

2021. 12. 24 인지연

1. 23일 문제 1-1

이 문제는 while 사용이 더 좋다

Final 사용 권장

Boolean 이용하여 답 도출하기

While에 isChar을 입력하여 true가 되도록 만든다

랜덤문 도출하기

거기서 불리언 1,2 각각 조건 체크하기  
맞으면 false로 바꿔서 실행을 끝내고?  
Break로 loop문(while)문 나감.

만들어주는?

만약 if문에 해당되는 답이 아니라면  
"문자가 아님" 값을 하면서 값이 처음부터 다시 실행  
즉, if문에 해당 될 때까지 값을 계속 반복한다.

```
public class Answer1 {
    public static void main(String[] args) {
        final int MAX = 122;
        final int MIN = 65;
        int range = MAX - MIN + 1;
        boolean isChar = true;

        // 루프는 보편적으로 반복 작업을 하는 녀석들을 루프라고 함
        // while, for 요런애들이 루프(loop)
        while (isChar) {
            int rand = (int) (Math.random() * range + MIN);

            // 65 ~ 122 사이의 랜덤한 문자를 생성하도록 한다.
            // 여기서 소문자나 대문자가 아니라면 다시 생성하도록 프로그램을 만들어보자
            boolean condition1 = rand >= 65 && rand <= 90;
            boolean condition2 = rand >= 97 && rand <= 122;

            // if()는 루프가 아님
            if (condition1 || condition2) {
                System.out.printf("rand는 영문자 대소문자중 하나임: %c(%d)\n", rand, rand);
                isChar = false;
                // break를 걸면 loop(반복)을 바로 빠져나감
                break;
            }

            System.out.printf("문자가 아님: %c(%d)\n", rand, rand);
        }
    }
}
```

일단 true로 통과시킨 후 불론형으로 참 / 거짓을 맞춘다

여기는 참

여기는 거짓으로 만들어서 while문을 끝낼 수 있게 만들어주는?

```

boolean isChar = true;

// 루프는 보편적으로 반복 작업하는 녀석들을 루프라고 함
// while, for 요런애들이 루프(loop)
while (isChar) {
    int rand = (int) (Math.random() * range + MIN);

    // 65 ~ 122 사이의 랜덤한 문자를 생성하도록 한다.
    // 여기서 소문자나 대문자가 아니라면 다시 생성하도록 프로그램을 만들어보자
    boolean condition1 = rand >= 65 && rand <= 90;
    boolean condition2 = rand >= 97 && rand <= 122;

    // if()는 루프가 아님
    if (condition1 || condition2) {
        System.out.printf("rand는 영문자 대소문자중 하나임: %c(%d)\n", rand, rand);
        isChar = false;
        // break를 걸면 loop(반복)을 바로 빠져나감
        break;
    }

    System.out.printf("문자가 아님: %c(%d)\n", rand, rand);
}

```

## 2. 23일 문제 1-2

### 좀 더 자세히 살펴보기

문자가 아님: \ (92)  
 rand는 영문자 대소문자중 하나임: L(76)

출력문 1 : 처음엔 if문에 해당하지 않았고,  
 두번째 반복문에서 if문 조건 해당

rand는 영문자 대소문자중 하나임: 0(79)  
 문자가 아님: 0(79)

만약? Break문이 사용되지 않았다면?  
 If문을 달성해도 뒤에 문자가 아님이 출력된다.  
 While문이 끝난게 아니기 때문

1. While 문 밖에 boolean문을 만들어서 true로 만들어놓는다. (일단 true로 통과시킨 후 불문형으로 참 거짓을 맞춘다.)
  2. While 문 소가로에 true로 입력하여 while문 내부가 돌 수 있게 한다.
- While 문 안에서 일어나는 일
- 1) 랜덤 번호 하나 생성
  - 2) 그 안에서 Boolean 문 65~90 / 97~122 생성하기 (rand 가 while문의 지역변수이기 때문에 다른 곳에서 boolean문을 쓰면 오류가 난다.)
  - 3) If문 사용하여 내가 원하는 조건 사용하기.
  - 4) If문이 사용된 후 isChar = false로 변경하기 (변경하면 while문 소가로가 거짓으로 판명되어 while문이 끝나게 된다.)
  - 5) Break를 거는 이유 : loop를 종료하기 위해/ 이게 무슨뜻이냐  
 (만약 break를 걸지 않아서 while문 자체를 나가지 않는다면 if문 만족하여 영문자를 출력했어도 '문자가 아님' 이 단어를 출력하게 될것임)

### 3. 23일 문제 1-3

## 내가 푼 답

```
public class Num1 {  
    public static void main(String[] args) {  
        //1.65~122사이의 랜덤한 문자를 생성하도록 한다.  
        //여기서 소문자나 대문자가 아니라면 다시 생성하도록 프로그램을 만들기  
  
        final int MAX = 122;  
        final int MIN = 65;  
  
        int range = MAX-MIN +1;  
        int rand = (int) (Math.random()*range+MIN);  
  
        for(int i=1;i<=122;i++){  
            if(rand >=65 && rand <=90 || rand>=97 && rand<=122){  
                System.out.printf("대문자 혹은 소문자, 출력번호 : %c, %d",rand,rand);  
                break;  
            }  
        }  
    }  
}
```

이 문제는 랜덤 값 중에 일정 조건이 될 때까지 프로그램 진행하는것.  
그러니 for문을 통해서 증감을 굳이 할 필요가 없음.  
While 문은 참인지 거짓인지에 따라 결정되므로 이번 건은 while문이 나옴

## 4. 23일 문제 2

### 선생님 답

```
public class Answer2 {
    public static void main(String[] args) {
        int first = 1;
        int second = 1;
        int result = 0, i;

        final int START = 2;

        // 컴퓨터에서 배열의 시작이 0이기 때문에
        // 반복 시작을 0을 기준으로 맞춰주는 것이 좋습니다.
        // 그래서 START 2는 결국 3번째
        for (i = START; i < 20; i++) {
            result = first + second;
            first = second;
            second = result;

            // 전체 뿌리기
            // System.out.printf("%d번째 항 %d\n", i, result);
        }

        System.out.printf("%d번째 항 %d\n", i, result);
    }
}
```

### 내가 쓴 답

```
public class Num2 {
    public static void main(String[] args) {
        //2. 피보나치 수열의 20번째 항을 구하는 프로그램 만들기

        //1,1,2,3,5,8,13,21,34,55,89,144
        int num1 = 1;
        int num2 = 1;
        System.out.println("1번째 : " + num1);
        System.out.println("2번째 : " + num2);

        for (int i=3;i<=20;i++){
            int num3 = num1+num2; //num3 = 1+1 =2 num4 = 1+2 num5 = 2+3
            System.out.println(i+"번째 : " + num3);
            num1 = num2; //그 전꺼 (직전꺼 + 그 전꺼 합하기)
            num2 = num3; //직전꺼
            //num1 과 num2 값 변경 시켜서 for문 다시 돌리기
        }
    }
}
```

큰 차이는 없지만, final + 변수 사용에 더 유의하기

## 5. 23일 문제 3

```
public class Answer3 {
    public static void main(String[] args) {
        int first = 1;
        int second = 1;
        int third = 1;
        int result = 0, i;

        final int START = 3;
        final int END = 20;

        // 컴퓨터에서 배열의 시작이 0이기 때문에
        // 반복 시작을 0을 기준으로 맞춰주는 것이 좋습니다.
        // 그래서 START 2는 결국 3번째
        for (i = START; i < END; i++) {
            result = first + third;
            first = second;
            second = third;
            third = result;

            // 전체 뿌리기
            System.out.printf("%d번째 항 %d\n", i + 1, result);
        }

        System.out.printf("%d번째 항 %d\n", i, result);
    }
}
```

문제 : 1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129, ...  
이와 같은 숫자의 규칙을 찾아 25번째 항을 구하도록 프로그램 해보자!

$X = (x-1) + (x-3)$  기준으로 봐야함(순서기준)

1~3번째가 순환하면서 값을 만들기 때문에  
1~3의 변수를 만들어서 값을 하나씩 늘리면서  
순환해야한다.

## 6. 23일 문제 7

```
public class Answer7 {  
    public static void main(String[] args) {  
        final int START = 1;  
        final int END = 100;  
        final int DECISION = 4;  
        final int REMAIN = 0;  
  
        int sum = 0;  
  
        for (int i = START; i <= END; i++) {  
            if (i % DECISION == REMAIN) {  
                System.out.printf("%d의 배수 i = %d\n", DECISION, i);  
                sum += i;  
            }  
        }  
  
        System.out.printf("%d ~ %d까지 %d의 배수들의 합은 %d\n",  
            START, END, DECISION, sum);  
    }  
}
```

### 4 배수의 합 구하기

Final을 다 만들어서, 값을 다 대입해줬다.

Int sum은 for문 밖에 사용하므로,  
지역변수 선언

Final선언한 변수를 사용하면 숫자가 변해도  
반복문 내 숫자를 변경하지 않아도 된다.

Sum + 1 를 반복함으로 총 합이 구해진다.

출력문은 for문 밖에서 해줘야지  
For문에서 반복적으로 말하지 않는다.

## 7. 23일 문제 8

```
public class Answer8 {  
    public static void main(String[] args) {
```

```
        final int START = 1;  
        final int END = 100;  
        final int REMAIN = 0;
```

Final값 선언, 시작부터 끝  
최대~최소 모두 선언

```
        final int MAX = 10;  
        final int MIN = 2;
```

```
        int range = MAX - MIN + 1;  
        int decision = (int) (Math.random() * range + MIN);
```

랜덤을 생성해준다.

```
        for (int i = START; i <= END; i++) {  
            if (i % decision == REMAIN) {  
                System.out.printf("%d의 배수 i = %d\n", decision, i);  
            }  
        }
```

랜덤 생성한 값의 배수를 구하기 위해  
1~100까지의 값 중에 랜덤 배수를 구하기

```
    }
```

```
}
```

```
}
```

랜덤 값의 배수를 모두 구해라

## 8. 23일 문제 9 -2 (중요)

`boolean isRandomAllocCheck = false;` Boolean 값 지정 false으로 해서, 검사를 하지 않게 둔다.

`int decision = 0;`

For문 밖에서 변수 선언, for문을 나가도 해당 변수 사용 가능하겠끔

`int sum = 0;`

`for (int i = START; i <= END; i++) {` For문 1~100 순환 할 수 있도록 도와준다.

`while (!isRandomAllocCheck) {`

`decision = (int) (Math.random() * range + MIN);`

`isRandomAllocCheck = true;`

`}`

`if (i % decision == REMAIN) {`

`System.out.printf("%d의 배수 i = %d\n", decision, i);`

`isRandomAllocCheck = false;`

`sum += i;`

`}`

`}`

`System.out.println("현재까지 나타난 숫자들의 합 = " + sum);`

While 문을 사용하면서 방어문을 만들었다.

랜덤 값을 무기한으로 사용되지 않도록 불리언을 통해 통제

While 문 안으로 보자 랜덤 문 생성해놓고,  
isRandomAllocCheck를 true로 바꾸었다.

while()를 !~으로 해놨기 때문에 true -> false가 되어서  
For문이 돌아가도 while이 실행되지 않는다.

If문이 완성되어서 불문이 false가 되어야 while이 실행가능



## 9. 23일 문제 9 -2 (중요)

```
5의 배수 i = 5
3의 배수 i = 6
9의 배수 i = 9
4의 배수 i = 12
3의 배수 i = 15
10의 배수 i = 20
2의 배수 i = 22
6의 배수 i = 24
2의 배수 i = 26
8의 배수 i = 32
10의 배수 i = 40
7의 배수 i = 42
8의 배수 i = 48
5의 배수 i = 50
5의 배수 i = 55
10의 배수 i = 60
8의 배수 i = 64
7의 배수 i = 70
3의 배수 i = 72
2의 배수 i = 74
5의 배수 i = 75
2의 배수 i = 76
8의 배수 i = 80
5의 배수 i = 85
9의 배수 i = 90
```

### 9의 답

1. 처음 rand 값 : 5 / for문이 1부터 시작해서 if문의 5 나머지0의 값이 나올때까지 순환.  $i \% rand = 0$ 이 나온다면, blun 값이 false로 바뀐다.
2. For문 i값 6일때 while() !blun -> true가 된다. 즉 rand 숫자를 새로 갱신 가능.
3. Rand 값 : 3 / if문 시작  $6 \% 3 = 0$  값 출력 후 blun 문 false로 변환  
이후 반복 i가 100이 될때까지 반복

Sum은 처음부터  $i \% rand = 0$ 일때 값을 계속 더하고있다.

현재까지 나타난 숫자들의 합 = 1539

```
for (int i = START; i <= END; i++) {
    while (!isRandomAllocCheck) {
        decision = (int) (Math.random() * range + MIN);
        isRandomAllocCheck = true;
    }

    if (i % decision == REMAIN) {
        System.out.printf("%d의 배수 i = %d\n", decision, i);
        isRandomAllocCheck = false;

        sum += i;
    }
}
```

## 10. 23일 문제 9 -2 (중요)

```
5의 배수 i = 5
3의 배수 i = 6
9의 배수 i = 9
4의 배수 i = 12
3의 배수 i = 15
10의 배수 i = 20
2의 배수 i = 22
6의 배수 i = 24
2의 배수 i = 26
8의 배수 i = 32
10의 배수 i = 40
7의 배수 i = 42
8의 배수 i = 48
5의 배수 i = 50
5의 배수 i = 55
10의 배수 i = 60
8의 배수 i = 64
7의 배수 i = 70
3의 배수 i = 72
2의 배수 i = 74
5의 배수 i = 75
2의 배수 i = 76
8의 배수 i = 80
5의 배수 i = 85
9의 배수 i = 90
```

### 9의 답

1. 처음 rand 값 : 5 / for문이 1부터 시작해서 if문의 5 나머지0의 값이 나올때까지 순환.  $i \% \text{rand} = 0$ 이 나온다면, 블론 값이 false로 바뀐다.
2. For문 i값 6일때 while() !블론 -> true가 된다. 즉 rand 숫자를 새로 갱신 가능.
3. Rand 값 : 3 / if문 시작  $6 \% 3 = 0$  값 출력 후 블론 문 false로 변환  
이후 반복 i가 100이 될때까지 반복

Sum은 처음부터  $i \% \text{rand} = 0$ 일때 값을 계속 더하고있다.

현재까지 나타난 숫자들의 합 = 1539

```
for (int i = START; i <= END; i++) {
    while (!isRandomAllocCheck) {
        decision = (int) (Math.random() * range + MIN);
        isRandomAllocCheck = true;
    }

    if (i % decision == REMAIN) {
        System.out.printf("%d의 배수 i = %d\n", decision, i);
        isRandomAllocCheck = false;

        sum += i;
    }
}
```

## 11. 23일 문제 10 -1

### 선생님 코드

```
final int START = 1;
final int END = 100;

final int MAX = 10;
final int MIN = 2;

int range = MAX - MIN + 1;

int decision = 0;
int sum = 0;

for (int i = START; i <= END; i += decision) {
    // 내부에 배치되면 매 반복마다 더하는 값이 바뀜
    decision = (int) (Math.random() * range + MIN);

    System.out.printf("뽑은 난수 = %d, 현재 이동한 위치 = %d\n", decision, i);

    sum += i;
}

System.out.println("현재까지 나타난 숫자들의 합 = " + sum);
}
```

1 ~ 100까지의 숫자를 순회한다.

9번과 유사하게 2 ~ 10을 가지고 작업을 진행한다.

다만 이번에는 배수를 찾는게 아니라 랜덤한 숫자가 나온만큼만 이동하고 이동했을때 나온 숫자들의 합을 계산하도록 만들어보자!

#### 주의해야할 점

For 증감은  $i += \text{랜덤 값}$ , 즉 한번 순환할때마다  $i$ 값이 변한다.

For문 안에 랜덤 문 배치해서 매 순환마다 rand 값이 달라지도록 했다.

순환마다 위치가 달라져야하기 때문,

값이 순환될때마다  $i += \text{decision}$ 을 했기 때문에,

그냥  $\text{sum} += i$ 만 해도 rand이 이동한 위치를 알 수 있다.

## 12. 23일 문제 10 -1

```
final int START = 1;
final int END = 100;

final int MAX = 10;
final int MIN = 2;

int range = MAX - MIN + 1;

int decision = 0;
int sum = 0;

for (int i = START; i <= END; i += decision) {
    // 내부에 배치되면 매 반복마다 더하는 값이 바뀜
    decision = (int) (Math.random() * range + MIN);

    System.out.printf("뽑은 난수 = %d, 현재 이동한 위치 = %d\n", decision, i);

    sum += i;
}

System.out.println("현재까지 나타난 숫자들의 합 = " + sum);
}
```

## 선생님 답

```
뽑은 난수 = 7, 현재 이동한 위치 = 1
뽑은 난수 = 5, 현재 이동한 위치 = 8
뽑은 난수 = 5, 현재 이동한 위치 = 13
뽑은 난수 = 6, 현재 이동한 위치 = 18
뽑은 난수 = 6, 현재 이동한 위치 = 24
뽑은 난수 = 6, 현재 이동한 위치 = 30
뽑은 난수 = 2, 현재 이동한 위치 = 36
뽑은 난수 = 10, 현재 이동한 위치 = 38
뽑은 난수 = 8, 현재 이동한 위치 = 48
뽑은 난수 = 4, 현재 이동한 위치 = 56
뽑은 난수 = 2, 현재 이동한 위치 = 60
뽑은 난수 = 3, 현재 이동한 위치 = 62
뽑은 난수 = 7, 현재 이동한 위치 = 65
뽑은 난수 = 8, 현재 이동한 위치 = 72
뽑은 난수 = 7, 현재 이동한 위치 = 80
뽑은 난수 = 7, 현재 이동한 위치 = 87
뽑은 난수 = 6, 현재 이동한 위치 = 94
뽑은 난수 = 2, 현재 이동한 위치 = 100
현재까지 나타난 숫자들의 합 = 892
```

### 13. 23일 문제 10 -2

#### 내가 쓴 코드

```
final int MAX = 10;
final int MIN = 2;
int sum1 = 0;
int sum2 = 0;

for(int i= 1;i<=100;i++) {
    int rand =(int)(Math.random()*(MAX-MIN+1)+MIN);
    System.out.println(i+"번째 랜덤값 : "+rand);
    System.out.println(i+"번째 이동 : " +sum1+"\n");
    sum1 += rand;

    sum2 +=sum1;
}

System.out.println("총 합 : "+sum2);
```

내가 쓴 답은 for안에 rand를 만들긴했는데..  
랜덤값을 더해야한다고 생각해서 sum을 2개 만들었다.  
선생님처럼 i의 값을 아예 rand값을 더하면 굳이 그럴 필요는 없다.

답은 비슷하지만 불필요한 코드는 짜지 않도록 유의하기.

1번째 랜덤값 :7  
1번째 이동 : 0

2번째 랜덤값 :7  
2번째 이동 : 7

3번째 랜덤값 :6  
3번째 이동 : 14

4번째 랜덤값 :4  
4번째 이동 : 20

5번째 랜덤값 :5  
5번째 이동 : 24

6번째 랜덤값 :10  
6번째 이동 : 29

7번째 랜덤값 :10  
7번째 이동 : 39

8번째 랜덤값 :5  
8번째 이동 : 49

9번째 랜덤값 :7  
9번째 이동 : 54

93번째 랜덤값 :4  
93번째 이동 : 585

94번째 랜덤값 :2  
94번째 이동 : 589

95번째 랜덤값 :10  
95번째 이동 : 591

96번째 랜덤값 :5  
96번째 이동 : 601

97번째 랜덤값 :5  
97번째 이동 : 606

98번째 랜덤값 :3  
98번째 이동 : 611

99번째 랜덤값 :3  
99번째 이동 : 614

100번째 랜덤값 :3  
100번째 이동 : 617

총 합 : 32019

내가 쓴 답  
비슷하지만 아쉬운점

1~100을 순환한다고 했으니,  
100이 나왔으면 끝나야하는 것으로 답을 적었어야했다.  
하지만 나는 100번째 이동이 될때까지 계속 진행했다.  
다른 답을 적어냈던것.