

2021. 12. 28 인지연

1. 문제1 -1

```
import java.util.Scanner;

public class Ans1 {
    public static void main(String[] args) {
        /* 아래와 같은 등비 수열이 있다.
           1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
           사용자 입력을 통해 원하는 위치의 값을 뽑아내도록 프로그래밍 해보자!
           (1 ~ 32번째 혹은 31번째 항까지만 올바른 결과가 나올 것임) */
```

```
// 2^8 = 1byte = 256개
// -128 ~ -1 / 0 ~ 127
// 0을 포함하기 때문에 2^32승도 맨 끝이 홀수로 되어 있음
// 그래서 2^n을 표현한다고 할 때 실질적으로 전체 비트 - 1까지만 표현이 가능함
// 결국 2^31승을 표현하지 못하고 2^31 - 1이 최대값이 되는데
// 그래서 이 문제에서는 2^30승을 표현하기 위해 31번째까지가 최대가 됨
```

```
final int MAX = 31;
```

```
final int START_IDX = 0;
```

```
final int BASE = 2;
```

필요한 상수들 입력

Base가 2인 이유는 2를 제공해야하는 값이 등비수열이기 때문,

arr[0] 값은 1	arr[26] 값은 67108864
arr[1] 값은 2	arr[27] 값은 134217728
arr[2] 값은 4	arr[28] 값은 268435456
arr[3] 값은 8	arr[29] 값은 536870912
arr[4] 값은 16	arr[30] 값은 1073741824
arr[5] 값은 32	arr[31] 값은 2147483647
arr[6] 값은 64	

이 말의 뜻은 출력문을 보면 확인하기 편하다.
Arr[31], 즉 출력값 32번째 값은 홀수가 된다.
2의 제곱이니 무조건 뒷자리는 짝수가 되어야하는데.

이부분이 잘 이해가 안갑니다.
비트 -1 이 무슨 뜻일까요?

9. 문제3 -3

```
boolean isRealloc = true;

int lottoIdx = 0;
int allocCnt = 0;

for (int i = 0; i < SELECT; i++) {    // 총 5개 배치
    while (isRealloc) {
        lottoIdx = (int) (Math.random() * TOTAL);

        isRealloc = false;

        for (int j = 0; j < allocCnt; j++) {
            if (selectIdx[j] == lottoIdx) {
                System.out.println("중복 발생!");
                isRealloc = true;
                break;
            }
        }
    }
}

lottoBox[lottoIdx] = true;
selectIdx[allocCnt++] = lottoIdx;

// 이대로 가면 무엇을 놓치게 될까 ? 중복을 놓치게됨
// 그러므로 중복 발생 여부를 체크하는 루틴이 추가로 필요해짐!

System.out.println("lottoBox[" + lottoIdx + "] = " + lottoBox[lottoIdx]);

isRealloc = true;
}
```

만약 for i = 3이었고, if문이 true로 break가 되어서 for문이 종료되면,
for i 값은 증감되지 않고 다시 i = 3으로 시작되는건가요?
→ 첫번째 for문으로 가는게 아니고 while문으로 돌아가서 다시 랜덤뽑는 루트가 맞나요?

Lottebox[랜덤번호] = true → 로또박스배열에 숫자입력
블린형 배열은 어떤걸 하나요?
배열 안에 값을 입력하는것도 아닌것같은데 왜 쓰는건가요?
→ 선생님 답변 : 당첨이 됐는지 안됐는지 판별하기 위해 쓰인다.
하지만 쓰지 않아도 if문으로도 확인가능하지 않을까요? 아직도 왜 써야하는지 모르겠습니다.

18. 문제6 – 4 흐름

```
numArr[i] = testNum.divide(new BigInteger( String.valueOf(BASE.pow(i))))
```

여기까지는 이해가 갑니다.

Ex) 4321의 10^n (높은순)을 해서 배열 값에 4를 넣는건 알겠는데,

```
numArr[i] = testNum.divide(new BigInteger( String.valueOf(BASE.pow(i)))).mod(TEN).intValue();
```

Mod(ten)을 해서 어떻게 나머지 값을 도출해내는지...

10을 나눠서 남은값?

만약에 Ex) 4321의 10^n (높은순)을 나머지만 값이 321이여서, 여기서 다시 나누고.. 이러면 이해하는데 제가 이해한게 맞나요?