

[자바 전공 트랙] 1회차 월말평가 - JAVA



| 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) 코드 제공 – 제공되는 소스 코드를 압축을 풀지 않은 채로 이클립스에서 import ->General-> Existing Projects into Workspace 선택 후 Select archive file을 선택하여 import 한다.

4) 프로젝트명 “Java_지역_반_이름” 을 올바르게 변경한다. (ex: Java_서울_09_홍길동)

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

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

6) 완성 후 프로젝트를 프로젝트명 “Java_지역_반_이름.zip”으로 압축하여 업로드 한다.

(ex: Java_서울_09_홍길동.zip)

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

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시, 0점 처리 및 학사 기준에 의거 조치 실시 예정

[자바 전공반] 1회차 월말평가 - JAVA



| 문제1. 반복문 활용 (Test1.java) (20점)

다음과 같이 출력되는 프로그램을 반드시 반복문을 활용하여 작성 하세요.
(* 사이에 공백 출력은 없다.)

[출력 예]

```
*
* *
* * *
* * * *
* * * * *
```

| 문제2. 평균과 평균에 가까운 값 구하기 (Test2.java) (20점)

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

구해진 평균값(실수값)과 평균에 가장 가까운 값을 출력하세요.

(평균과 가까운 값이 여러 개인 경우는 아무거나 출력해도 상관없다.)

```
int[] su= { 45, 80, 68, 19, 34, 55, 27, 60, 27, 18 }
```

[출력 예]

```
43.3 45
```

[자바 전공반] 1회차 월말평가 - JAVA



| 문제3. 빈도수 구하기 (Test3.java) (20점)

아래와 같은 숫자들(0~9)로 구성된 문자열이 제공되었을 때 각 숫자들의 출현 빈도수를 체크하여 아래와 같이 출력하세요.

단, 숫자로 구성된 문자열만 제공되며, 빈도수가 0인 숫자는 출력하지 않는다.

문자열 예시1:

53290539955364534323455987827332679340558347453272569584

[출력 예]

0 : 2
2 : 6
3 : 11
4 : 7
5 : 12
6 : 3
7 : 5
8 : 4
9 : 6

문자열 예시2:

13334444555557777777

[출력 예]

1 : 1
3 : 3
4 : 4
5 : 5
7 : 7

[자바 전공반] 1회차 월말평가 - JAVA



| 문제4. 팀별 평균 구하기 (Test4.java) (20점)

S전자에서 신입생을 선발하기 위해 팀별 과제가 주어졌다.

각 팀별 평균 점수보다 높은 점수를 받는 학생들의 비율을 알기 위해 프로그램을 작성하려 한다.

다음 조건에 맞는 프로그램을 구현하세요.

[입력]

첫 줄에는 팀 수가 T개가 주어진다.

둘째줄에는 순서대로 팀원 수($1 \leq N \leq 1000$, N은 정수)와 각 팀원의 점수가 주어진다. 점수는 0보다 크거나 같고 100보다 작거나 같다.

[출력]

각 팀마다 팀의 평균을 넘는 학생들의 비율을 반올림하여 소수 셋째자리까지 출력한다.

#팀 (평균점수이상)비율%

[입력 예] Test4.txt 파일로 아래 내용 제공됨

```
5 // 팀수
5 50 50 70 80 100 // 팀원수 점수1 점수2 점수3 점수4 점수5
7 100 95 90 80 70 60 50
3 70 90 80
3 70 90 81
9 100 99 98 97 96 95 94 93 91
```

[출력 예]

```
#1 40.000%
#2 57.143%
#3 33.333%
#4 66.667%
#5 55.556%
```

[자바 전공반] 1회차 월말평가 - JAVA



| 문제5. 오목 승자 체크하기 (Test5.java) (20점)

(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.

하지만 여섯 알 이상이 연속적으로 놓인 경우에는 이긴 것이 아니다.

입력으로 주어진 바둑알의 정보를 체크하여 검정과 흰색 바둑알 중 어떤 색의 바둑알이 이겼는지, 또는 승부가 결정되지 않았는지를 판단하는 프로그램을 작성하세요.

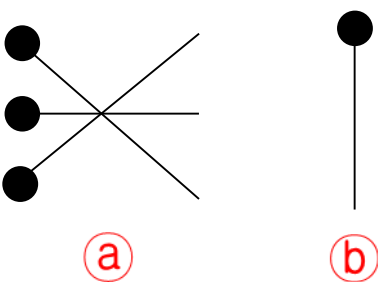
단, 검은색과 흰색이 동시에 이기거나 검은색 또는 흰색이 두 군데 이상에서 동시에 이기는 경우는 없다고 가정한다.

[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, 검은 색이 이기면 1, 흰 색이 이기면 2를 출력한다. 둘째 줄에는 이기게 된 5개의 바둑알 중 가장 왼쪽에 있는 바둑알의 가로줄 번호와, 세로줄 번호를 순서대로 출력한다. (아래 그림과 같이 대각선이나 가로로 5개의 바둑알이 연속 되었을 때(a) 가장 왼쪽 바둑알을 구하고, 세로로 5개의 바둑알이 연속 되었을 때(b)는 가장 위쪽의 바둑알을 구한다.)



[자바 전공반] 1회차 월말평가 - JAVA



[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
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

[출력 예]

1
3 2