

## 제출결과

## 13438. [2022 동계 대학생 양성과정] Battle Ship

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----

## 문제 내용

시간 : 10개 테스트케이스를 합쳐서 C++의 경우 1초 / Java의 경우 2초

메모리 : 힙, 정적 메모리 합쳐서 256MB 이내, 스택 메모리 3MB 이내

※ SW expert 아카데미의 문제를 무단 복제하는 것을 금지합니다.

- 코드배틀 비밀번호 및 게시된 문제의 타인 공유 / 무단 복제는 엄격히 금지됩니다.

- 지원언어 : C++ / JAVA

\* STL 라이브러리 사용 가능

- ① C 또는 C++로 답안을 작성하시는 응시자께서는 경쟁시스템에 제출 시, Language 에서 C++ 를 선택하신 후
- ② Main 과 User Code 부분으로 구성되어 있습니다.
  - A. Main : 수정할 수 없는 코드이며, 채점 시 비 정상적인 답안 검증 등 평가를 위한 로직이 추가 될 수
  - B. User Code : 실제 응시자가 작성해야 하는 코드이며, 제출 시에는 표준 입출력 함수가 포함되어 있으면
- ③ Local PC 에서 프로그래밍 시 유의 사항
  - A. 2개의 파일을 생성하셔야 합니다. ( main.cpp / solution.cpp 또는 Solution.java / UserSolution.java )
  - B. Main 부분의 코드를 main.cpp 또는 Solution.java 에 복사해서 사용하시기 바랍니다.
  - C. sample\_input.txt 를 사용하기 위해서는 Main 부분의 코드 내에  
표준 입력을 파일로 전환하는 코드 ( 주석처리 되어 있음 ) 의 주석을 풀어서 사용하시면 됩니다.
  - D. User Code 부분의 코드를 작성하신 후 서버에 제출하실 때,  
디버깅을 위한 표준 입출력 함수를 모두 삭제 또는 주석 처리해 주셔야 합니다.
- ④ 문제 내에 제약조건을 모두 명시하지 않으므로 주어지는 코드를 분석하셔야 합니다.
- ⑤ 코드는 개발 언어에 따라 상이할 수 있으므로, 작성할 언어를 기준으로 분석하셔야 합니다.

## [문제 설명]

아래는 Battleship 보드 게임에 대한 설명이다.

10 x 10 크기의 바다를 의미하는 보드가 있다.

한, 각각 1개의 항공모함(Carrier), 전함(Battleship), 순양함(Cruiser), 잠수함(Submarine), 구축함(Destroyer)의 모양을 한 말들이 있다.

## 제출결과

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----

R = 9



[Fig. 1]

보드의 각 칸은 좌표를 가진다.

세로 방향으로 R 값을 가지고 가로 방향으로 C 값을 가진다.

맨 왼쪽 위가 R = 0, C = 0 이고 맨 오른쪽 아래가 R = 9, C = 9 이다.

말들의 크기는 다르다.

항공모함(Carrier)의 말은 1 x 5의 크기를 가진다. 전함(Battleship)의 말은 1 x 4의 크기를 가진다.

순양함(Cruiser)의 말은 1 x 3의 크기를 가진다. 잠수함(Submarine)의 말은 1 x 3의 크기를 가진다.

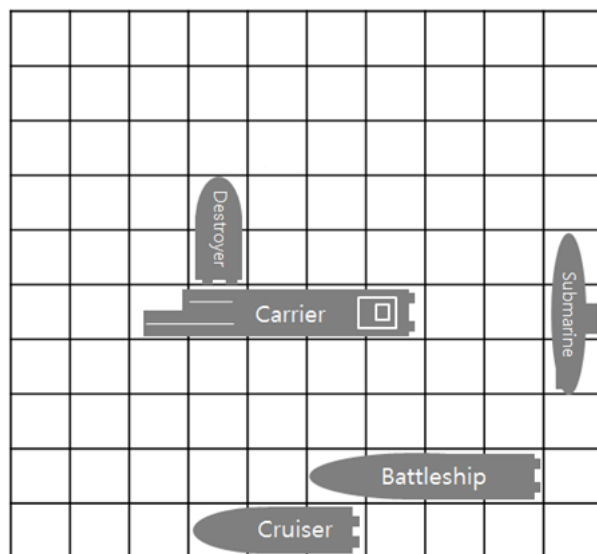
구축함(Destroyer)의 말은 1 x 2의 크기를 가진다.

게임이 진행되기 전, 플레이어가 모르게 5 개의 말들이 보드에 임의로 배치된다.

말은 가로 또는 세로 방향으로만 배치된다.

말은 보드 밖으로 나가거나 다른 말과 같은 칸을 공유하지 않는다. 말은 그 말의 크기 만큼 칸을 차지한다.

[Fig. 2]는 말들이 보드에 배치된 예이다.



[Fig. 2]

게임이 진행되면 배치된 말들의 위치는 변경되지 않는다.

게임 진행은 다음과 같이 된다.

제출결과

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----

이다.

그러면, 공격 결과를 알려준다.

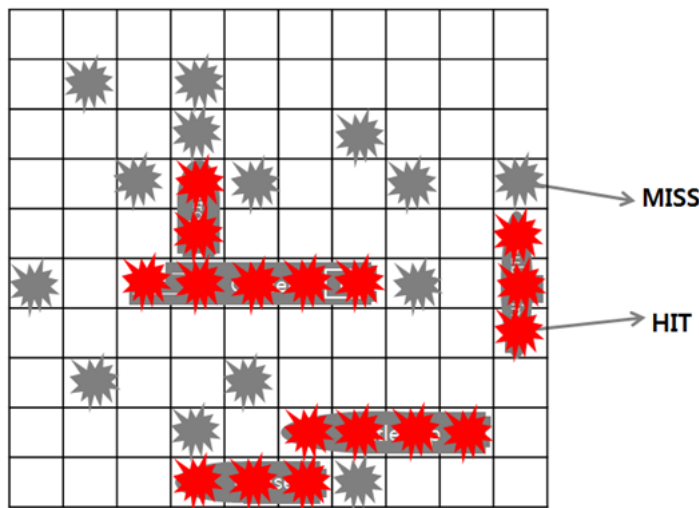
만약, 공격하는 칸에 말이 있는 경우 그 칸은 명중(HIT)이 되고 그 칸에 있는 말의 타입을 알려준다.

만약, 공격하는 칸에 말이 없는 경우 실패(MISS)라고 알려 준다.

이렇게 플레이어가 계속 공격하여 말이 배치된 칸을 모두 명중 시키면 게임이 끝난다.

[Fig. 3]은 모두 명중된 예이다.

붉은 색의 칸은 명중된 칸을 의미하고 회색의 칸은 실패한 칸을 의미한다.



[Fig. 3]

Main 부분은 Battleship 보드 게임을 시뮬레이션하는 코드이다.

각 테스트 케이스에서 배틀쉽 게임이 10회 진행된다.

되도록 적은 횟수로 공격하여 임의로 배치된 말들의 칸을 모두 격침하도록 API 함수들을 작성하라.

각 테스트 케이스의 모든 게임에서 fire() 함수의 호출 횟수가 주어진 limit 값 이하로 게임을 끝낼 때 해당 테스트 케이스의 점수를 얻는다.

동점자 간에는 모든 테스트 케이스에 대한 fire() 함수의 호출 총 횟수가 적을 수록 유리하다.

다음은 구현해야 할 API이다.

※ 아래 함수 signature는 C/C++에 대한 것으로 Java에 대해서는 제공되는 Solution.java와 UserSolution.java를 참고하라.

<b>void init(int limit)</b>
테스트 케이스에 대한 초기화 함수.
테스트 케이스의 맨 처음 1회 호출된다. 그 다음 play() 함수가 10회 호출된다. 각 play() 함수가 호출될 때 마다 게임은 새롭게 진행된다. 말들은 새롭게 임의로 배치된다.
limit는 play() 함수가 호출될 때 fire() 함수 호출 제한 횟수이다.
<b>Parameters</b>
limit : 제한 횟수 (70 ≤ limit ≤ 90)
<b>void play()</b>
게임이 진행되는 함수.
limit 횟수 이하로 fire() 함수를 호출하여 말이 배치된 모든 칸을 격침한다.

제출결과

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----

칸의 좌표가 세로 방향으로  $r$  번째이고 가로 방향으로  $c$  번째인 칸을 공격한다.

만약, 해당 칸에 말이 배치되어 명중(HIT)되는 경우 명중된 말의 타입이 반환된다.  
말의 타입은 다음과 같다.

항공모함(Carrier) = 1  
전함(Battleship) = 2  
순양함(Cruiser) = 3  
잠수함(Submarine) = 4  
구축함(Destroy) = 5

만약, 해당 칸에 배가 없어 실패(MISS)하는 경우 값 0 이 반환된다.

또한 이미 공격한 위치를 다시 공격하는 경우도 값 0 이 반환된다.

**Parameters**  
 $r$  : 공격하고 싶은 칸의 세로 방향 좌표 ( $0 \leq r \leq 9$ )  
 $c$  : 공격하고 싶은 칸의 가로 방향 좌표 ( $0 \leq c \leq 9$ )

**Return**  
공격하고 싶은 칸을 공격한 결과

- [제약사항]
- 1. 제한 횟수 limit는 70 이상 90 이하의 정수 값이다.
  - 2. 동점자 간에는 모든 테스트 케이스에 대한 fire() 함수의 호출 총 횟수가 적을 수록 유리하다.

[입출력]

입출력은 제공되는 Main 부분의 코드에서 처리하므로 User Code 부분의 코드에서는 별도로 입출력을 처리하지 않는다.

sample input에 대한 정답 출력 결과의 예시이며, 실제 출력 값과 동일하지 않을 수 있다.

```
#1 100 417
#2 100 473
#3 100 466
#4 100 434
#5 100 472
total score = 100, total callcount = 2262
```

입력

5  
37131113 90  
767 90  
6617 90  
211129102 90  
6891 90

sample\_input.txt

출력

#1 100 417  
#2 100 473  
#3 100 466  
#4 100 434  
#5 100 472  
total score = 100, total callcount = 2262

sample\_output.txt

제출결과

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----

Language C++14 (gcc-10.3)

Main

```
1 #ifndef _CRT_SECURE_NO_WARNINGS
2 #define _CRT_SECURE_NO_WARNINGS
3 #endif
4
5 #include <stdio.h>
6
7 extern void init(int limit);
8 extern void play();
9
10 static int board[10][10];
11
12 #define MISS      0
13 #define CARRIER  1
14 #define BATTLESHIP 2
15 #define CRUISER   3
16 #define SUBMARINE 4
17 #define DESTROYER 5
```

User Code

```
1 #define MISS      0
2 #define CARRIER  1
3 #define BATTLESHIP 2
4 #define CRUISER   3
5 #define SUBMARINE 4
6 #define DESTROYER 5
7
8 extern int fire(int r, int c);
9
10 void init(int limit)
11 {
12
13 }
14
15 void play()
16 {
```

소스 코드 초기화

TEST

Input 을 입력하고 Run을 선택하면 Output 결과를 확인할 수 있습니다. Input 값을  
넣지 않으면 기본 Input값이 적용되어 실행됩니다. Test는 채점을 하는 것이 아니며  
정답 여부를 알려주지 않습니다.

Input

Output

Input값을 입력해 주세요.

Run

Clear

제출결과

제출횟수 0 / 10

제출회차	제출시간	결과
------	------	----