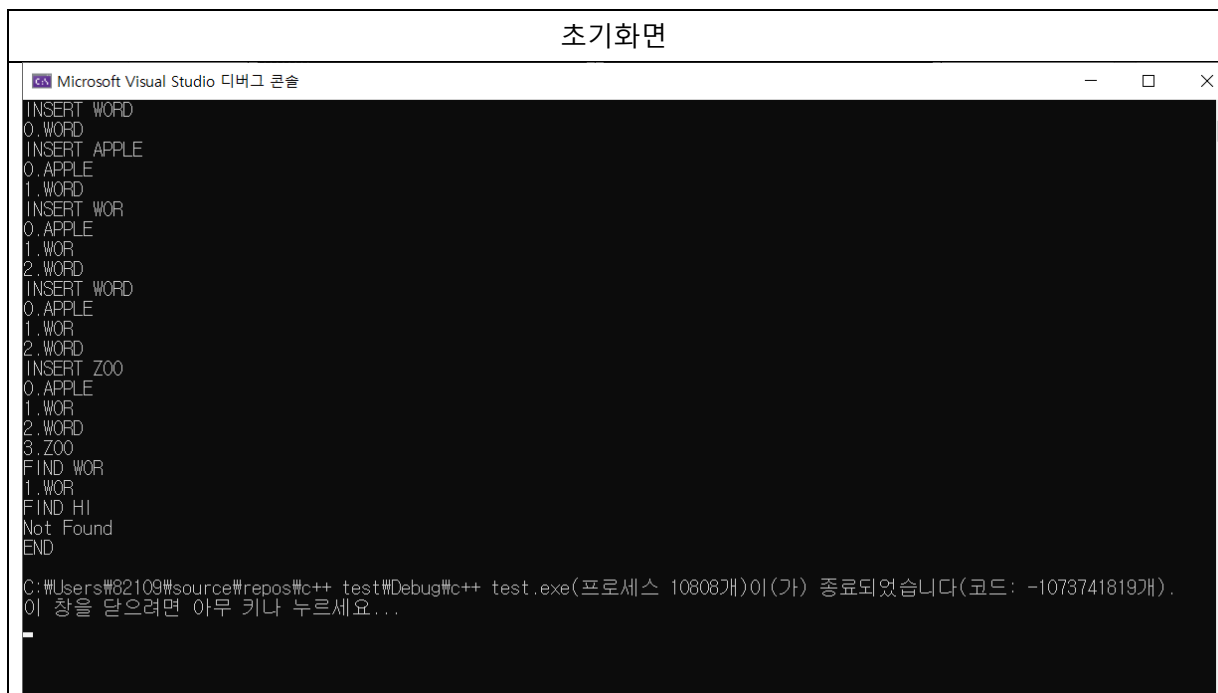
`Command()`의 반환값을 `int cmd`에 저장하고 이 값이 1일 경우에는 `num`의 값이 100 이하인지를 확인하며 이하가 아니라면 "is Fullled"를 출력하고 아닐 경우에는 `list[num]` 위치에 `char[1024]`를 동적할당을 한다. 그 후 `Insert(list, words, num)`을 통해 리스트와 단어, 저장된 단어의 수를 넘긴다. `cmd`의 값이 2일 경우 `Find(list, words, num)`을 통해 리스트를 탐색한다. 증가 0일 경우 그동안 센 `num`까지의 위치의 동적할당을 해제하고 `list`의 동적할당을 해제한 뒤 함수를 종료한다.

`Insert()`함수는 리스트가 아무것도 없을 경우 `words`의 단어를 그대로 리스트로 옮기고 `num`의 값을 1카운트 한 뒤 `PrintVoca(list, end)`를 통해 리스트를 출력한다. 만약 `num`의 값이 0보다 클 경우 `Mystrcmp()`함수를 통해 현재 위치의 단어와 같은 지의 여부를 확인한 후 만약 같으면 리스트를 출력하고 아까 할당했던 `end` 위치는 배열 할당을 해제하고, 함수를 종료한다.

만약 Mystrcmp() 함수의 반환값이 1 일 경우에는 현재 문자가 먼저 와야 하기 때문에 현재 위치의 값부터 한 칸씩 뒷배열로 미는 작업을 수행하고 end 를 카운트한 후 리스트를 출력한다. 만약 이도 아닐 경우에는 다음 배열을 확인하는 것을 반복한다.

Find()함수에서는 처음위치부터 end 의 위치까지 탐색을 진행하는데, 만약 탐색하는 문자와 Mystrcmp()함수를 통해 같다는 결과가 나오면 특정 출력값을 내보내고 함수를 종료한다. 만약 찾지 못할 경우에는 "Not Found"출력을 내보낸 후 함수를 종료한다.

#### - 결과 화면



```
Microsoft Visual Studio 디버그 콘솔
INSERT WORD
0. WORD
INSERT APPLE
0. APPLE
1. WORD
INSERT WOR
0. APPLE
1. WOR
2. WORD
INSERT WORD
0. APPLE
1. WOR
2. WORD
INSERT ZOO
0. APPLE
1. WOR
2. WORD
3. ZOO
FIND WOR
1. WOR
FIND HI
Not Found
END

C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe (프로세스 10808개)이(가) 종료되었습니다(코드: -1073741819개).
이 창을 닫으려면 아무 키나 누르세요...
```

#### - 고찰

해당 과제를 진행할 때 처음에 접근한 방법은 Linked List 를 이용하여 리스트를 계속 추가하는 방식을 사용하고 싶었다. 그러나 과제 당시 Linked list 에 대한 이해의 부족으로 이는 하기 어려웠고, 실패하고 말았다. 그 후 어쩔 수 없이 일반적인 동적할당을 통해 이를 구현하였다. 동적할당을 할 경우 리스트의 크기가 제한된다는 단점이 있고, Linked list 의 경우에는 계속 추가가 가능하다는 장점이 있으나, 파일 탐색 시 처음부터 찾아가야 한다는 단점이 있다. 현재는 융합 과제를 linked list 로 구현하면서 이에 어느정도 숙달이 되었기에 지금 다시 한다면 이를 구현하는데 성공할 수 있을 것이다.

### 3. Assignment 3

- 문제 설명

해당 문제는 비밀번호를 입력하면 비밀번호에 따라 함수 포인터를 통해 결과값을 나타내는 것이다.

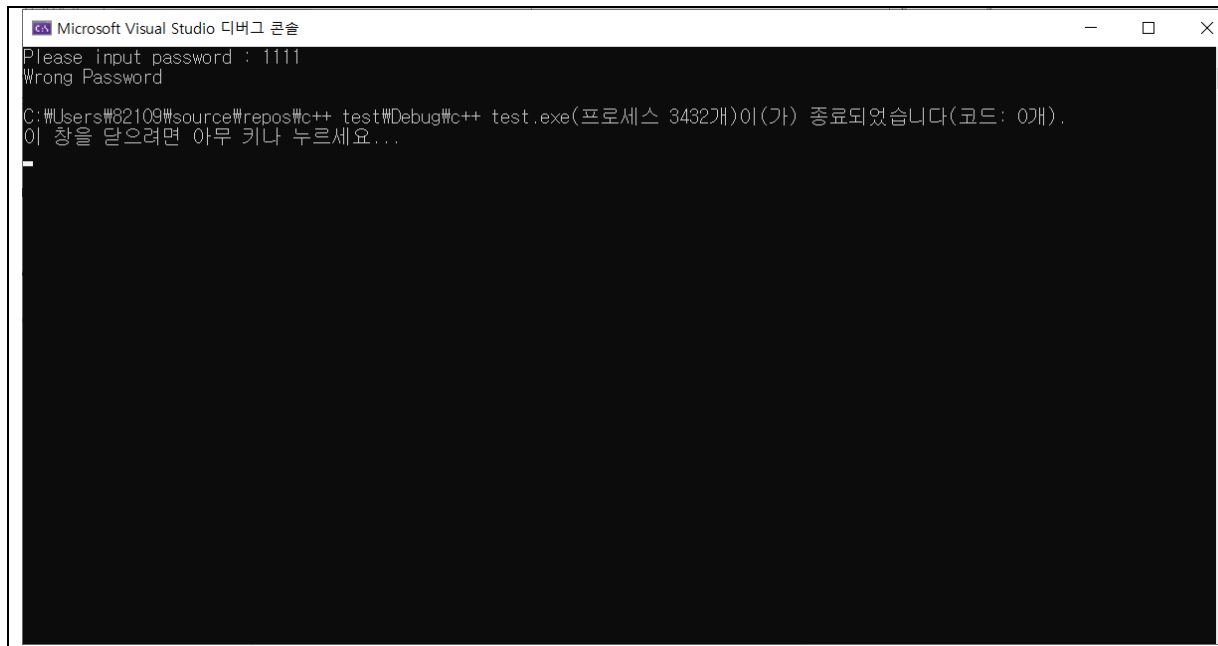
- 해결 과정

우선 정보가 저장된 Infor 라는 구조체를 선언한다. 그리고 PW 는 1234 로 설정하고 사용자에게 int input 을 입력받는다. 만약 비밀번호와 입력이 동일한 경우, void (\*fp)()의 값을 Rightprint 로 바꾸고 fp()를 통해 Rightprint()를 실행시킨다. 만약 입력이랑 다를 경우 fp 의 값을 Wrongprint 로 바꾸고 fp()를 통해 Wrongprint()를 실행시킨다.

Rightprint()함수는 Infor 형 변수 infor 를 선언하고, 이를 형식에 맞게 출력한다. Wrongprint()함수는 "wrong Password" 문장을 출력한다.

- 결과 사진

입력 : 1234	
	
입력 : 1111	



```
Microsoft Visual Studio 디버그 콘솔
Please input password : 1111
Wrong Password
C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe(프로세스 3432개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 고찰

해당 과제는 함수 포인터를 사용할 수 있는 능력을 키워주는 과제였다. 그리고 비밀번호가 어떻게 구현되는지를 알 수 있는 시간이었다. 만약 비밀번호를 숫자 이외의 문자도 받으려 하게 만들면 char PW 로 선언하고 strcmp 를 통해 같은 문자인지를 확인하는 방법으로 바꾸면 가능하겠다는 생각이 들었다.

#### 4. Assignment 4

- 문제 설명

해당 과제는 9\*9 행렬에 0~80 의 값을 오름차순으로 할당한 후 3\*3 행렬끼리의 행렬 곱을 한 값을 누적하여 저장하고 순서대로 출력하는 문제다.

- 해결 과정

우선 int matrix[9][9]와 aftermatrix[9][9]를 통해 9\*9 행렬 2 개를 준비하고 matrix 는 0 부터 80 까지의 값을 순차적으로 할당하고 aftermatrix 에는 0 의 값을 할당한다. 그 후 int count 는 순서를 표시하는 역할을 한다. 행렬의 곱의 방식을 보면 만약 2 개의 3\*3 행렬을 곱한다면 가정했을 때 C(0,0) 위치에 더할 값은  $\sum_{x=0}^2 A(0,0+x) * B(0+x,0)$ 로 이루어진 다는 것을 볼 수 있다. 이를 먼저 3\*3 행렬들끼리의 큰 틀로 보았을 때는 행렬 곱 함수에 들어가는 3 가지 좌표 ax, ay or bx, by 는 (3 \* (i % 3), 3 \* k, 3 \* (j % 3))이 된다는 것을 알 수 있다. 이 값을



```
Microsoft Visual Studio 디버그 콘솔
25164 25848 26532 27216 27900 28584 3303 3522 3741
N = 26
1836 1872 1908 1944 1980 2016 2052 2088 2124
4752 4869 4986 5103 5220 5337 5454 5571 5688
7668 7866 8064 8262 8460 8658 8856 9054 9252
10584 10863 11142 11421 11700 11979 12258 12537 12816
13500 13860 14220 14580 14940 15300 15660 16020 16380
16416 16857 17298 17739 18180 18621 19062 19503 19944
19332 19854 20376 20898 21420 21942 22464 22986 23508
22248 22851 23454 24057 24660 25263 25866 26469 27072
25164 25848 26532 27216 27900 28584 29268 29952 30636
N = 27
1836 1872 1908 1944 1980 2016 2052 2088 2124
4752 4869 4986 5103 5220 5337 5454 5571 5688
7668 7866 8064 8262 8460 8658 8856 9054 9252
10584 10863 11142 11421 11700 11979 12258 12537 12816
13500 13860 14220 14580 14940 15300 15660 16020 16380
16416 16857 17298 17739 18180 18621 19062 19503 19944
19332 19854 20376 20898 21420 21942 22464 22986 23508
22248 22851 23454 24057 24660 25263 25866 26469 27072
25164 25848 26532 27216 27900 28584 29268 29952 30636
C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe(프로세스 9132개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 고찰

해당 문제에서는 행렬의 곱의 규칙성을 이끌어내는 것이 어려움을 느꼈다. 가장 효율적으로 행렬의 곱을 구하는 데에 생각을 많이 해보았고 규칙성을 함수를 표현하기 위한 고민을 가질 수 있는 기회였다. 이 과제를 통해 규칙성을 찾는 안목을 키우고, 선형대수학에서 배우는 행렬에 대한 이해도 또한 키우는 계기가 되었다.

## 5. Assignment 5

- 문제 설명

해당 문제는 class 를 사용하여 char\* MyStrcpy(), 즉 문자열 복사 함수와 int Find\_char(), 문자 탐색 함수를 구현하는 문제다.

- 해결 과정

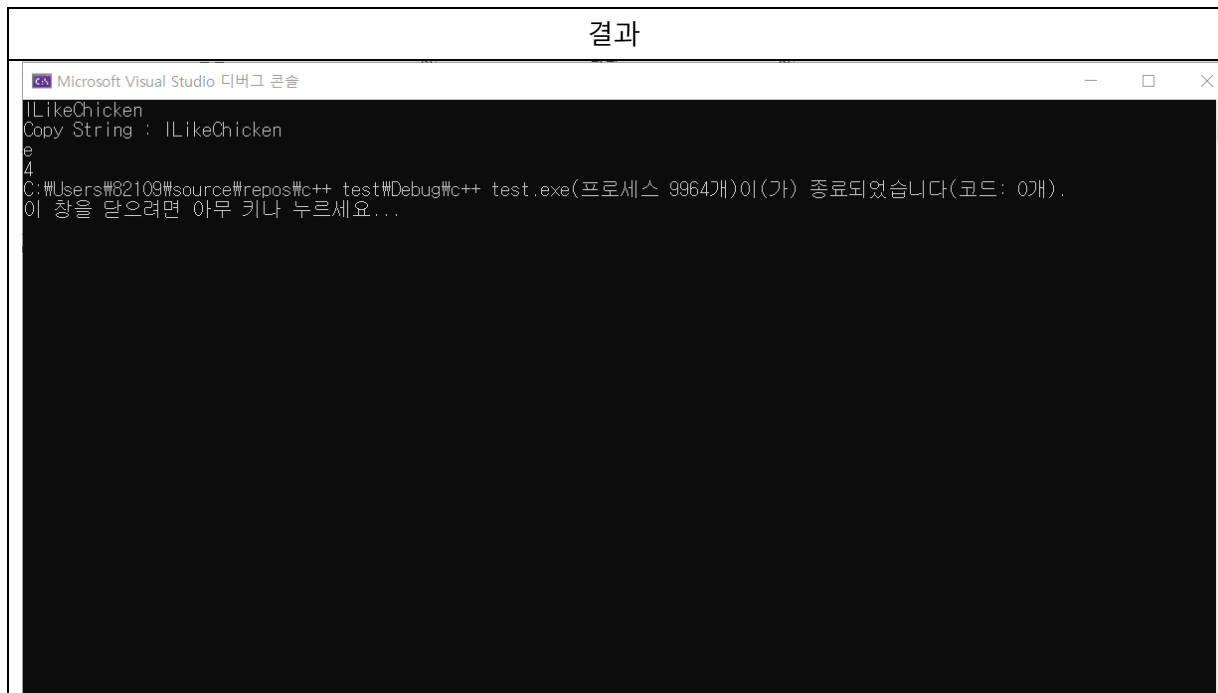
우선 char inputStr[1024]를 통해 문자열을 입력 받는다. 그 후 입력받은 문자의 길이를 측정하는데 int length=0 를 'w0'가 나오기 전까지 카운트하여 길이를 구한다. 그리고 str 클래스에 str.MyStrcpy(inputStr, length) 함수의 반환값을 정해진 형식으로 출력한다.

MyStrcpy()함수는 주어진 길이만큼을 src 에서 MyStr 변수로 복사한 후 마지막에 'w0'를 추가한 뒤 MyStr 을 반환한다.

위의 과정을 마친 후 inputChar 을 입력 받는다. 입력한 inputChar 는 str.Find\_char()의 인자값으로 쓰인다. 그리고 이 인자를 정해진 형식으로 출력 후 프로그램은 종료된다.

Find\_char()함수는 'W0'를 만날 때까지 반복하며 같은 문자를 찾을 경우 해당 위치 i 를 반환하고 찾지 못했을 경우 -1 을 반환한다.

- 결과 사진



- 고찰

해당 과제를 통해 우리가 흔히 코딩을 하는 데에 많이 사용하는 함수인 strcpy 와 strlen 등의 함수가 어떻게 구현되는 지를 알 수 있는 시간이었다. 또한 class 의 대한 기본적인 활용을 함으로써 클래스를 어떻게 사용해야하는 지 알 수 있던 기본을 다지는 시간이 되었다.

## 6. Assignment 6

- 문제 설명

해당 문제는 MyAccount 클래스에 private 인자(돈, 이름, 계좌번호)를 갖고 이를 5 개짜리 array 로 선언하고(단 동적할당 사용) 5 개의 명령어(NEW, DWPOSIT, WITHDRAW, PRINT, EXIT)을 통해 조작하는 문제이다.

- 해결 과정

우선 MyAccount\* account 클래스를 선언한다. 그리고 이를 동적할당을 통해 5 개의 MyAccount 를 할당한다. 그 후 char cmd[12]를 통해 명령어를 입력 받고, 이 입력을 strcmp() 함수를 통해 "NEW", "DEPOSIT", "WITHDRAW", "PRINT", "EXIT" 명령어와 비교하여 동일할 경우 각각의 명령을 수행한다.



“NEW”와 같은 경우 만약 저장된 account 의 수(count)가 5 개 이상이라면 더 이상 저장할 수 없다는 말을 출력한다. 만약 아닐 경우 account[count].NewAccount() 함수를 통해 해당 계좌를 생성하고 count 변수를 카운트한다. 해당 함수는 이름과 계좌 번호를 동적할당을 하여 각각 저장하고 계좌생성을 출력한다.

“DEPOSIT”이나 “WITHDRAW”을 입력 받은 경우 count 가 0 일 시 계좌가 없다는 문구를 출력한다. 만약 계좌가 있을 경우 for 문을 하여 이미 만들어진 account 들을 DepositAccount() 함수와 WithdrawAccount() 함수를 각각 사용하여 역할을 수행한다.

DepositAccount()함수는 현재 class 의 계좌번호와 입력 받은 계좌번호를 확인한 후 같을 때 입력받은 금액만큼 누적한다.

WithdrawAccount()함수는 현재 class 의 계좌번호와 입력 받은 계좌번호를 확인한 후 같을 때 입력받은 금액만큼 차감한다.

“PRINT”를 입력 받은 경우 PrintAccount()함수를 통해 이름과 계좌번호, 금액을 출력한다.

“EXIT”을 입력 받은 경우 DestructAccount() 함수를 통해 이름과 계좌번호의 동적할당을 해제하고, 이가 모두 완료되면 account 의 동적할당을 해제한 후 함수를 종료한다.

#### - 결과 사진

결과

```

선택 Microsoft Visual Studio 디버그 콘솔
DEPOSIT
Number of Account is 0
WITHDRAW
Number of Account is 0
NEW 이준휘 2018202046-1
계좌생성
PRINT
0 이름 : 이준휘 ACount_Number : 2018202046-1 Money : 0
NEW XXX 1111-111
계좌생성
PRINT
0 이름 : 이준휘 ACount_Number : 2018202046-1 Money : 0
1 이름 : XXX ACount_Number : 1111-111 Money : 0
DEPOSIT 2018202045-1 700
PRINT
0 이름 : 이준휘 ACount_Number : 2018202046-1 Money : 0
1 이름 : XXX ACount_Number : 1111-111 Money : 0
DEPOSIT 2018202046-1 500
PRINT
0 이름 : 이준휘 ACount_Number : 2018202046-1 Money : 500
1 이름 : XXX ACount_Number : 1111-111 Money : 0
WITHDRAW 2018202046-1 200
PRINT
0 이름 : 이준휘 ACount_Number : 2018202046-1 Money : 300
1 이름 : XXX ACount_Number : 1111-111 Money : 0
EXIT
계좌삭제 :0
계좌삭제 :1
C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe(프로세스 7740개)이(가) 종료되었습니다(코드: 0개).

```

#### - 고찰

해당 과제를 수행하면서 class 를 어떻게 활용할 수 있는 지에 대해 더욱 자세하게 알게 되었다. 과제에 나온 것처럼 class 단위로 데이터를 관리하고 조작하면 좀 더 효율적으로 데이터를 이용할 수 있고, private 를 통해 데이터를 안전하게 보관할 수 있는 사실을 알 수 있었다. 앞으로 class 를 잘 활용하면 더욱 많은 작업을 효율적으로 수행할 수 있을 것이라는 생각 또한 드는 과제였다.

## 7. Assignment 7

### - 문제 설명

해당 문제는 class 대학, 컴퓨터정보공학, 학생정보를 활용하고, "INSERT"를 입력 받을 시 데이터를 입력 받아 저장하고, "FIND"를 통해 학생 정보를 탐색하며 "EXIT"을 통해 프로그램을 종료한다.

### - 해결 과정

우선 대학 클래스는 컴퓨터정보공학 클래스에 상속되어 있고, 컴퓨터정보공학 클래스는 학생정보 클래스에 상속되어있다. cmd 를 통해 명령을 입력받는다. 학생정보는 학생정보\*\* studentinfo 에 저장되어있고, 이는 동적할당 되어있다. 그리고 \*studentinfo 는 학생정보[100] 크기의 배열을 할당한다.

"INSERT"를 입력한 경우 입력된 학생정보가 100 개 미만일 경우 NewInfo((\*studentinfo)[count])함수를 통해 입력을 받고 count 를 1 증가시킨다. 배열이 100 개 이상일 경우 특정 문자열을 출력한다. NewInfo(학생정보&) 함수는 학생 정보를 studentinfo.Input\_XXXX()을 통해 입력받아 클래스에 저장한다.

"FIND"를 입력한 경우 이름과 ID 를 입력받고 FindInfo((\*studentinfo)[count], inputname, inputID)를 호출한다. 이 함수에서는 입력받은 이름과 ID 가 같을 경우 해당 입력된 정보를 studentinfo.Get\_XXXX 를 통해 값을 얻어 출력하는 함수이다.

"EXIT"을 입력받은 경우 \*studentinfo 배열을 동적할당 해제하고 studentinfo 의 할당을 해제하고 프로그램을 종료한다.

### - 결과 사진

결과
----

```
Microsoft Visual Studio 디버깅 콘솔
INSERT
학교 : 광운대학교
학번 : 2018202046
학년 : 2
객체지향프로그래밍 학점 : A
디지털논리회로 학점 : B
이름 : 이준휘
나이 : 23
INSERT
학교 : 광운대학교
학번 : 2000000000
학년 : 1
객체지향프로그래밍 학점 : C
디지털논리회로 학점 : D
이름 : XXX
나이 : 22
FIND 이준휘 2018202046

학교 : 광운대학교
학번 : 2018202046
학년 : 2
객체지향프로그래밍 학점 : A
디지털논리회로 학점 : B
EXIT
C:\Users\82109\source\repos\c++ test\Debug\c++ test.exe(프로세스 13392개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

#### - 고찰

해당 문제는 상속된 class 를 활용하는 방법을 이해할 수 있는 과제였다. 상속을 통해 데이터를 따로 관리하면서 데이터를 효율적으로 저장할 수 있다. 만약 여기서 다른 학교 컴퓨터정보공학이 있다면 다른 학교 클래스를 만들고 이를 대학에 할당하고 학생정보를 할당하는 방식으로 데이터를 관리할 수도 있다. 그리고 private 인자의 값을 저장하거나 가져올 때 함수를 이용하여 이를 가져오는 방식도 활용해볼 수 있었다.

## 8. Assignment 8

#### - 문제 설명

해당 문제는 동물 레이싱 게임으로 동물의 순서를 입력받고 이를 정해진 Animal 클래스를 활용하여 레이스를 하는데 만약 같은 위치에 동물이 2 마리 이상일 경우 먹이사슬에 따라 한 동물은 다시 시작지점으로 간다. 모두 도착했을 경우 프로그램은 종료한다.

#### - 해결 과정

Animal class 에는 기존에 있던 생성자와 가상함수 run(), 그리고 거리를 입출력하기 위한 GetDistance()와 PushDistance()함수, 도착여부를 입출력하기 위한 GetArrival()와 PushArrival()가 존재한다.

각 동물의 클래스는 Animal 을 상속받는다. 해당 클래스들에는 순서를 저장하기 위한 int seq 변수와 생성자에 각 동물에 맞는 먹이사슬이 할당되어있다. 그리고 run 에는 각 동물이 가는 거리가 Getdistance() 함수와 PushDistance()함수를 통해 조작되어있다.

메인함수에서는 각 동물의 클래스를 불러오고, 동물의 순서를 입력받는다. 그리고 입력받은 순서에 따라 각 동물의 seq 에 값을 할당한다. 그리고 char arr[11]에는 동물들의 현재 위치를 나타낼 문자열이다.

각 동물의 순서가 맞으면서 아직 도착상태가 아닐 경우, 각 동물의 거리를 run()을 통해 이동한다. 그리고 각 동물의 거리가 10을 넘어갈 경우 PushArrival()을 통해 true로 값을 바꾼다. 10 이 넘어가지 않았을 경우 문자열에 위치를 넣는다.

각 동물의 위치를 개구리부터 비교하여 겹치는 동물이 있을 경우 개구리는 위치를 0 으로 바꾸는 방식으로 독수리까지의 동물을 다른 동물(먹이사슬이 높은)과 비교하여 0 으로 만든다.

그 후 배열을 출력하고 만약 모든 동물이 도착했을 경우 프로그램을 종료한다.

#### - 결과 사진

결과

Microsoft Visual Studio 디버그 콘솔

```

Select Animal Sequence(E,S,E,T / ex : S F T E) :E S E T
0 F 0 0 0 0 0 0 0 0
0 S 0 F 0 0 0 0 0 0
0 S 0 T 0 0 0 0 0 0
0 S 0 F 0 0 0 0 0 0
0 S 0 S 0 0 0 0 0 0
0 0 0 S 0 0 0 0 0
0 0 0 0 0 S 0 0 0
0 0 0 0 0 S 0 0 0
0 0 0 0 0 0 0 S 0
0 0 0 0 0 0 0 S 0
0 0 0 0 0 0 0 0 S
0 0 0 0 0 0 0 0 0

C:\Users\#82109#source#repos#c++ test#Debug#c++ test.exe(프로세스 6456개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

#### - 고찰

해당 과제를 만들면서 가상 함수라는 개념을 직접 적용해볼 수 있는 기회가 되었다. 가상함수를 통해 각각의 클래스에서 같은 이름으로 다르게 동작할 수 있는 함수를 만들 수 있다는 부분은 활용성이 높다고 생각했다. 또한 생성자를 통해 상속한 클래스의 인자값을 보낼 수 있다는 사실을 알게 된 과제였다.

## 9. Assignment 9

### - 문제 설명

해당 문제는 주어진 메인함수만을 사용해야 한다. Mystring 클래스를 활용해서 `cout << class` 로 값이 출력되도록 연산자 operator 를 사용해야 한다.

### - 해결 과정

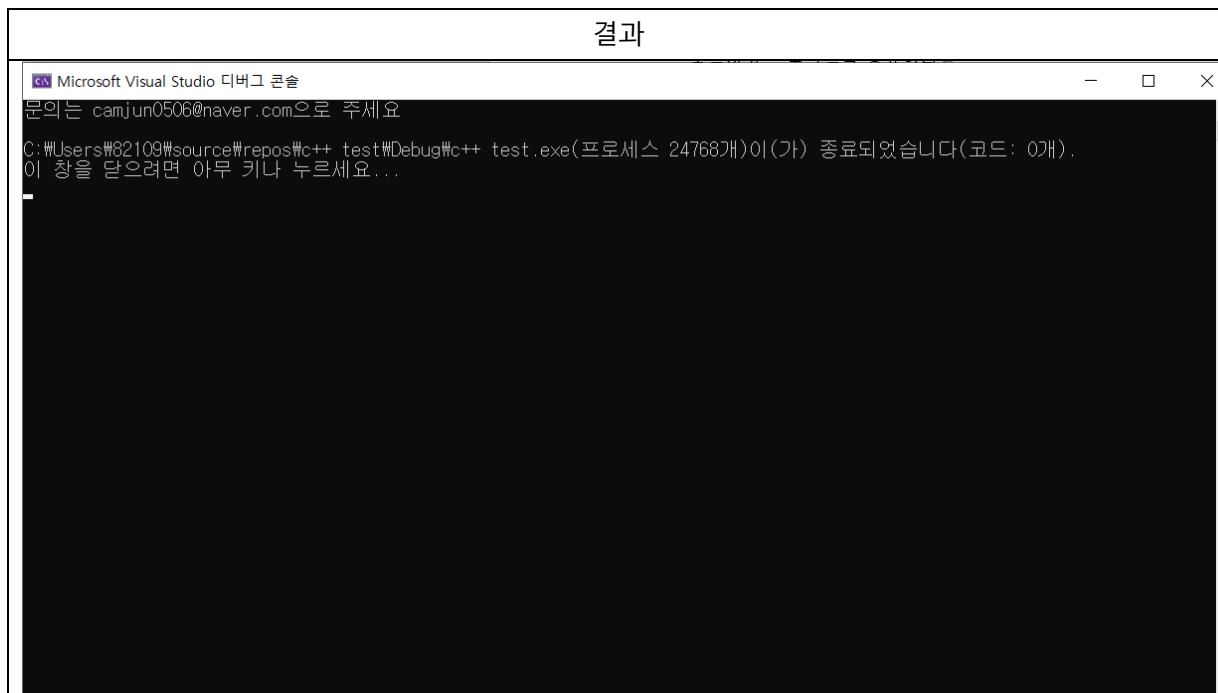
Mystring class 는 `char *str` 값을 가지고 있다. 그리고 `const char*` 인자를 받고 생성될 경우 `str` 에 동적할당을 하고 여기에 `strcpy` 를 통해 문자열을 복사한다.

=연산자를 받을 경우 지금 위치의 `str` 을 동적할당하고 Mystring 의 `str` 값을 이곳으로 복사하고 현재 클래스를 반환한다.

+연산자를 만날 경우 오른쪽의 `const char*`의 문자열과 `str` 문자열을 `strcat()` 함수를 통해 합치고 이를 현재 class 의 `str` 에 다시 저장한다. Friend 로 연결된 `ostream` 의 <<연산자는 Mystring 을 만났을 경우 `str` 을 출력하고 `os` 를 반환하도록 한다. 소멸자는 클래스의 `str` 을 할당 해제한다.

메인 함수는 정해진 양식으로 인자가 없는 클래스와 "문의는 "이 인자인 클래스를 선언하고 인자가 없는 클래스에 인자가 있는 클래스와 문자열을 합한 값을 전달하고 이 합한 클래스를 출력하고 프로그램을 종료한다.

### - 결과 사진



- 고찰

해당 과제를 구현하면서 생성자와 소멸자에 대해 더욱 심도있게 알 수 있었던 기회가 되었다. 생성자를 통해 동적할당을 하고 소멸자로 동적할당을 해제하는 효율적인 방법을 알 수 있었다. 그리고 연산자를 통해 이 클래스와 다른 값들을 조작할 수 있다는 사실을 직접 확인할 수 있는 기회였다.