

평가 시험

참고 사항

- 문제에 관해 부족한 설명은 테스트 코드를 참조한다.
- 생성형 AI를 이용해 작성된 코드는 0점 처리된다.

1. System.arraycopy와 같이 배열 복사 함수를 작성하고, 테스트 코드로 확인하라.(10)

요구 사항

- 입력에 대한 유효성 검사를 한다.
 - 입력이 null인 경우, NullPointerException() 발생시킨다.
 - 인덱스 오류로 인해 정상적인 복사가 수행되지 못할 경우, ArrayIndexOutOfBoundsException() 발생시킨다.
- 동일한 배열 내에서도 복제가 가능하다.

2. 임의의 진수를 받아 임의의 진수로 변환하는 함수를 작성하고, 테스트 코드로 확인하라.(25)

요구 사항

- 입력과 출력은 문자열로 한다.
- 임의의 진수는 2, 8, 10, 16 진수로 제한한다.
- 함수 형식은 테스트 코드를 참고한다.
- 각각의 진수는 고유의 형식을 갖는다.
 - 2진수 -[^]-? (0|1[0|1]*)\$
 - 8진수 -[^]-? (0|[1-7][0-7]*)\$
 - 10진수 -[^]-? (0|[1-9][0-9]*)\$
 - 16진수 -[^]-? (0|[1-9A-Fa-f][0-9A-Fa-f]*)\$
- 입력 형식에 지정된 형식과 맞지 않을 경우, exception을 발생시킨다.
 - 숫자 형식이 잘못된 경우, NumberFormatException()
 - 지원하지 않는 진수의 경우, NotSupportedException()

3. 소수를 찾는 함수를 작성하고 이를 이용해 리만 가설을 위한 함수를 작성하라.(15)

요구 사항

- 리만 가설은 연속된 제곱 쌍 사이에는 적어도 하나의 소수가 존재한다.
- 두개의 정수 n, m을 인수로 받아, n 제곱과 m 제곱 쌍 사이에 존재하는 소수를 찾는다.

○ n, m은 양의 정수만 허용하며, 음수가 입력될 경우, IllegalArgumentException() 발생

4-1. 입력되는 계산식을 abstract syntax tree를 이용해 구성하고, 이를 이용해 계산하라. (35)

요구 사항

- 4칙연산자(괄호 미포함)만 지원하는 계산기를 작성한다.
- 입력은 공백으로 구분되는 문자열로 하며, 연산자와 비연산자를 요소로 입력한다.

```
// 1 + 2 * 3 / 4 - 5  
String expression = "1 + 2 * 3 / 4 - 5";
```

- Abstract Syntax Tree작성하여 계산한다.

```
Node generateAST(String expression)
```

- AST를 문자열로 아래와 같은 형식으로 출력한다.

```
System.out.println(generateAST(expression));
```

```
((1),+,(((2),*(3)),/(4))),-(5))
```

- 수식이 잘못된 경우, InvlaidExpressionException()을 발생시킨다.

4-2. AST를 이용해 연산하는 evaluation() 함수를 작성하라. (15)

- 함수 형식은 아래와 같다.

```
int evaluation(Node ast)
```

- recursion을 이용해 구현한다.