

17103 골드바흐 파티션

2021년 1월 20일 수요일 오후 1:12

문제

- 골드바흐의 추측: 2보다 큰 짝수는 두 소수의 합으로 나타낼 수 있다.

짝수 N을 두 소수의 합으로 나타내는 표현을 골드바흐 파티션이라고 한다. 짝수 N이 주어졌을 때, 골드바흐 파티션의 개수를 구해보자. 두 소수의 순서만 다른 것은 같은 파티션이다.

입력

첫째 줄에 테스트 케이스의 개수 T ($1 \leq T \leq 100$)가 주어진다. 각 테스트 케이스는 한 줄로 이루어져 있고, 정수 N은 짝수이고, $2 < N \leq 1,000,000$ 을 만족한다.

출력

각각의 테스트 케이스마다 골드바흐 파티션의 수를 출력한다.

예제 입력 1 복사

```
5
6 → 33 check → 3개
8 → 35 check → 4개
10 → 37 45 check → 5개
12 → 57 check → 6개
100
```

① 2가 배열 → 에라토스테네스의 체 → $O(n \log n)$
② 2가 배열로 돌까지 돌아 → $O(n^2)$
 $O(n \log n) + O(n) = O(n \log n)$

6-2=4
6-3=3
6-5=1

예제 출력 1 복사

```
1
1
2
1
6
```

소스코드

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

# 에라토스테네스의 체
def eratos(max):
    a = [False, False] + [True] * (max-1)
    for i in range(2, max+1):
        if a[i]:
            primes.append(i)
            for j in range(2*i, max+1, i):
                a[j] = False
    return a

# 골드바흐 파티션을 구하는 함수
def goldbach(arr, primes):
    # arr의 개수만큼 반복
    for num in arr:
        result = 0
        # primes내의 수가 arr/2보다 클 경우 result 그대로 출력
        for i in range((num//2)+1):
            if primes[i] and primes[num-i]:
                result += 1
        print(result)

n = int(input())
max = 0
primes = []
arr = []
for i in range(n):
    k = int(input())
    if max < k:
        max = k
    arr.append(k)
primes = eratos(max)
goldbach(arr, primes)
```

☆
 $i-j+i = 2i$ 가
0이므로
2가까이 있는
소수 2개
2가까이 있는
소수 2개

54 12 9 7 5
54 12 9 7 5
54 12 9 7 5