

알고리즘 3주차 과제

201701969 강인한

```
if name == 'main':
    num_str = input()
    nums = list()

    for i in range(len(num_str)):
        per = list(permutations(num_str, i+1))
        sum = 0
        for j in range(len(per)):
            nums.append(make_num(per[j]))

    ansList = deque()

    for i in range(len(nums)):
        if find_prime(nums[i]) == True:
            if nums[i] not in ansList: ansList.append(nums[i])
        else: None

    print(sorted(ansList))
    print(len(ansList))
```

main코드는 다음과 같다. 코드에서 위의 for문과 아래의 for문을 나눠서 설명하자면, 먼저 위의 반복문은 입력받은 문자열을 순열을 이용해 길이가 0인 것부터 문자열의 길이인 n까지 가능한 모든 순열의 리스트를 만든다. 그 후 모든 순열의 담긴 리스트를 숫자로 만들어주는 함수인 make_num을 이용해 숫자로 바꾼다.

```
def make_num(lst):
    s = 0
    for i in range(len(lst)):
        s = s + int(lst[len(lst)-i-1]) * (10**i)
    return s
```

```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/user/PycharmProjects/
127
첫 반복문을 통해 만들어진 리스트는 : [1, 2, 7, 12, 17, 21, 27, 71, 72, 127, 172, 217, 271, 712, 721]
```

위의 반복문까지 작성하고 실행을 돌려 127을 입력하면 위와 같은 결과가 나온다. 즉, 127을 이용해 만들 수 있는 모든 경우의 숫자들이 들어가 있는 리스트가 만들어진다.

두 번째 반복문은 리스트에 담긴 숫자가 소수인지를 판별하는 함수, find_prime을 통해 소수면 새로운 리스트에 추가하고 소수가 아니라면 그냥 넘어가는 리스트를 만드는 작업을 한다.

```
def find_prime(n):  
    if n==1: return False  
    for i in range(2,n):  
        if (n % i == 0):  
            return False  
    return True
```

```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/user/PycharmProjects/pythonP  
127  
첫 반복문을 통해 만들어진 리스트는 : [1, 2, 7, 12, 17, 21, 27, 71, 72, 127, 172, 217, 271, 712, 721]  
두번째 반복문을 통해 만들어진 리스트는 : [2, 7, 17, 71, 127, 271]
```

두 번째 반복문을 통해 소수인 숫자들만 존재하는 리스트가 만들어진다.

결과화면 : 두 번째 반복문을 통해 만들어진 리스트의 길이인 len()을 씌우면 총 개수를 알 수 있다.

```
findPrime X  
C:\Users\user\PycharmProjects\pythonP  
127  
6  
Process finished with exit code 0
```

```
findPrime X  
C:\Users\user\PycharmProjects\pythonP  
11527  
40  
Process finished with exit code 0  
|
```