

```

public class AllocTeamwork {
    public static void main(String[] args) {
        final int TEAMNUMBER = 2;
        System.out.println("지금부터 랜덤 팀 구성을 시작합니다.");

        String[] nameA = {
            "고동영", "장해솔", "류슬기", "박재민", "한다운",
            "최현정", "오진욱", "조진형", "이정현"
        };
        String[] nameB = {
            "이범진", "박소현", "탁성진", "노찬욱",
            "박기범", "하진주", "이승운", "최임식"
        };
    }
}

```

생성자 생성.

String [] 타입 'nameA' {~~}
String [] 타입 'nameB' {~~} 으로 할것이다.
TeamNumber= 2다.

```

private ArrayList<String> AteamArrayList;
private ArrayList<String> BteamArrayList;

private String[] AteamArr;
private String[] BteamArr;

private int AnumOfPerson;
private int numOfTeam;

private int BnumOfPerson;

public CreateTeam(String[] arrA, String[] arrB, final int TEAMNUMBER) {
    AteamArr = arrA; //----nameA 의 {}값
    BteamArr = arrB;

    AnumOfPerson = arrA.length; //--9
    BnumOfPerson = arrB.length;

    numOfTeam = TEAMNUMBER;

    AteamArrayList = new ArrayList<String>(); // ** 이걸 왜 굳이 public 안에 넣어되나?
    BteamArrayList = new ArrayList<String>();
}

```

```
// 단순히 고정된 사람들을 받는 작업
// CreateTeam ct = new CreateTeam(name);
CreateTeam ct = new CreateTeam(nameA, nameB, TEAMNUMBER);

ct.allocRandomTeam();
```

1) 클래스? 메소드? createTeam를 ct라고 지정하고 메소드 createTeam를 메모리에 (nameA, nameB, 2) 형식으로 값을 넣을 것이다.

3) (nameA, nameB, 2) 값을 allocRandomTeam 해라.

```
public CreateTeam(String[] arrA, String[] arrB, final int TEAMNUMBER) {
    AteamArr = arrA; //----nameA의 {}값
    BteamArr = arrB;

    AnumOfPerson = arrA.length; //--9
    BnumOfPerson = arrB.length;

    numOfTeam = TEAMNUMBER;

    AteamArrayList = new ArrayList<String>(); // ** 이걸 왜 굳이 public 안에 넣어야되나?
    BteamArrayList = new ArrayList<String>();
}
```

2) (nameA, nameB, 2) 값으로 ct CreateTeam으로 넣어라
 AteamArr = nameA
 BteamArr = nameB
 AnumOfPerson = nameA {}값의 길이 = 9
 BnumOfPerson = nameB {}값의 길이 = 8
 numOfTeam = 2

```

public void allocRandomTeam() {

    allocArrayList(
        // {}의값 상자? , 배열[0~8]의방들, 방들의 갯수 9
        AteamArrayList, AteamArr, AnumOfPerson
    );

    allocArrayList(
        BteamArrayList, BteamArr, BnumOfPerson
    );

}

```

allocRandomTeam() 메소드로
넘어가서 allocArrayList()를
진행해라.

(AteamArrayList, AteamArr,
AnumOfPerson)을 해라.

()의 추측값은

(AteamArr []의 값을 담는 상자,
배열 [] 0~8번까지의 방,
*여기서 [0]-고동영/[1]-장해솔~[8]-
이정현*

방들의 갯수=9) 라고 생각하는데
맞는지 모르겠습니다.

randNum 값이 0이 나왔다고 가정.

여기서 al는 AteamArrayList 이며 contains
(arr[0])이 있다면 isDup을 true로 만들고
continue해라.

**이때 continue는 다시 do 로 가는건지, add(arr[])를
거쳐서 while(isDup)가 true가 되고 while 조건이 '참'으로
다시 반복해라. 인지 모르겠습니다.**

else라면 al.add(arr[0]) 추가 하고 for문으로 반복으로
i=1이되면서 do를 반복.

```

public void allocArrayList(ArrayList<String> al, String[] arr, int loopNum) {
    boolean isDup = false;

    for(int i = 0; i < loopNum; i++) {
        // 우리는 A 배열, B 배열이 나눠진 케이스를 받으므로
        // 하나의 배열에서 모든 값을 처리하지 않는다.
        // 그러므로 start를 별도로 만들 필요가 없었다.
        // int randNum = (int)(Math.random() * loopNum) + start;
        do {
            int randNum = (int)(Math.random() * loopNum); // --0~8

            if(al.contains(arr[randNum])) {
                isDup = true;
                continue;
            } else {
                isDup = false;
            }

            al.add(arr[randNum]);
        } while(isDup);
    }
}

```

```
public ArrayList<String> getAteamArrayList() {
    return AteamArrayList;
}
```

allocRandomTeam() 로 get한
값들-AteamArrayList 라는 상자에 담겨있는 arr[]
값들을 AteamArrayList 로 리턴한다.

```
ct.printArrayList(ct.getAteamArrayList());
```

ct의 그 get값을 printArrayList 해라.

```
public void printArrayList(ArrayList<String> al) {
    String name;
    // get값들 순회할 수 있는 정보를 얻음
    Iterator e = al.iterator();

    int cnt = 1;
    int divNum;
    // 9/2=4.5
    int quot = al.size() / numOfTeam;
    //4

    // %-- 나머지값 so -1
    int remain = al.size() % numOfTeam;
    boolean needException =
        // a면 1이고 b면 0임 a면 트
        (remain > 0) ? true : false;

    int randValue = 0;
```

여기선 ct.getAteamArrayList 라고
생각합니다.

al = 여기선 get값이며 iterator 'e'로 칭한다.
iterator는 get값의 순회할수 있는 정보를
얻는것이다.

get한 값들-AteamArrayList 상자로 정보는
arr[0]~[8] 들이다.

size 랑 length랑은 다른것인가요?

quot 값은 4이며, remain 값은 1로 추측
needEx는 1>0으로 true
randvalue 0이다.

```
while(e.hasNext()) {  
    name = (String) e.next();  
    System.out.printf("%s ", name);
```

```
//...
```

```
if(needException) {  
    randValue = (int)(Math.random() * 2); //---0,1  
    needException = false;  
}
```

```
/*...*/
```

```
if((cnt % (quot + randValue)) == 0) {  
    System.out.println("");  
  
    if(cnt == 4) {  
        randValue = 1;  
    } else {  
        randValue = 0;  
    }  
  
    cnt = 0;  
}
```

```
cnt++;
```

```
System.out.println("");
```

while() 안에 e.hasNext()가 정확히
원지 모르겠습니다.

(String)타입으로 e.next -> iterator
'e'의 next()?)를 name로 한다.

4,5 로 할지 5,4로 할지 정하는
코딩으로 이해했습니다.



while 을 수행하고 출력해라.