

백준 9020 골드바흐의 추측

202020988 조아영

# 001 문제 소개

# 002 아이디어

# 003 코드 설명

## # 문제 소개

### 골드바흐의 추측

2보다 큰 모든 짝수는 두 소수의 합으로 나타낼 수 있다.  
이러한수를 골드바흐의 수라고 하고, 짝수를 두 소수로 나타내는 표현을 골드바흐 파티션이라고 한다.

예시)

$$4 = 2 + 2 / 6 = 3 + 3 / \dots / 14 = 3 + 11 = 7 + 7 / \dots$$

입력) 테스트 케이스 T, 짝수  $n$  ( $4 \leq n \leq 10,000$ )

출력)  $n$ 의 골드바흐 파티션

- 골드 바흐 파티션이 여러가지 -> 두 소수의 차이가 가장 작은 것 출력
- 출력하는 소수는 작은 것부터 출력

## # 아이디어

입력받은  $n$ 까지의 소수를 배열에 저장

골드바흐 파티션이 두 개 이상일 경우 두 소수의 차가 작은 것 출력  
-> 처음부터 소수가 저장된 배열의 길이 이용,  
    중간 인덱스 부분부터 골드바흐 파티션 찾음

Index Error / 시간초과



$n$ 의 최댓값인 10000까지의 소수를 배열에 저장 (에라토스테네스의 체 이용)

골드바흐 파티션이 두 개 이상일 경우 두 소수의 차가 작은 것 출력  
-> 소수가 저장된 배열의 0번째 인덱스부터 하나씩 증가하며 비교,  
    최댓값 저장 / 배열의 중간 넘어가면 종료

# # 코드 설명

```
import sys
```

```
prime = [True for i in range(10001)]
```

```
prime[0] = False
```

```
prime[1] = False
```

소수인지 True / False

저장할 배열 초기화

```
for i in range(2, int(10001**(1/2))+1):
```

 에라토스테네스의 체 알고리즘

```
    if(prime[i] == 0):
```

```
        continue
```

```
    for j in range(i*2, 10001, i):
```

```
        prime[j] = False
```

```
prime_num = []
```

96%

```
for i in range(10001):
```

```
    if(prime[i] == True):
```

```
        prime_num.append(i)
```

소수 배열에 저장

# # 코드 설명

```
T = int(sys.stdin.readline())           테스트 케이스 입력

for _ in range(T):
    n = int(sys.stdin.readline())       골드바흐의 수 n 입력
    index = 0
    temp = 0

    배열의 중간 넘어가면 종료
    while (n - prime_num[index]) >= prime_num[index]:
        if prime[n - prime_num[index]] == True:
            if(temp < prime_num[index]):  최댓값 저장
                temp = prime_num[index]  ( 두 소수의 차이가 가장
            index += 1                  작은 것 출력하기 위해 )

    print(temp, n-temp)                 골드바흐 파티션 출력
```

감사합니다