HW4 - Factoring

(정수론)

18013195

정보보호학과

이풍원

1) factoring_simple(n): it returns the list of prime factors of n by using the simple division method.

```
1   from math import*
2   def factoring_simple(n):
3       FactorList=[]
4       i=2
5       while int(n)!=1:
6           if n%i==0:
7               n/=i
8               FactorList.append(i)
9           else: i+=1
10      return FactorList
```

3 행: factorList 는 List 형으로 소인수들을 담을 곳이다.

4 행: 2 부터 나누기위해 i 를 2 로 초기화한다.

5 행: while 문 종료조건은 n!=1 로 몫이 1 이면 나눌 것이 없기 때문이다.

6~8 행: n 에 i 를 나눠 나머지값이 0 일 때 실행 하고 n 값을 i 나눈값으로 저장하고 FacotrList 에 i 값을 추가한다.

9 행: i 로 나누기를 못 하면 i 를 1 증가 시킨다.

10 행: 소인수들을 반환한다.

2) factoring_fermat(n): it returns the two prime factors p and q of the given odd integer n by using the fermat factorization method where n = p*q.

```
12  def factoring_fermat(n):
13      a=int(sqrt(n)+1)
14
15      while True:
16          b=int(sqrt(a**2 - n))
17          if sqrt(a**2-n)==b:
18              return a-b,a+b
19          else: a+=1
```

13 행: a 에 n 제곱근을 한 값에 +1 하여 소수점을 버린 값을 저장한다.

16 행: b 값에 a^2- n 의 제곱근을 한 값에 소수점을 버린 값을 저장한다.

17~18 행: 만약 a^2-b 제곱근과 b 값이 같으면 a^2-b 제곱근이 완전 제곱이기 때문에 a-b,a+b 를 반환한다.

19 행: 위의 조건을 만족 못할 때 a 에 1 을 더한다

3) main 및 실행 결과

```
22  print(factoring_simple (11))
23  print(factoring_simple (100))
24  print(factoring_simple (12345))
25  print(factoring_simple (1000001))
26  print(factoring_simple (2 ** 16))
27
28  print(factoring_fermat (15))
29  print(factoring_fermat (119))
30  print(factoring_fermat (187))
31  print(factoring_fermat (2987))
32  print(factoring_fermat (6750311))
33
```

문제 출력 디버그 콘솔 터미널 2: Python

```
[11]
[2, 2, 5, 5]
[3, 5, 823]
[101, 9901]
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
(1, 11)
(7, 17)
(11, 17)
(29, 103)
(103, 65537)
PS C:\Users\vnddn\OneDrive\바탕 화면\js>
```