

# 12월 27일 과제 복습 중 의문 사항

## 문제3

중복을 확인하기 위해 10자리로 할당해 보았습니다.

```
for(int i = 0; i < SELECT; i++){ // 5개 당첨 자리 배치
    while (isRealloc){
        lottoIdx = (int)(Math.random()* TOTAL ); // 배열 자리 할당이므로 0 ~ 99까지 중 당첨 자리 배치

        isRealloc = false; // 자리를 막 할당 받았으므로 와일 구문을 거짓으로 지정

        for (int j =0; j < allocCnt; j++){ //5번 배치되어야 하므로 제대로 당첨 자리가 배치될 때마다 카운트 횟 수 1증가
            if (selectIdx[j] == lottoIdx){ //당첨 자리가 이 전에 할당된 당첨 자리면....
                System.out.println("당첨 자리를 재할당합니다!");
                isRealloc = true; //중복이 발생하였으므로 와일 구문을 재실행
                break;
            }
        }

        lottoBox[lottoIdx] = true; //당첨 자리를 참으로 설정
        selectIdx[allocCnt++] = lottoIdx; // 0부터 1씩 올라가며 할당된 당첨 자리 횟수가 5까지 올라간다.

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

        isRealloc = true;
    }
}
```

당첨 자리를 재할당하기 위해 while 구문을 참으로 바꿔 다시 돌려야 하는데 for 문 바로 위로 가야 할 것 같아서 For문 바로 위로 올려놓고 돌리니 제대로 나오지는 않네요. 밑에 오류도 발생하네요.

```
lottoBox[0] = true
당첨 자리를 재할당합니다!
lottoBox[6] = true
당첨 자리를 재할당합니다!
lottoBox[3] = true
lottoBox[1] = true
lottoBox[7] = true
```

바른 결과

```
lottoBox[2] = true    lottoBox[1] = true
lottoBox[7] = true    당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! lottoBox[6] = true
lottoBox[4] = true    당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! lottoBox[0] = true
lottoBox[5] = true    당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! 당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! lottoBox[7] = true
lottoBox[3] = true    당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! 당첨 자리를 재할당합니다!
당첨 자리를 재할당합니다! 당첨 자리를 재할당합니다!
```

잘못된 결과1

중복 발생

잘못된 결과2

```
selectIdx[allocCnt++] = lottoIdx; // 0부터 1씩 올라가며 할당된 당첨 자리 횟수가 5까지 올라간다.
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
at Hw3.main(Hw3.java:41)
```

```

for(int i = 0; i < SELECT; i++){ // 배열은 0부터 시작하므로 초기값을 0으로 설정, 5번의 루프를 돈다.
    while (isRealloc){ // isRealloc 이 참이나 거짓인 동안
        lottoIdx = (int)(Math.random()* TOTAL ); // 당첨 자리를 100개 중 랜덤으로 할당(0 ~ 99까지이므로 TOTAL - 1을 할 필요가 없음)

        isRealloc = false; // 당첨 자리를 막 할당 받았으므로 와일 구문을 거짓으로 지정

        for (int j = 0; j < allocCnt; j++){ // 5번 배치되어야 하므로 제대로 당첨 자리가 배치될 때마다 카운트 횟수 1증가
            if (selectIdx[j] == lottoIdx){ // 당첨 자리가 이 전에 할당된 당첨 자리면....
                System.out.println("당첨 자리를 재할당합니다!"); // 중복이므로 당첨 자리를 재할당
                isRealloc = true; // 중복이 발생하였으므로 와일 구문을 참으로 바꿔 재실행
                break;
            }
        }

        lottoBox[lottoIdx] = true; // 당첨 자리를 참으로 설정
        selectIdx[allocCnt++] = lottoIdx; // 0부터 1씩 올라가며 할당된 당첨 자리 횟수가 4까지 올라간다.

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

        isRealloc = true;
    }
}

```

중복 확인하며 반복

당첨 자리를 막 할당 후 false로 바뀐 isRealloc에 중복되지 않는 5번의 당첨 자리가 나올 때까지 isRealloc을 true로 바꿔 while구문을 반복하며 당첨 자리를 다시 생성하는 건 알겠습니다. 중복되지 않았으면 당첨 자리를 출력 후 false인 isRealloc을 다시 true로 바꿔주어 당첨 숫자를 생성하는 것도 알겠습니다.

하지만 마지막 5번까지 당첨 자리를 전부 할당하고 출력하며 종료되는데 맨 마지막에 isRealloc이 true로 되어 있어서 다시 while 구문으로 들어가는데 어떤 방식으로 종료되는지 궁금합니다. 아니면 5번 다 정상 출력되어서 밑에 isRealloc = true를 실행 안하고 바로 종료되는 건지 아니면 다시 실행 후 while 구문으로 들어가서 또 당첨자리를 생성하고 isRealloc = false로 만들어서 while 구문을 빠져 나오게 해서 종료되는 건가요?

## 문제6

```
int length = mantissa.intValue(); // 최종 결과에 + 1을 해야함 (초기 10 나눈건 계산 안함)
System.out.println("45678911234의 길이: " + (length + 1));

int[] numArr = new int[length + 1];

// 조건이 없음 - 언제 끝날지 모르니까 (feat by 무한 정수)
for (int i = length; i >= 0; i--) {
    // 실제 BigInteger의 경우엔 new BigInteger("문자열") 형식으로 숫자를 초기화함
    // 그러므로 String.valueOf()를 활용하여
    // valueOf() 내에 들어가는 숫자를 문자열로 변환하도록 한다.
    // 그리고 이 변환된 문자열 값을 new BigInteger() 내부에 배치하면 완료!
    numArr[i] = testNum.divide(
        new BigInteger(
            String.valueOf(
                //Math.pow(BASE, i)
                BASE.pow(i) //10의 i승
            )
        )
    ).mod(
        TEN
    ).intValue(); // BigInteger를 int로 변환함
```

10의 i승의 BigInteger 값

???

numArr[i] = testNum.divide(10의 i승의 BigInteger의 값)을 나눠서 몫이 나오는데 왜 또 10으로 나눠 나머지 값을 구하는지 모르겠습니다~