

자바 프로그램 배열 테스트

1. 다음 중 배열에 대한 설명으로 틀린 것은?

- ① 자바에서 배열은 1차원씩만 관리한다.
- ② 자바에서 배열은 객체이다.
- ③ 자바의 배열은 해당 자료형의 기본 값으로 자동적으로 초기화된다.
- ④ 배열의 길이를 알고 싶을 때는 length라는 예약어를 사용한다.
- ⑤ 가변길이 배열은 행에 해당하는 부분만 초기화 하고 사용이 가능하다.

2. 아래와 같이 배열을 선언하였다. 다음 코드의 결과 값을 예측해 보세요.

```
int[][] number = { {3,4,5} , {2} , {1,3} };
```

- ① number[1].length ② number.length
③ number[2][0] ④ number[1][1]

3. 아래의 코드가 메모리에서 어떠한 형태로 잡히는지 표현해 보세요. (메모리 그릴 것)

```
int[][] test = new int[3][];
test[0] = new int[2];
test[1] = new int[3];
test[2] = new int[1];
```

※ 아래 코드의 결과를 예측해 보세요.

```
4.  int index = 1;
    int[] foo = new int[3];
    int bar = foo[index];
    int baz = bar[index];
    System.out.println(bar = " " + bar);
```

```
5. int index = 1;
   boolean[] test = new boolean[3];
   boolean foo = test[index];
   System.out.println(foo == " " + foo);
```

6. 이름(name), 국어(kor), 영어(eng), 수학(mat) 점수를 입력받아서 총점(tot), 평균(avg)을 구하는 프로그램을 작성하세요.

이때 평균은 소수점 3째 자리에서 반올림해서 소수저점 2째 자리까지 출력한다.

실행방법 : `java ExamEx05 홍길동 100 100 96`

```
*** 홍길동 성적표 ***
국어   영어   수학   총점   평균
100    100    95     296   98.67
```

7. 3과목의 점수를 입력받아서 합격이지 불합격인지 출력하는 프로그램을 작성하세요.
단, 3개의 과목이 전부 40이상이어야 하고 평균이 60이어야 "합격"을 출력하고
평균이 60이 넘지만 1개의 과목이라도 40 미만이면 "과락으로 불합격"을 출력하며
평균이 60미만이면 "불합격"을 출력한다.

실행예시 1 :

실행방법 : `java ExamEx06 75 80 100`

결과 = 합격

실행예시 2 :

실행방법 : `java ExamEx06 95 80 38`

결과 = 과락으로 불합격

실행예시 3 :

실행방법 : `java ExamEx06 32 40 22`

결과 = 불합격

8. 시작 값과 끝 값을 입력하여 시작부터 끝 값까지 출력하고 합을 구하는 프로그램을 작성하세요

실행예시 1 :

실행방법 : `java ExamEx08 2 20`

2 3 4 5 6 7 8 ... 18 19 20

합 = 209

실행예시 2 :

실행방법 : `java ExamEx08 20 2`

2 3 4 5 6 7 8 ... 18 19 20

합 = 209

9. 정수를 입력받아서 1 ~ 정수까지 약수를 구하는 프로그램을 작성하세요.
약수는 나누어서 떨어지는 수를 말합니다. (예 9의 약수는 1, 3, 9)

실행방법 : `java ExamEx09 52`

결과출력 : 1 2 4 13 26 52

10. 숫자를 입력받아서 2진수로 바꾸어주는 프로그램을 작성해 보세요.
단, 숫자는 0에서 255까지만 입력 합니다.

숫자 = 124

2진수 = 0111 1100

11. 프로그램 시작 시 2개의 매개변수를 받아서 처리하는 프로그램이 있다.
아래와 같이 실행될 수 있도록 작성하세요.

실행예시 1 :

실행방법 : `java ExamEx14 3 국어 영어 수학`

입력1 = 국어

입력2 = 영어

입력3 = 수학

실행예시2 :

실행방법 : `java ExamEx14 2 국어 영어 수학 과학`

입력한 숫자와 매개변수의 개수가 올바르지 않습니다.

Usage : `java ExamEx count param1 param2 ... paramCOUNT`

실행예시 3 :

실행방법 : `java ExamEx14 4`

매개변수의 개수는 반드시 2개 이상이어야 합니다.

Usage : `java ExamEx count param1 param2 ... paramCOUNT`

12. 아래와 같이 실행되는 야구게임을 만들어 봅시다.

단, 숫자는 무조건 3자리이며, 컴퓨터가 랜덤하게 만들어 냅니다.

(예를 들어 컴퓨터가 863이란 숫자를 만들어 낸 경우를 가정하겠습니다.)

게임을 실행 하시겠습니까? (y/n) : **y**

숫자를 생성했습니다.

당신이 생각하는 숫자 = **357**

1 구 : 1볼

당신이 생각하는 숫자 = **835**

2 구 : 1스트라이크 1볼

// 이런 식으로 계속해서 10구까지 진행합니다.

// 만약 10구 이내에 맞추지 못한다면 실패했다는 메시지를 출력하고 종료합니다.

당신이 생각하는 숫자 = **863**

정답입니다.^^

13. 아래와 같이 동작하는 가위바위보 게임을 만드세요.

money = 1000; // 게임이 시작되면 무조건 1000원의 잔액을 드립니다.

1. 주먹

2. 가위

3. 바위

0. 프로그램 종료

메뉴선택 : **1**

배팅금액 : **200**

결과를 보려면 Enter를 치세요^^ **Enter**

computer : 보자기, you : 주먹

You win !!! 잔액 : 1200원

// 여기서 다시 처음 메뉴로 이동합니다.

// 기타 세부적인 규칙은 자유롭게 적용하도록 합니다.

// 예를 들어 돈이 0원이하이면 종료한다. 등

14. 주민번호를 입력받아서 개인정보를 출력해 주는 프로그램을 아래와 같이 작성하세요.

이 름 : 홍길동
주민번호 : 1234567890123

올바른 주민번호일 경우 출력
홍길동님의 개인정보 분석결과
생년월일 : 1912년 34월 56일생
나 이 : 106세
성 별 : 남성
출생지역 : 경남출생

잘못된 주민번호일 경우 출력
잘못된 주민번호입니다.

참고사항 :

정보입력 방법
이 름 : 박성현 (br.readLine();)
주민번호 : 123456 1234567
int[] jumin = new jumin[13];
System.in.read(); //이용해서 입력
for(int i=0; i<jumin.length; i++)

주민번호확인 방법

1	2	3	4	5	6	-	1	2	3	4	5	6	7
* 2.0	3.0	4.0	5.0	6.0	7.0		8.0	9.0	2.0	3.0	4.0	5.0	

X + X + X + X + X + X + X + X + X + X + X + X + X = Hap

Temp = 11.0f * (int)(Hap/11.0f) + 11.0f - Hap
Total = Temp - 10.0f * (int)(Temp/10.0f)
올바른 주민번호 : Total == 7번자리...(jumin[jumin.length-1])
만약 다르다면 잘못된 주민번호...

정보분석 방법

1. 출생년도 확인 방법

주민번호 뒷부분의 첫 번째 자리가 아래와 같다면

9 0 : 1800년도

1 2 : 1900년도

3 4 : 2000년도

5 6 : 외국인(1,2 포함)

2. 출생지역 확인 방법

주민번호 뒷부분의 두 번째 자리는 시도를 의미한다.

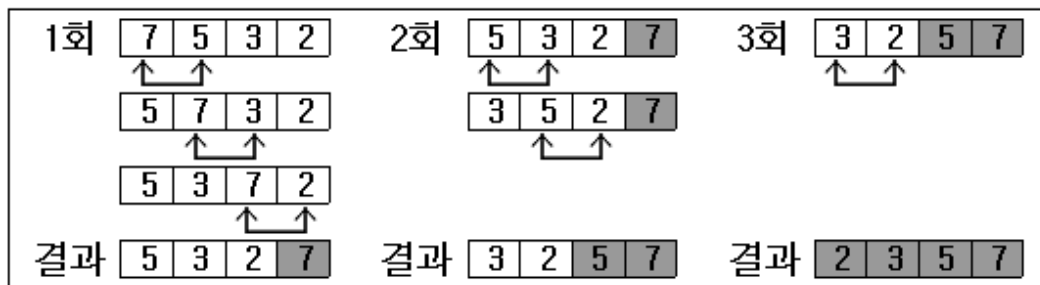
0 서울 1 경기 2 강원

3 충북 4 충남 5 전북

6 전남 7 경북 8 경남

9 제주

15. 다음은 버블정렬을 수행하는 알고리즘입니다.



위의 버블정렬 알고리즘을 참고하여 아래와 같이 진행되는 프로그램을 작성 하세요

숫자 개수 = 5

숫자 1 = 7

숫자 2 = 5

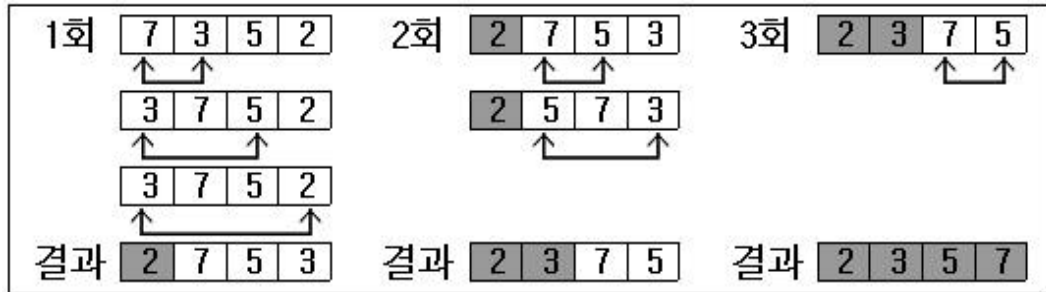
숫자 3 = 8

숫자 4 = 4

숫자 5 = 2

결과출력 : 2 4 5 7 8

16. 다음은 선택정렬 알고리즘입니다.



위의 선택정렬 알고리즘을 참고하여 아래와 같이 진행되는 프로그램을 작성 하세요

숫자 개수 = 5

숫자 1 = 7

숫자 2 = 5

숫자 3 = 8

숫자 4 = 4

숫자 5 = 2

결과출력 : 2 4 5 7 8

17. 아래의 코드는 마방진의 일부입니다.

프로그램이 정상적으로 수행될 수 있도록 코드를 추가해 주세요.

```
System.out.println("숫자(홀수만) : ");
int num = System.in.read() - 48;
System.in.read(); System.in.read();

int[][] mabang = new int[num][num]; //배열생성
int row = 0; int col = num / 2;
for(int su=1; su<= num*num; su++){
    mabang[row][col] = su;
    //여기에 코드를 추가합니다.
}

//결과출력
```

```

System.out.println();
for(int i=0; i<mabang.length; i++){
    for(int j=0; j<mabang[i].length; j++){
        System.out.print(mabang[i][j] + "Wt");
    }
    System.out.println();
}

```

18. 아래의 프로그램은 로또 추천 프로그램의 일부입니다.

여러 게임을 할 경우에도 서로 다른 게임이 완전히 일치하지 않는다는 것을 보증할 수 있도록 코드를 추가해서 프로그램을 완성해 주세요.

```

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
int lotto[][]; // 로또 저장할 배열공간 선언
System.out.print("Wt몇 게임을 하실꺼예요 ? ");
int number = Integer.parseInt(br.readLine());
lotto = new int[number][6]; //2차원 배열 초기화
for(int i=0; i<number; i++){//내가 선택 게임수 만큼
    for(int j=0; j<lotto[i].length; j++){
        lotto[i][j] = (int)(Math.random()*45) + 1;
        for(int k=0; k<j; k++){
            if(lotto[i][j] == lotto[i][k]){
                j = j-1;
                break;
            }
        }
    }
}
// 여기에 알맞은 코드를 추가해 주세요.
}

```

```

//결과출력
for(int i=0; i<number; i++){
    System.out.print("Wt" + (i+1) + "번째 추천번호 : ");
    for(int j=0; j<lotto[i].length; j++){
        if(lotto[i][j]>=10)
            System.out.print(" " + lotto[i][j]);
    }
}

```



```
        else
            System.out.print(" " + lotto[i][j]);
    }
    System.out.println();
}
```