

객체지향프로그래밍 실습

결과 보고서

학 과: 컴퓨터정보공학부

담당교수: 이형근 교수님

실습분반: 화요일(7,8교시)

학 번: 2018202028

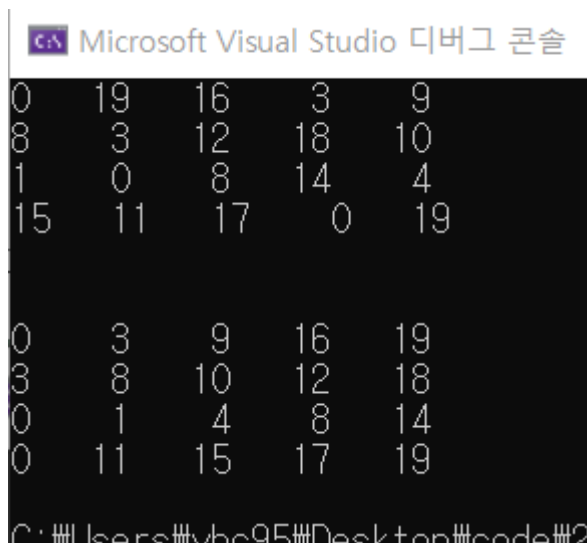
성 명: 유해찬

<1번문제>

(1) 문제 및 설명

이 문제는 주어진 변수들을 통해서 4*5행렬의 각 row만 정렬시키는 문제이다. 변수로 주어진 `int *array_2D`에 20만큼의 크기를 할당하여 배열같이 써서 0~19까지의 숫자들을 랜덤으로 저장하고 배열의 각행의 첫번째 주소를 `int *row[4]`에 저장하여 sort를 실행한다.

(2) 결과화면



```
Microsoft Visual Studio 디버그 콘솔
0 19 16 3 9
8 3 12 18 10
1 0 8 14 4
15 11 17 0 19

0 3 9 16 19
3 8 10 12 18
0 1 4 8 14
0 11 15 17 19
```

결과 화면에 대해서 두 번째 줄에 2,4,5,12,18이 0~19까지의 숫자가 랜덤으로 돌아간 뒤에 row를 기준으로 sort가 된 모습을 볼 수 있다.

밑에 줄도 마찬가지로 전체에 대해서는 sort가 되어있지 않다.

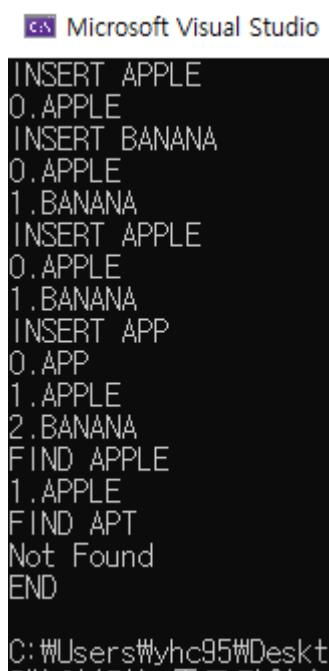
(3) 고찰

이 문제에서 가장 중요한 것은 배열의 대한 이해이다. 1차원 배열과 2차원배열에서는 설명할 때에는 따로 떨어져서 row마다 메모리주소값을 보여주면서 보통 설명을 하는데 사실은 하나의 포인터로 주소값이 int형이면 4바이트씩 이어져있다. Column의 MAX값을 지나면 그 다음은 row가 바뀌는데 `row[0]`의 값이 Column MAX의 다음 주소값이다. 또한, `row[4]`의 포인터에 row값의 시작하는 점의 각 주소 값을 입력하여 각 row마다의 정렬이 쉬워지게 되었다. 이를 통해서 1차원포인터와 2차원포인터의 관계를 알게되었다.

<2번문제>

(1) 문제 및 설명

이 문제는 단어장을 만드는 문제이다 Command값에 INSERT를 입력하고 단어를 넣어주면 사전식으로(대소문자구별없이,알파벳순으로,알파벳순이 같다면 길이를 그 다음으로 비교해준다) 정렬을 하고 만약 같은 문자가 이미 단어장에 있으면 INSERT를 하지 않는다. 또한 단어장의 단어 수가 100개가 넘어간다면 is Full을 출력한다. 또한, FIND함수를 이용하여 저장된 값을 대소문자 구별을 하지 않고 똑 같은 값을 찾고 저장된 값의 단어장의 번호와 단어를 출력한다. 만약 존재하지 않는다면 NOT Found를 출력한다.



```
Microsoft Visual Studio
INSERT APPLE
0. APPLE
INSERT BANANA
0. APPLE
1. BANANA
INSERT APPLE
0. APPLE
1. BANANA
INSERT APP
0. APP
1. APPLE
2. BANANA
FIND APPLE
1. APPLE
FIND APT
Not Found
END
C:\Users\whc95\Desktop
```

INSERT 단어를 입력하면 단어가 그 즉시 index 0 번부터 저장되고 출력되는 것을 알 수 있다. 그 후 BANANA 를 입력하였는데 이것을 보고 BANANA 의 B 는 APPLE 의 A 보다 뒤이기 때문에 뒤로 간 것을 알 수 있다.

APPLE 을 INSERT 했는데 다시 APPLE 을 입력하였더니 무시한 것을 볼 수 있다.

그 후 APP 를 INSERT 를 하였더니 APPLE 보다 모두 같은데 글자수 가 제일 적기 때문에 맨 앞으로 온 것을 볼 수 있다.

FIND APPLE 을 하니까 그 INDEX 번호와 단어가 출력되는 것을 볼 수 있다.

FIND APT 는 없는 단어 이므로 Not Found 를 출력한다.

(2) 고찰

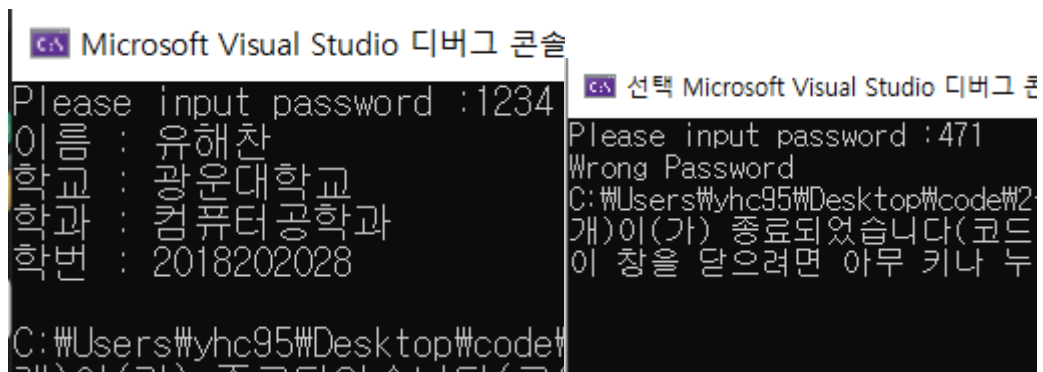
이 문제는 특이하게 단어가 대소문자를 구별하지 않는다는 조건 때문에 구현하기 힘들었다. 함수를 총 단어장을 출력하는 print함수 단어를 찾고 단어의 번호를 반환하는 find_equal함수 단어를 입력받고 정렬하는 sort함수를 구현하였는데 이 함수들에서 print함수를 제외하고 모든 경우에서 대문자 소문자 구별을 한다는 게 너무 어려웠다. Find_equal에서는 temp라는 함수를 사용하여 대문자를 입력하면 소문자를 소문자를 입력하면 대문자를 받아서 temp 값에 저장하여 하나씩 단어들을 비교해보고 맞는 단어가 있으면 그 단어의 위치를 반환해준다. 없다면 -1을 반환한다. 가장 중요한 Sort에서는 케이스를 2개,4개로 나누어서 비교를 해주었다. 입력한 값과 비교하는 값의 길이를 보고 비교를 하는 케이스 2개와 입력하는 값과 비교하는 값이 각각 (대문자,대문자), (대문자,소문자), (소문자,대문자), (대문자,대문자) 일 때를 비교하고 들어갈 단어의 위치를 찾고 그 곳이 맞다면 그 위치에 있는 단어들을 모두 한칸씩 밀고 빈자리에 INSERT한 단어를 넣어주었다. 이 두가지 함수를 이용하여 나머지 FIND와 INSERT를 구현하고 END로 사용한 메모리들을 모두 할당을 해제를 해주었다. 예외처리 조건과 Sort에서 많은 애를 먹은 문제이다.

<문제 3번>

(1) 문제 및 설명

이 문제는 함수 포인터를 사용할 수 있는지를 물어보는 문제이다. 비밀번호를 받고 비밀번호가 만약 1234가 맞으면 Rightprint()함수를 호출 아니면 Wrongprint()를 호출하는 문제인데 여기서 Rightprint()와 Wrongprint()는 직접적으로 호출하면 안되고 void(*fp)라는 함수포인터를 이용하여 호출해야한다.

(2) 결과 화면



```
C:\Users\yhc95\Desktop\code\2018202028\main.cpp
Please input password :1234
이름 : 유해찬
학교 : 광운대학교
학과 : 컴퓨터공학과
학번 : 2018202028
C:\Users\yhc95\Desktop\code\2018202028\main.cpp
Please input password :471
Wrong Password
C:\Users\yhc95\Desktop\code\2018202028\main.cpp
이(가) 종료되었습니다(코드 0)
이(가) 종료되었습니다(코드 0)
```

비밀번호를 1234를 입력하니 Rightprint()함수에 적은 값이 정상적으로 함수포인터를 통해 출력되었다.

하지만 471이라는 다른 값을 입력하니 Wrongprint()함수에서 구현한 값이 함수포인터를 통해 정상적으로 출력된 모습을 볼 수 있다.

(3) 고찰

함수 포인터의 사용에 의의를 두고 낸 문제 같다. 함수도 함수포인터라는 변수를 이용해서 주소값을 저장한 다음 함수를 마음대로 쓸 수 있다는 것을 깨달았다.

<문제 4번>

(1) 문제 및 설명

0~80까지의 9*9행렬을 3*3행렬 9개로 쪼개서 쪼갠것에 대해서 행렬의 곱셈을 진행하면 된다. Arr[9][3][3]행렬에 각 번호마다 9*9행렬을 쪼개서 저장한 뒤에 행렬의 곱을 번호마다 불러서 시행했다.

(2) 결과화면

```

Microsoft Visual Studio 디버그 콘솔
22248 22851 23454 24057 24660 25263 25898 3090 3282
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 25898 3090 3282
25164 25848 26532 27216 27900 28584 3303 3522 3741
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 25898 3090 3282
25164 25848 26532 27216 27900 28584 3303 3522 3741
C:\Users\yh95\Desktop\code\2-1학기(유해찬)\백프실(최상호)\과제\2차\Assignment2-4\Debug\Assignment2-4.exe(프로세스 32832)이(가) 종료되었습니다(코드: 0x).
이 창을 닫으려면 아무 키나 누르세요...
  
```

크게 9*9행렬을 3*3으로 나눈 값에 번호를 붙여 그림과 같이 생각한다면



각 번호에 3*3의 곱셈을 해준값을 차례대로 더해주어서 총 9칸이기 때문에 한칸에 3번씩
이므로 27번을 반복하는 것을 볼 수 있다. 그림에서는 N이 26이기 때문에 9번째의 3*3행

렬과 9번째의 3*3행렬의 곱의 값을 더해주는 것을 볼 수 있다.

(3) 고찰

생각은 쉽지만 구현은 어려운 문제였다. 생각대로 하면 구현하기 쉽지만 for문과 정해진 변수는 정해진 로직대로만 움직이다 보니 너무 어려웠다. 3중 for문에서 i j k값만으로 주어진 행렬의 곱셈에서 값들을 지정해준다는 것이 너무 어려워서 시간이 제일 많이 걸렸다. 2차원 배열로도 가능하지만 각 행렬의 row와 column간의 관계를 파악하기가 힘들어서 그러지 못했다.

<문제 5번>

(1) 문제 및 설명

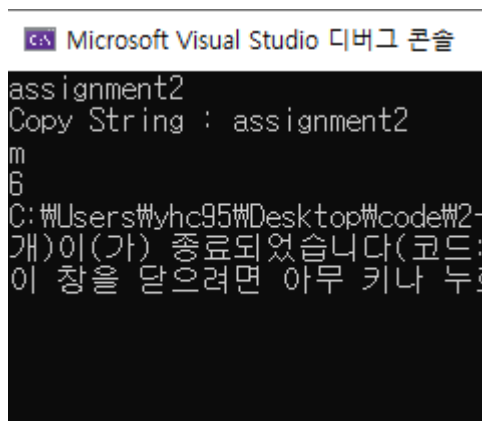
Mystring 객체를 strcpy같은 이미 bulit-in string funtion을 쓰지않고 strcpy와 find_char함수를 구현하는 문제이다.

Mystring에서 Mystrcpy를 구현할 때 처음으로 받은 매개변수의 length를 구하고 Mystr에 매개변수의 값이 'W0'이 나올때까지 복사해주고 Mystr의 마지막에 'W0'문자를 넣어준다.

그후 mystr을 출력해야하는데 private 값이므로 get함수를 구현하여 class안에서 this->mystr을 return하여 값을 받아온다.

Find index를 할 때는 index의 값에 -1을 넣어주고 for문을 'W0'이 만날 때까지 돌려서 만약 값이 있을경우는 그 즉시 index값을 return해주고 아니면 index의 초기화 값인 -1을 반환해주어서 출력하게한다.

(2) 결과화면



```
Microsoft Visual Studio 디버그 콘솔
assignment2
Copy String : assignment2
m
6
C:\Users\yhc95\Desktop\code\2-
개)이(가) 종료되었습니다(코드:
이 창을 닫으려면 아무 키나 누
```

첫번째에는 string 단어를 입력하고 그 단어를 Mystring이라는 class로 생성된 객체에 있는 char *변수에 copy해주는 행동을 하고 그 값을 반환하여 출력한다.

두번째 입력은 원하는 글자를 find하는 것인데 m을 입력하고 assignment2에 m이 있는 위치의 index값이 출력되는 것을 볼 수 있다.

(3) 고찰

Class라는 것을 처음 사용하는 문제여서 많이 생소했었다. Private의 값을 이용하기 위해서는 get함수라는 개념이 필요한데 나머지 문제에서도 자주쓰이는 개념을 미리 알아서 나머지 문제들이 수월했다. 또한, for문의 조건을 'W0'이 만날때까지라고 정의를 해서 'W0'이

안들어가있는 상태라서 값을 처음에 출력을 했을 때 복사된값뒤에 쓰레기값들이 나왔었다. 'w0'이 char* 에서는 필수적이라는 것을 느꼈다.

<문제 6번>

(1) 문제 및 설명

클래스의 매소드 활용을 물어보는 문제 같았다. NEW 명령어를 입력하면 이름과 계좌번호를 만들고 총 5개까지의 계좌까지 만들 수 있게 제한한다. 생성자는 기본생성자와 오버로딩을 해서 계좌와 이름을 받는 생성자를 만들고 그 안에 값을 넣어준다. 그리고 money의 값은 0으로 초기화를 해준다.

DEPOSIT을 Command에 입력하면 돈에 예금을 해준다. 메인 함수에서 입력한 계자와 똑같은 계좌가 있는지 지금까지 생성한 계좌만큼 포문을 돌려 Deposit 매소드에 가면 객체에 저장된 Account와 내가 입력한 계좌를 하나씩 비교를 하고 만약 같다면 예금을 하도록 함수를 구성하였다.

WITHDRAW도 DPOSIT과 마찬가지로 메인 함수에서 입력한 계자와 똑같은 계좌번호가 있는지 지금까지 생성한 계좌만큼 포문을 돌려 매소드에서 객체에 저장된 계좌번호와 입력 받은 계좌번호를 하나씩 비교하여 만약 같다면 돈을 출금 시키는 것으로 구성하였다.

EXIT에서는 지금까지 만들 객체들의 메모리를 동적할당 해제를 포문을 돌려서 완성하였다

(2) 결과화면

```
Microsoft Visual Studio 디버그 콘솔
NEW 유해찬 213-54-123
계좌생성
NEW 해찬유 6677-12-435
계좌생성
NEW 찬해유 6412-43-231
계좌생성
PRINT
0 이름 : 유해찬 Account_Number : 213-54-123 Money : 0
1 이름 : 해찬유 Account_Number : 6677-12-435 Money : 0
2 이름 : 찬해유 Account_Number : 6412-43-231 Money : 0
DEPOSIT 213-54-123 5000000
PRINT
0 이름 : 유해찬 Account_Number : 213-54-123 Money : 5000000
1 이름 : 해찬유 Account_Number : 6677-12-435 Money : 0
2 이름 : 찬해유 Account_Number : 6412-43-231 Money : 0
DEPOSIT 6677-12 500
PRINT
0 이름 : 유해찬 Account_Number : 213-54-123 Money : 5000000
1 이름 : 해찬유 Account_Number : 6677-12-435 Money : 0
2 이름 : 찬해유 Account_Number : 6412-43-231 Money : 0
WITHDRAW 213-54-123 100
PRINT
0 이름 : 유해찬 Account_Number : 213-54-123 Money : 4999900
1 이름 : 해찬유 Account_Number : 6677-12-435 Money : 0
2 이름 : 찬해유 Account_Number : 6412-43-231 Money : 0
EXIT
EXIT
계좌삭제 :0
계좌삭제 :1
계좌삭제 :2
```

NEW 이름 계좌번호 를 입력하면 계좌가 생성자를 통해 동적할당하여 생성되는 것을 볼 수 있다. 또한 PRINT를 입력하면 지금까지 생성된 계좌주의 이름 계좌번호 가지고 있는 돈의 값을 볼 수가있다.

DEPOSIT 계좌번호 돈을 입력하면 계좌번호가 있는지 계좌목록에서 찾은 뒤 돈을 입금해주는 것을 볼 수있다.

그 밑에 DEPOSIT에는 계좌번호를 중간까지만 입력하여 맞는 계좌번호가 없어서 돈이 추가가 되지 않는 것을 볼 수 있다,

WITHDRAW 계좌번호 돈을 입력하면 계좌번호가 있는지 계좌목록에서 찾은 뒤 돈을 출금 해 주는 것을 볼 수 있다.

마지막으로 EXIT를 입력하면 지금까지 생성했던 계좌들의 동적할당을 해제해주는 것을 볼 수 있다.

(3) 고찰

객체의 매소드 구현을 본격적으로 이 문제를 통해서 시작하였다. 계좌의 값을 이 문제를 처음시작 할 때에는 몰랐는데 get과 set함수를 통해서 private 변수를 쉽게 값을 가져올 수 있고 설정 할 수 있다는 것을 알게되었다. 또한, 메모리 동적할당 해제를 위해서 객체에 Name과 Account_Number의 변수에 대한 동적할당을 해줬는데 해제를 어떻게 하는지 잘 몰라서 고생을 많이 했습니다. 또한, 생성한 객체에 대한 메모리 할당 해제를 까먹어서 누수확인을 했을 때 고생을 했습니다.

<문제 7번>

(1) 문제 및 설명

이 문제는 Class의 상속에 대해서 구현하는 것을 물어보는 문제이다. 대학이라는 객체에 컴퓨터정보공학이라는 객체가 상속하고 컴퓨터정보공학이라는 객체에 학생정보라는 객체가 상속을 한다. 대학 <- 컴퓨터정보공학 <- 학생정보 이렇게 상속관계가 있다.

대학에는 학교이름과 학년과 학번을 저장하는 변수들이있고 컴퓨터공학에는 객체지향프로그래밍의 성적과 디지털 논리회로 성적을 받는 변수가 있고 마지막으로 학생정보에는 이름과 나이를 받는 변수가 있다.

이 객체들의 변수를 받는 Command는 INSERT로 INESRT면 객체를 동적할당하여 생성자를 통해 변수들의 값을 넣어준다. 또한 학생들의 정보들의 수가 100이 넘어가면 Number is 100이라는 말을 출력해준다.

FIND는 학번이 같은게 있는지 검사하는 매소드와 이름이 같으게 있는지 검사하는 매소드 두개가 모두 true값을 반환했을 때 그 값을 출력하고 없으면 출력을 하지 않는다.

EXIT를 하면 지금 까지 동적 할당했던 객체인 학생정보**를 해제 해준다.

(2) 결과화면

```
Microsoft Visual Studio 디버그 콘솔
INSERT
학교 :광운대학교
학번 :2018202028
학년 :2
객체지향프로그래밍 학점 :A
디지털논리회로 학점 :A
이름 :유해찬
나이 :23
INSERT
학교 :광운대학교
학번 :2015000000
학년 :4
객체지향프로그래밍 학점 :F
디지털논리회로 학점 :F
이름 :홍길동
나이 :00
FIND 유해찬 2018202028

학교:광운대학교
학번:2018202028
학년:2
객체지향프로그래밍 학점: A
디지털논리회로 학점: A
FIND 홍길동 2015000000

학교:광운대학교
학번:2015000000
학년:4
객체지향프로그래밍 학점: F
디지털논리회로 학점: F
EXIT
C:\Users\ybc95\Desktop\code\2-1\
```

INSERT명령어를 입력하면 학교, 학번, 학년, 객체지향프로그래밍학점, 디지털논리회로 학점, 이름, 나이를 입력하는 창이 뜨고 이 입력을 다 완료했으면 다시 command를 입력하는 창이 뜬다.

FIND 이름 학번을 검색하면 두개가 동일하게 맞다면 그 학생의 학교, 학번, 학년, 과목의 학점정보를 출력하게 되어있다.

EXIT를 하면 동적할당 했던 값들을 해제 해주고 프로그램이 종료가 되는 것을 볼 수 있다.

(3) 고찰

객체의 상속에 대한 개념이 부족하여 하나하나 객체에 변수 값을 변경하는 매소드를 만

들어서 INSERT할 때 해주는 것인 줄 알았는데 찾아보니 상속에서는 생성자가 상속한 값
서 먼저 들어가고 그 후 상속되어있는 값으로 차례대로 들어가는 것을 알았다. 학생정보
라는 객체를 생성하면 대학 -> 컴퓨터정보공학 -> 학생정보 순으로 생성자가 호출되는
것을 공부한 결과 알아서 쉽게 생성할 수 있었다. 또한, get함수를 통해서 FIND에서 찾은
값을 출력하였다.

<문제 8번>

(1) 문제 및 설명

뱀과 개구리와 독수리와 호랑이 사이에 달리기 게임이다. Virtual을 써서 각 객체의 상속에 따라 구현하도록 만들었다. 객체는 동물간 계급 순위를 나타내는 Foodchain과 이동한 거리를 나타내는 Distance와 도착했는지를 확인하는 Arrival이라는 변수가 있다. 이것들을 확인하여 경주를 시작한다.

맨 처음에는 `sequence[4]`라는 변수에 동물들의 출발하는 순서를 입력한다. 그 후 순서대로 출발하고 모든 동물이 출발하면 동시에 동물들을 출발시키고 모든 동물이 도착할 때까지 계속 반복한다.

(2) 결과화면

```

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

```

경주 게임으로 처음에는 동물들이 어떤 순서로 이동하는지 체크하는 것이다.

- (1) 줄에는 아직 아무런 동물도 출발하지 않은 것을 볼 수 있다.
- (2) 줄에는 S F T E 순서로 S가 한 칸 출발하였다.
- (3) 줄에는 S가 한 칸 출발하고 F도 출발하였지만 S의 Foodchain이 더 높기 때문에 F가 죽어서 출력이 되지 않는 모습을 볼 수 있다.

(4)줄에는 그 다음 순서인 T가 출발하였고 S가 이동하는 곳에 T를 만나서 잡아 먹혀 S는 처음부터 출발하는 것을 볼 수 있다. 또한 F는 두 칸 움직였다.

(5)줄에는 그 다음 순서인 E도 출발하였고 E자리에 F가 갔는데 E에게 잡아먹혀 사라진 것을 볼 수 있다. S는 다시 출발하였다. T는 3칸 이동하였다.

(6)줄에는 S가 있는자리에 F가 갔기 때문에 잡아먹혀서 F는 출력이 안되고 T E는 각각 3칸, 4칸 출발한 것이 보인다.

(7)줄에는 F와 S가 출발하고 E와 T는 도착한 것으로 보인다.

(8)줄에는 F가 갈자리에 S가 있기 때문에 잡아먹혀 다시 출발한다.

(9)~(11)줄은 F와 S가 출발한다.

(12)줄에서 F는 S에게 잡아먹혀서 다시 출발한다.

(13) ~ (15) F와 S가 계속해서 달리고 (15)에서 S는 도착하였다.

(16)~(마지막) F도 꾸준히 가서 결국 도착하였다.

(3) 고찰

이 문제에서 어려웠던점은 가상함수에 대한 개념에 대해서 생소했던 것이 문제였다. 문제를 구현할 때 row들이 무슨 의미를 나타내는지 몰라서 파악하는데 시간이 걸렸다. Row는 각각의 출발한 순서였고 도착할 때까지 반복하는 것이었다. 그리고 이 동물들을 모든 동물이 한번이라도 출발후 동시에 출발시켜야 하는데 어떤 동물을 우선으로 출발시켜야할지도 고민을 했었다. 그 뒤로는 get함수와 set함수들을 이용하여 상속을 하는 Animal 객체의 private 변수들을 사용하였다.

<문제 9번>

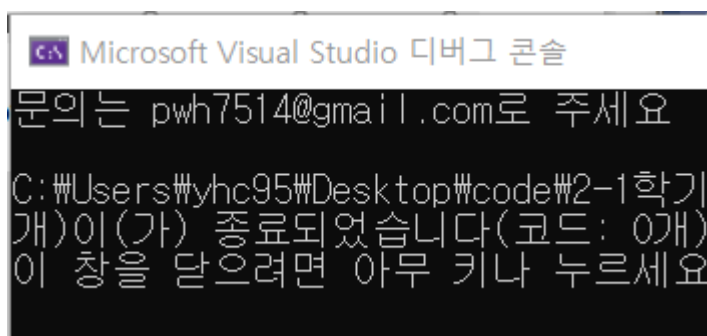
(1) 문제 및 설명

Mystring 객체를 구성하는데 operator overlaoding를 사용하여 +와 <<에 대한 기능을 구현한다.

Mystring에서 생성자를 오버로딩해서 char*형을 매개변수로 받는 생성자를 하나 만들어 my_str2에 저장한다. 또한 + operator를 오버로딩해서 두개의 char*를 합쳐서 lvalue에 저장한다.

마지막으로 cout의 <<를 오버 friend를 써서 구현을 하여 public으로 모두가 쓸 수 있도록 구현하고 아까 my_str2 + value를 저장했던 값을 출력한다.

(2) 결과화면



결과화면에 my_str2 + value를 저장한 my_str의 char*값이 +연산자 오버로딩에 제대로 들어가서 수행하고 값이 출력된 것을 볼 수 있다.

(3) 고찰

Operator 오버로딩을 사용하여 함수를 구현한다는 내용이 생소한 개념이라서 공부가 필요하였다. + 연산자 오버로딩을 사용할 때 값은 2개가 있는데 매개변수가 하나라서 어떻게 해야 할지 잘 몰랐다. 알고보니 this->가 가리키는 값이 +연산에서 왼쪽항목 매개변수로 받은 값이 오른쪽 항목이라는 것을 알고 함수를 구성하였다. 또한, 만약 +연산자에서 +앞 값에 아무것도 들어가지 않았을 때 /0이 저장되서 아무것도 안뜨는 것을 고치기 위해서 if문을 넣어주었다.