

문제1. Prob1의 main() 메소드에는 정수형 배열이 초기화되어 있다. 이 배열을 매개변수로 받아서 배열의 내용을 홀수, 짝수인지 구분하여 홀수의 합과 짝수의 합을 아래와 같이 출력할 수 있도록 calc() 메서드 내부를 구현하세요. (main() 메서드 수정하면 안됨.)

[출력 결과는 다음과 같다]

```
( 배열 ia )
홀수의 합 : 26
짝수의 합 : 32
( 배열 ib )
홀수의 합 : 9
짝수의 합 : 6
```

문제2. JAVA Program 문제입니다.(Prob2.java이용)

문제 개요 : 1~30 사이의 10 개의 난수(랜덤한 정수)를 발생시켜 배열에 저장하여 리턴하는 generateRandom() 메소드와 매개변수로 받은 정수형 배열에서 최대값을 리턴하는 getMaxNum() 메소드를 각각 구현하시오.

<<실행결과>>

```
<< 생성된 난수 >>
5, 12, 3, 22, 17, 2, 17, 16, 7, 6,
=====
배열에서 가장 큰 숫자 : 22
```

<<주의사항>>

1. main() 메서드는 직접작성하세요.
2. 실행 결과는 예시일 뿐 발생된 난수는 매번 실행 때마다 달라집니다.
3. generateRandom() 메소드는 Math 클래스를 이용하여 구현할 수 있습니다.
4. getMaxNum() 메소드 구현 시, java.util.Arrays 클래스를 사용해서는 안되며, 반복문을 사용하여 로직을 구현해야 합니다.
5. getnerateRandom() 메소드를 구현했고 getMaxNum() 메소드 구현시, java.util.Arrays 클래스를 사용해서 구현한 경우 부분점수(B)를 부여합니다.
6. getnerateRandom() 메소드만 구현한 경우 부분점수(C)를 부여합니다.

문제3. JAVA Program 문제입니다.(Prob3.java)

문제 개요 : 매개변수로 받은 정수형 배열에서 중복되지 않은 숫자만 출력하는 printUniqueNumber() 메소드를 구현하시오.

<<메소드 시그니처>>

```
private static void printUniqueNumber (int[] array) {  
    // 구현하세요.  
}
```

<<실행결과>>

```
<< 중복되지 않은 숫자 >>  
12, 43, 21
```

<주의사항>>

1. 제공되는 main() 메서드는 수정하지 않고 실행해야합니다.
2. printUniqueNumber() 메소드의 시그니처는 수정할 수 없습니다.
3. 채점 시 정수형 배열의 내용을 변경해서 채점하므로 정확한 로직을 작성해야합니다.
4. 중복된 값이 출력된 경우는 오답으로 처리하며, 부분점수는 없습니다.
5. 컴파일이 안되거나 Exception 이 발생하는 경우는 무조건 **오답(0)으로 처리**됩니다.

문제5. Prob5 클래스의 `int getValue(int[] values, boolean isMax)` 메소드를 완성하십시오.
`getValue` 메소드는 `int` 타입의 배열을 입력받아 최대값 또는 최소값을 리턴합니다. `isMax` 값이 `true` 이면 최대값을 리턴하고, `false` 이면 최소값을 리턴합니다.

```
int getValue(int[] values, boolean isMax)
- isMax 값이 true 이면 int 타입의 배열값(values)에서 최대값을 리턴한다.
- isMax 값이 false 이면 int 타입의 배열값(values)에서 최소값을 리턴한다.
```

주의 사항 :

1) 주어진 메소드의 signature는 변경하지 않습니다.

문제6. Prob6 클래스는 입력 받은 숫자 배열 중 평균값에 제일 가까운 중간 값을 계산하는 클래스입니다. `int findMedian(int[] values)` 메소드를 완성하십시오.

```
int findMedian(int[] values)
- 중간 값이란 평균값에 제일 가까운 값을 말합니다.
- 중간 값이 여러 개 일 경우, 작은 수를 리턴 합니다.
```

<<처리 결과>>

```
input : [10, 4, 53, 63, 17, 37, 52, 16, 33, 65]
median : 33
=====
input : [32, 53, 52, 76, 15, 98, 76, 65, 36, 10]
median : 52
```

참고 및 주의 사항 :

- `Math.abs(int)` 메소드를 사용하여 평균값과의 차이의 절대값을 계산할 수 있습니다.
- 주어진 메소드의 signature는 변경하지 않습니다.

문제 7. 다음은 주어진 정수 배열로부터, 전체 정수의 합 대비 각 정수의 비율을 분석하는 CalculateRates 클래스 입니다. 입력값에 따라 다음과 같은 결과가 수행되는 calculate 메소드를 작성하시오.

```
void calculate(int[] nums)
- 주어진 정수들의 합을 구하고,
- 각 정수의 합을 이용하여 주어진 정수 배열에서 각 정수가 차지하는 비율을 구한 후,
- 1%당 1개의 *을 이용하여 비율을 표현합니다. *를 이용하여 비율을 표현한 후, 괄호 내에
  비율을 숫자로도 표현합니다.
- 단, 비율은 반올림하여 처리합니다.
```

<<처리 결과>>

입력된 정수 : {10, 30, 20, 80, 50} 일 경우

```
*** (5.0%)
***** (16.0%)
***** (11.0%)
***** (42.0%)
***** (26.0%)
```

문제8. 5행 5열 배열에 다음과 같이 숫자를 채우고 이를 출력하는 자바 프로그램을 작성하세요.
(Prob8.java)

[조건]

프로그램 내에서 2차원 배열을 하나만 사용한다.
출력시 표 형태는 출력하지 않으며 숫자 값만 출력한다.
숫자는 1부터 25까지 1씩 증가한다.

다음그림은 배열에 저장된 형태이다.

5	4	3	2	1
10	9	8	7	6
15	14	13	12	11
20	19	18	17	16
25	24	23	22	21

문제9. 다음과 같은 결과가 출력 되도록 Prob09.java 를 완성하시오.
 주의사항) main 메서드를 수정하지 않습니다.

실행결과)

원본 배열				
A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

수정된 배열				
*	B	C	D	E
F	*	H	I	J
K	L	*	N	O
P	Q	R	*	T
U	V	W	X	*

문제 10. 2차원 배열을 이용하여 두 행렬에 대한 차를 구할 수 있는 sub() 메소드와 결과를 출력하는 prn() 메소드를 Prob3 클래스 내에 구현하십시오.

Class : Prob3

Method 1 : public int[][] sub(int[][] a, int[][] b, int[][] c)

- 행렬의 차를 구하기 위한 메소드. [a - b = c]

배열 a	-	배열 b	=	배열 c
30 30 30 30		1 2 3 4		29 28 27 26
40 40 40 40	-	5 6 7 8	=	35 34 33 32
50 50 50 50		9 10 11 12		41 40 39 38

Method 2 : public void prn(int[][] c)

- 결과 배열 c 의 내용을 출력하기 위한 메소드.

- 아래 실행 결과 예시와 같이 각 행별로 줄바꿈을 하며, 각 열의 내용은 공백으로 구분.

실행 결과 예시)

```

29 28 27 26
35 34 33 32
41 40 39 38
    
```

문제11. 매개변수로 받은 두개의 int형 배열객체를 비교하여 배열의 내용이 같으면 true를 리턴하고, 다르면 false를 리턴하는 equalsArray() 메서드를 구현하시오.

<< ArrayCompare.java의 실행결과 >>

```
two array equals
=====
two array not-equals
```

문제 12. SortUtil 클래스의 int[] ascending(int[] inputNumbers) 메소드를 완성하시오.
ascending 메소드는 입력된 숫자들을 오름차순으로 정렬하여 리턴합니다.

```
int[] ascending(int[] inputNumbers)
- 숫자 배열의 순서를 오름차순으로 정렬하여 int 배열로 리턴하는 메소드를 구현한다.
- 숫자 배열 내에 동일한 숫자는 존재하지 않는 것으로 간주합니다.
```

<< 실행 예시 >>

입력 배열 : int[] inputNumbers = {7, 5, 2, 19, 34, 51, 32, 11, 67, 21};

리턴값 : {2, 5, 7, 11, 19, 21, 32, 34, 51, 67}

주의 사항 :

1. 테스트 목적으로 스스로 main 메소드를 구현할 수 있습니다.
main 메소드의 내용은 채점하지 않습니다.
2. 주어진 메소드의 signature는 변경하지 않습니다.
주어진 signature가 다를 경우 틀린 것으로 간주합니다.

문제 13. 입력값에 따라 다음과 같은 출력이 나오도록 문자열의 순서를 뒤집는 메소드를 정의합니다.

```
char[] reverseString(char[] s, int start, int end)
```

- 문자배열의 순서를 뒤집어서 char 배열로 리턴하는 메소드를 구현한다
- 문자를 가르키는 인덱스(start, end)는 0부터 시작한다.

<< 실행 예시 >>

입력 배열 : char[] s = {'J', 'a', 'v', 'a', 'P', 'r', 'o', 'g', 'r', 'a', 'm'};

- start = 0, end = 3 인 경우 => **ava**JProgram 리턴
- start = 4, end = 7 인 경우 => Javagor**P**ram 리턴
- start = 8, end = 10 인 경우 => JavaProg**mar** 리턴
- start = 0, end = 10 인 경우 => **margorPava**J 리턴

주의 사항 :

3. 테스트 목적으로 스스로 main을 구현할 수 있습니다.
main 메소드의 내용은 채점하지 않습니다.
4. 주어진 메소드의 signature는 변경하지 않습니다.
주어진 signature가 다를 경우 틀린 것으로 간주합니다.