

과제 #4

HW1, HW2, HW3

과제 제출 마감: 10/31 24:00

조교 노인우, inwoo13@hanyang.ac.kr

조교 한중수, soohan@hanyang.ac.kr

Homework

- ◆ 과제 제출 마감: 10/31 24:00
- ◆ Gitlab repository에 “HW4” 폴더를 만든 후 진행
- ◆ 실습 서버 제출, 개인 PC 제출 중 편한 방법으로 제출
- ◆ 각 과제마다 HW4 폴더 내에 과제별 폴더를 만든 후 제출
- ◆ 채점 기준은 실습서버 환경에서 채점

Homework_01 – “블랙잭 (Black Jack)”

- ◆ 2장 이상의 카드를 입력받고, 블랙잭게임 결과를 출력하는 프로그램 작성
- ◆ 작성 규칙
 - 전체 카드의 합이 21이면 Black Jack
 - 10, J, Q, K는 모두 10, A는 1 또는 11이 될 수 있음
(21을 초과하지 않고, 21에 가깝도록 유리한 방향으로 사용)
 - Blackjack 함수는 전체 합이 21 이하이면 전체 합을, 21을 초과하면 0을, 잘못된 부분이 있을 때는 -1 리턴
 - Main 함수는 21미만은 전체 합, 21 이라면 “BlackJack”, 21을 초과하면 “Exceed” 출력
 - 자유롭게 로직 구현
- ◆ 파일명: **BlackJack** 폴더 내에 **BlackJack.cc**
- ◆ 입력: 입력 받을 카드의 수를 입력하고 (2 이상) 각 카드를 입력
 - 잘못된 카드를 입력했을 때 아무 출력 없이 프로그램 종료
 - 영문자는 대문자만 사용하고 소문자는 잘못 입력된 것으로 간주함
- ◆ 출력: 전체 합이 21 미만이면 전체 합을 출력
 - 21이라면 “BlackJack” 출력
 - 21을 초과하면 “Exceed” 출력

Homework_01 – “블랙잭 (Black Jack)”

◆ 입력 및 출력 예시

```
$ ./BlackJack
```

```
5 A 2 3 4 5
```

5 개의 카드 입력 (A, 2, 3, 4, 5)

```
15
```

A는 1로 간주하는 것이 유리하므로, $1 + 2 + 3 + 4 + 5$ 하여 15 출력

```
2 A J
```

```
BlackJack
```

```
3 A 5 6
```

```
12
```

```
6 5 6 K 10 9 A
```

```
Exceed
```

```
q
```

규칙 외의 입력으로 종료

```
$
```

Homework_02 – 댓글 관리 프로그램(Reply Administrator) F

- ◆ 지난번 만든 댓글 관리 프로그램 `reply_admin.h`와 `reply_admin.cc`를 기반으로 다음 사항을 만족하도록
 - `string* chats`를 `std::list<string> chats`로 변경한다.
 `string* chats`: 댓글 목록
 - `Int getChatCount()` 함수를 삭제한다.
 - `addChat, removeChat` 함수 구현을 `std::list`에 맞추어 간단하게 변경한다.
 함수 내에서 `vector`의 기능을 활용한다.
 - `removeChat(int *_indices, int _count)`를 `removeChat(list<int> _indices)`로 변경
 (변경 가능하지만, 필수 사항은 아니며, 굳이 변경하지 않아도 감점 없음)
 - 필요 시 interface (`reply_admin.h`)에 새로운 함수나 변수를 추가할 수 있다.
 - 이외의 사항은 지난 댓글 관리 프로그램 과제와 동일하다. 기본으로 추가되는 사항 모두 동일.
- ◆ 파일명: `reply_admin` 폴더 내에 다음과 같은 파일들 존재 (전 과제와 동일)
 (`reply_admin.h`, `reply_admin.cc`, `reply_admin_main.cc`)
- ◆ 입출력 및 자세한 기능은 지난 과제의 댓글 관리 프로그램과 동일

Homework_02 – 댓글 관리 프로그램(Reply Administrator) F

◆ 입력 및 출력 예시 (전과 동일)

```
$ ./reply_admin
Hello, World!
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 I think TAs look cool.
6 Hello, World!
#remove 5
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 Hello, World!
```

→ Hello, World! 입력

→ 미리 주어진 문자열들

→ 추가된 Hello, World! 문자열 출력

→ 5번을 제거하는 명령 입력

→ 기존 목록의 5번이 사라진 것을 확인

```
#remove 0,5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
C++ is so hard
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
Yesterday was too bad.
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
5 Yesterday was too bad.
#remove 4-5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
#quit
```

Homework_03 – “지뢰 찾기(minesweeper)”

- ◆ 지뢰(*)을 포함한 맵을 입력 받고, 근처에 존재한 지뢰의 개수를 출력하며 플레이 하는 게임 작성
minesweeper.h 코드:
<https://drive.google.com/file/d/0B3UgHoEW6yS8YmswLTFTZFFKcDg/view>
- ◆ 인터페이스를 유지하면서 필요한 경우 멤버 변수 및 함수를 추가한다.
 - 주어진 Minesweeper (minesweeper.h 내부에 존재) 클래스를 구현하고 이를 이용한다.
 - 지도 설정시 지뢰는 ‘*’, 빈칸은 ‘.’ 으로 입력 받는다.
 - 설정 후 입력 좌표의 지뢰를 toggle 할 수 있도록 한다. 지뢰가 변경되는 경우 주변의 지뢰 수를 다시 정확하게 계산하도록 한다.
- ◆ 파일명: **MineSweeper** 폴더 내에 다음과 같은 파일들 존재
(minesweeper.h, minesweeper.cc, minesweeper_main.cc)
- ◆ 입력: ‘.’로 시작하는 명령, 지도 입력, 지뢰 설정 위치, 종료 명령어
- ◆ 출력: 주변의 지뢰 수를 포함한 지도

Homework_03 – “지뢰 찾기(minesweeper)”

◆ 지뢰찾기 플레이 기능

- :set 으로 맵을 입력받은 후, :play 로 게임 시작 가능
- :play 이후 모든 맵이 ‘_’로 표시되며, :toggle 명령어는 사용 불가 (:set 명령 수행시 처음으로 돌아가며, :toggle 사용 가능)
- :play 이후에는 :touch ## 으로 맵을 터치 가능. 터치된 부분이 지뢰가 아닐 경우 숫자를 출력하며, 이미 터치된 경우에도 동일
- 터치된 부분이 지뢰일 경우, 죽었다는 알림과 함께 지뢰를 건드리기까지 터치한 횟수를 출력한다.
- :play 입력으로 처음부터 동일 맵으로 재시작 가능
- :set 입력으로 맵 재입력 가능

+ 원래 지뢰찾기에는 0을 터치하면 주변의 모든 숫자를 보여주지만, 이 과제에서 구현하지 않음

+ 지뢰를 제외한 모든 곳을 열면 게임이 종료되어야 하지만, 역시 이 과제에서는 구현하지 않음

Homework_03 – “지뢰 찾기(minesweeper)”

◆ 입력 및 출력 예시 (set 및 toggle 명령어)

```
./minesweeper
:set 5 4
*....
...*
.*...
.....
*1011
2211*
1*111
11100
:toggle 0 0
00011
1111*
1*111
11100
```

→ set 명령: 가로 5 세로 4 생성

→ 해당 맵 입력

→ 주변 지뢰 숫자 출력

→ (0, 0) 위치 반전 명령어 입력

→ 변경 반영하여 맵 출력

```
:toggle 3 2
00011
1122*
1*2*2
11211
:set 5 3
.....
.....
.....
00000
00000
00000
:quit
```

→ 종료 명령어 입력 시 종료

Homework_03 – “지뢰 찾기(minesweeper)”

◆ 입력 및 출력 예시 (play 명령어)

```
./minesweeper
```

```
:set 5 4
```

→ set 명령: 가로 5 세로 4 생성

```
*....  
...*  
*...  
....
```

→ 해당 맵 입력

```
*1011  
2211*  
1*111  
11100
```

→ 주변 지뢰 숫자 출력

```
:play
```

```
_____  
_____  
_____  
_____
```

→ play 후 모든 맵을 _로 출력

```
:touch 1 1
```

→ 1, 1 좌표를 터치

```
_____  
2____  
_____  
_____
```

→ 맵에서 해당 위치 숫자 출력

```
:touch 4 1
```

```
DEAD(2)
```

→ 지뢰 밟음 알림으로 DEAD 출력
괄호 안에 총 터치 횟수 출력

```
:quit
```

