

JAVA 월말평가 Project



| Background

- ✓ 객체지향 자바의 기본 문법
- ✓ 배열을 활용한 문제 해결

| Goal

- ✓ 주어진 문제를 정확히 분석하여 자바로 프로그래밍 할 수 있다.

| 환경 설정

1) 이클립스 시작시 워크스페이스는 **c:\WJavaTest** 로 변경합니다.

(만약, 이클립스 시작시 워크스페이스를 물어보는 창이 보이지 않고 바로 이클립스가 실행될 경우 File> Switch Workspace > Other 메뉴를 이용하여 워크스페이스를 변경합니다.)

2) 인코딩 (utf-8)

이클립스 메뉴-> Window -> Preferences -> General -> Workspace : UTF-8 로 변경

3) 메뉴 코드 제공 – 제공되는 소스 코드를 이클립스에서 import ->General-> Existing Projects into Workspace 선택 후 Select archive file을 선택하여 import 한다.

4) 프로젝트명 "Java_지역_반_이름" 을 올바르게 변경한다.

5) 제공되는 문제를 잘 읽고 분석해서 Test1.java~Test5.java를 완성하세요.

(제공되는 코드는 참고용이다. 처음부터 다시 작성해도 상관없다. 문제에서 원하는 출력 결과가 나오도록 작성하세요.)

6) 완성 후 프로젝트를 프로젝트명 "Java_서울_01_홍길동.zip"으로 압축하여 업로드 한다.


성실과 신뢰로 테스트에 볼 것 (부정 행위시 강력 조치 및 근거가 남음)

JAVA 월말평가 Project




| 문제1. 도형 만들기 (Test1.java)

키보드로 부터 1, A 중 하나의 문자를 입력 받아 아래와 같이 출력되는 프로그램을 작성 하세요.

[1 입력일 경우  모양의 삼각형을 아래와 같이 출력]

```
1
2 3
4 5 6
7 8 9 10
```

[A 입력일 경우  모양의 삼각형을 아래와 같이 출력]

```
      A
    B C D
  E F G H I
J K L M O P Q
R S T U V W X Y Z
```

| 문제2. 평균과 평균에 가까운값 구하기 (Test2.java)

다음과 같이 배열이 생성되어 있을 때 평균을 구하고, 평균과 값의 차이(절대값)가 가장 작은 값을 찾아서 출력하는 프로그램을 작성하세요.

구해진 평균값(정수값)과 평균에 가장 가까운 값을 출력하세요.

```
Int[] su= {34, 55, 27, 67, 45, 82, 68, 99, 77, 18}
```

출력

```
57 55
```

JAVA 월말평가 Project



| 문제3. 빈도수 구하기 (Test3.java)

키보드로 부터 아래와 같은 숫자들이 입력되었을 때 각 숫자들의 출현 빈도수를
체크하여 아래와 같이 출력하세요.

단, 숫자만 입력되며, 빈도수가 0인 숫자는 출력하지 않는다.

입력 예:

5329053995535987827332679340558347453272569584

출력 예:

0 : 2
2 : 5
3 : 8
4 : 4
5 : 10
6 : 2
7 : 5
8 : 4
9 : 6

입력 예:

1223334444555556666666

출력 예:

1 : 1
2 : 2
3 : 3
4 : 4
5 : 5
6 : 6



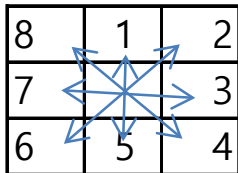
| 문제4. 여행 총 여행 거리 (Test4.java)

$N \times N$ 의 칸을 여행할 수 있는 지도(지도의 좌표는 (1,1) 부터 시작한다)가 있다.

각 칸에는 인접한 칸으로 이동할 수 있는 문이 있다. 현재 위치가 주어지고 여행을 가이드 하기 위해 이동 방향과 이동 칸수를 제공한다.

이동을 하다 보면 함정이 있다. 함정에 빠졌거나 이동 시 $N \times N$ 의 칸을 벗어나면 여행은 종료된다. 주어진 여행가이드 지시(방향과 이동칸수)대로 모두 수행 했을 때 여행자가 이동한 총 거리를 구하여 출력한다.

방향 값



[제한조건]

- N 은 자연수 3~20 의 값이다.
- 출발점 좌표에서 이동 지시의 방향으로 제시된 이동 칸 수만큼 반복적으로 이동(여행)한다.
- 이동 시 함정에 빠지거나, $N \times N$ 칸을 벗어 났을 경우 이후 이동 지시는 무시하고 현재까지의 총 거리를 출력한다.

[입력]

- 첫 줄에는 테스트 케이스 $T(1 \sim 100)$ 가 주어진다.
- 두 번째 줄에는 배열의 크기인 N 과 출발점의 좌표(행,열), 함정의 개수($1 \sim N$)가 주어진다.
- 세 번째 줄에는 함정의 개수만큼 함정의 좌표(행,열)가 주어진다..
- 네 번째 줄에는 방향 지시의 개수($0 \sim 50$)가 주어진다.
- 다섯 번째 줄에는 방향 지시의 개수만큼 (방향, 이동칸수)*방향지시개수 가 차례로 주어진다.
- 다음 테스트 케이스가 반복적으로 주어진다.

JAVA 월말평가 Project



[출력]

각 줄은 #Ti(테스트케이스 번호) 공백을 하나 둔 다음 방문횟수의 합을 출력한다.

[입력 예] Test4.txt 파일로 아래 내용 제공됨

```
3 // 테스트 케이스 수
8 5 3 4 // 배열크기, 출발점좌표, 함정의 개수
1 8 5 5 1 2 6 7 // 함정의 좌표
5 // 이동지시 개수
1 3 2 1 4 1 5 4 8 1 // 이동지시(방향, 이동칸수) * 5
3 1 1 1
1 2
3
4 2 1 1 2 1
8 5 3 4
1 8 5 5 1 2 6 7
5
1 3 2 1 6 2 8 1 5 2
```

[출력 예]

```
#1 7
#2 3
#3 9
```

JAVA 월말평가 Project



| 문제5. 오목 승자 체크하기 (Test5.java) (제한시간 1S, 메모리 32MB, 코드의 최적화 체크함)

SSAFY 친구들은 쉬는 시간에 오목 게임을 진행하였다.

오목은 가로줄과 세로줄에 1~19까지 번호가 붙어있는 오목판에 흰 바둑알과 검정 바둑알을 교대로 놓아서 먼저 정확히 5개의 같은 색 알이 연속적으로 일직선(직선, 대각선 포함)을 이루면 이기는 게임이다.

입력으로 주어진 바둑알의 정보를 체크하여 검정과 흰색 알 중 어떤 색이 이겼는지, 또는 승부가 결정되지 않았는지를 판단하는 프로그램을 작성하세요.

단, 검은색과 흰색이 동시에 이기거나 검은색 또는 흰색이 두 군데 이상에서 동시에 이기는 경우는 없다고 가정한다.

[입력]

검은 바둑알은 1, 흰 바둑알은 2, 알이 놓이지 않는 자리는 0으로 표시되며, 숫자는 한 칸씩 띄어서 입력된다.

[출력]

첫 줄에 승부가 결정되지 않았다면 0, 검은색이 이기면 1, 흰색이 이기면 2를 출력한다.

둘째 줄에는 이기게 된 5개의 바둑알 중 가장 왼쪽과 위쪽에 있는 바둑알의 가로줄 번호와 세로줄 번호를 순서대로 출력한다.

JAVA 월말평가 Project



[입력 예] Test5.txt 파일로 아래 내용 제공됨

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19 (가로줄번호 예시)
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	2	0	0	2	2	2	1	0	0	0	0	0	0	0	0	0	0
4	0	0	1	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(세로줄번호 예시)

[출력 예]

1
3 2