



Part 01

알고리즘

1. 알고리즘 이해

2. 기본 알고리즘 – 수열

3. 기본 알고리즘 – 수학

- Count 알고리즘, 최댓값과 최소값, 합계와 평균, 소수 판별, 소인수 분해, 배수와 공배수, 약수와 완전수, 최대공약수와 최소공배수, 근사값, 10진수와 2진수의 변환, 10진수와 16진수의 변환, BCD 코드와 3초과 코드의 변환, 1의 보수와 2의 보수, 패리티 비트 검증

4. 응용 알고리즘 – 배열

5. 응용 알고리즘 – 자료 구조

6. 실무 응용 알고리즘

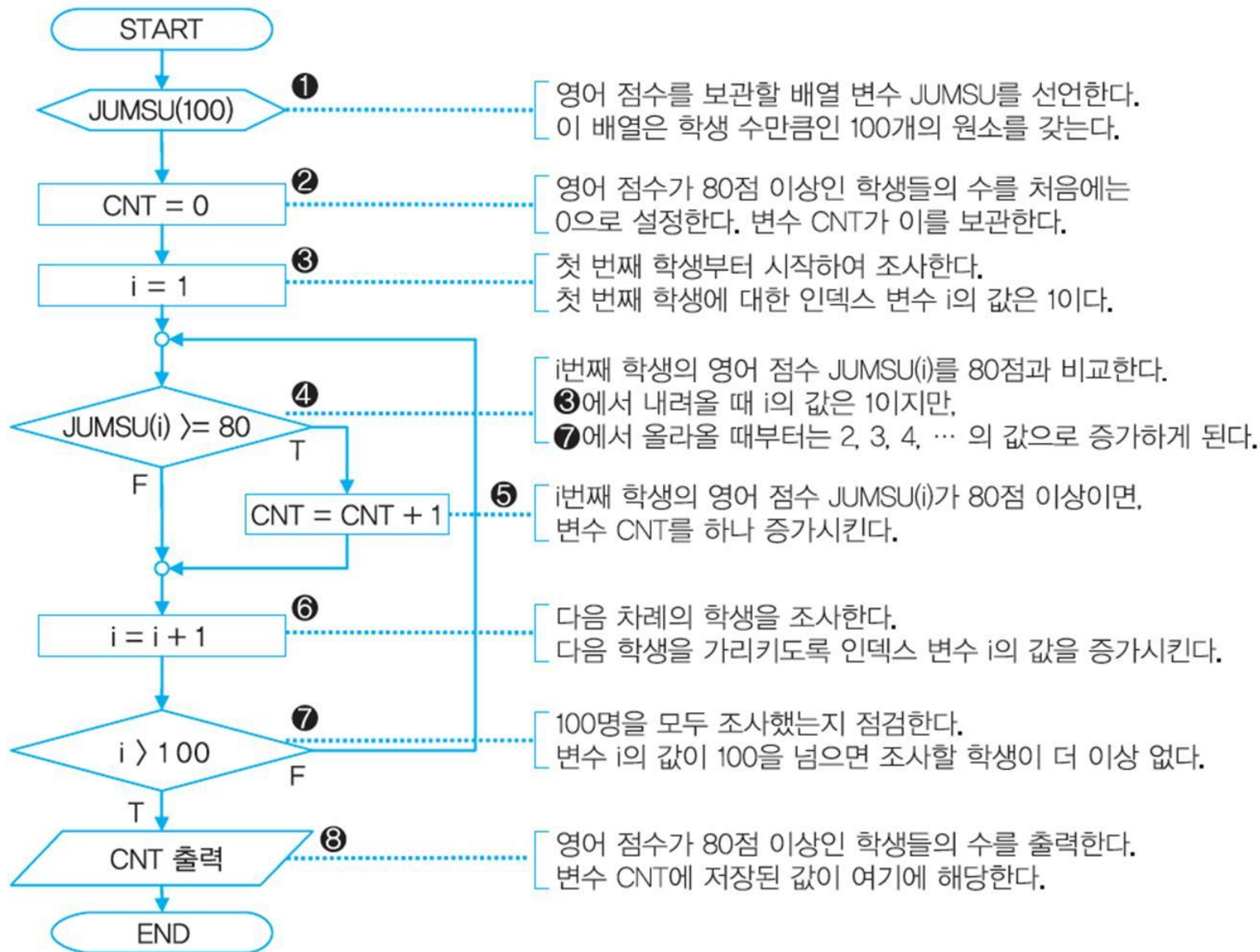
7. 알고리즘과 소프트웨어 개발

Section 1 Count 알고리즘

(문제) 영어 시험 성적이 80점 이상인 학생들의 수를 구하는 알고리즘을 제시하라.

- 전체 학생의 수는 100명이다.
- 영어 점수는 100점 만점을 기준으로 채점되었다.
- 영어 점수는 배열 변수 JUMSU(100)에 이미 저장되어 있다고 가정한다.

(순서도)

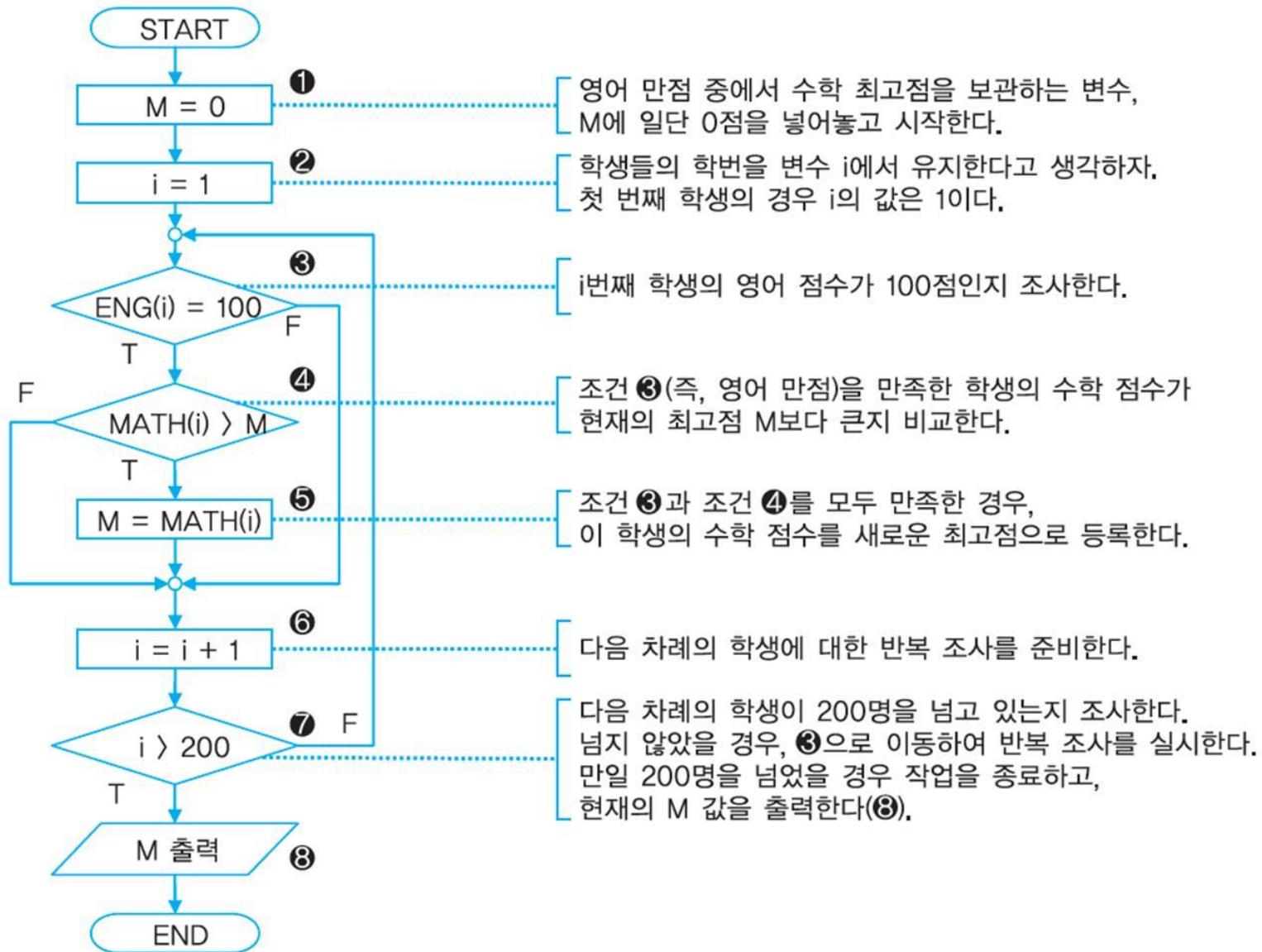


Section 2 최댓값과 최솟값

(문제) 영어 시험 만점 학생들 중에서 가장 높은 수학 시험 점수를 가지고 있는 학생의 수학 점수를 찾아서 출력하는 알고리즘을 제시하라.

- 시험을 본 학생은 모두 200명이다.
- 영어 점수는 배열 변수 ENG(200), 수학 점수는 배열 변수 MATH(200)에 저장되어있다.
- 영어, 수학 모두 100점 만점이다.

(순서도)

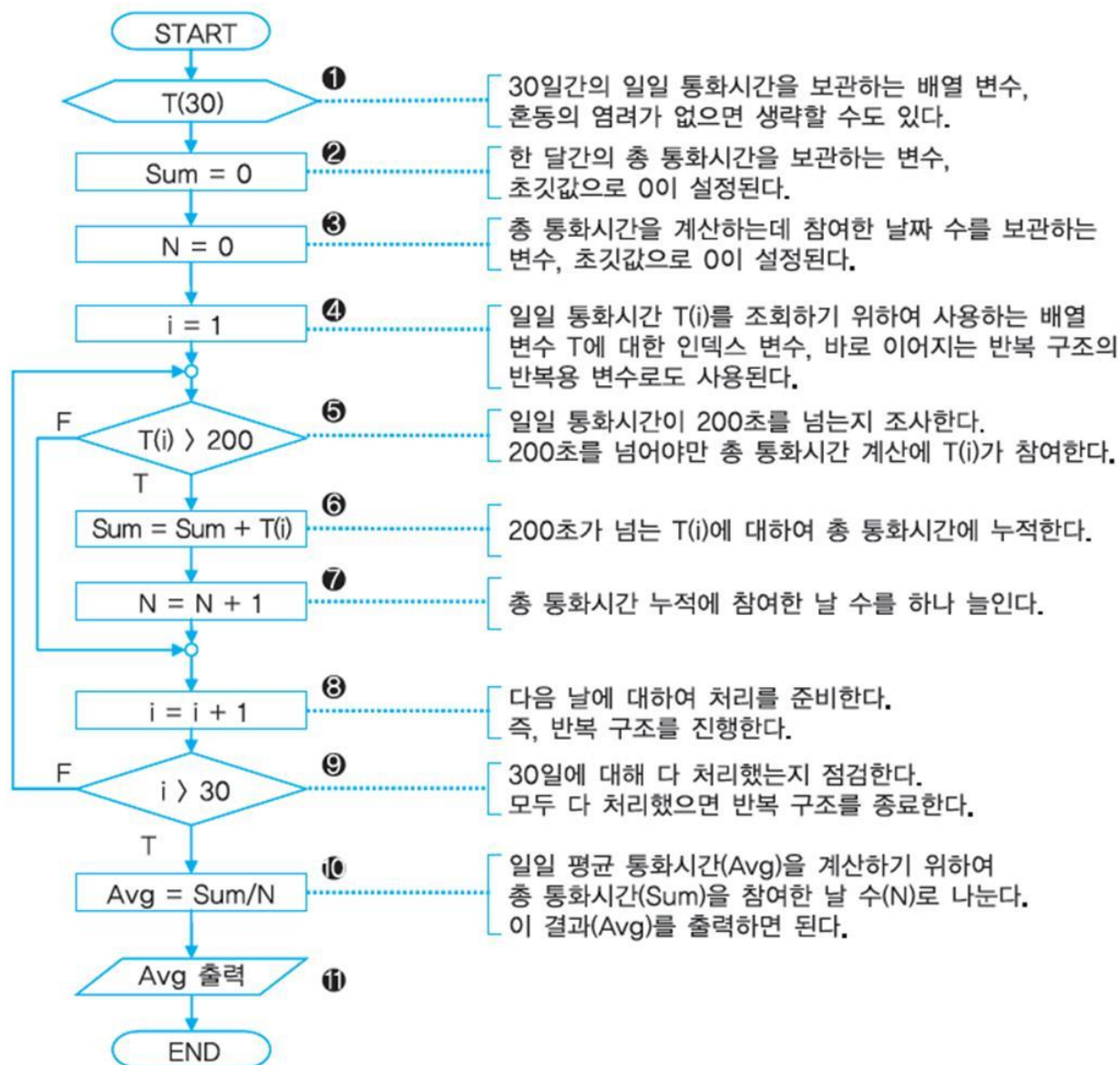


Section 3 합계와 평균

(문제) 다음과 같은 조건에서 휴대폰 고객 1명이 한 달 동안 사용하는 총 통화시간을 토대로 일일 평균 통화시간(초)을 구하는 알고리즘을 제시하라.

- 한 달을 30일로 잡고, 매일(i)의 통화시간은 변수 $T(i)$ 에 저장된다.
- 만일 일일 통화시간이 200초 이하이면 무료 서비스를 해 주며 총 통화시간에서 제외하고 평균 통화시간을 산정하는 과정에서도 제외한다.

(순서도)

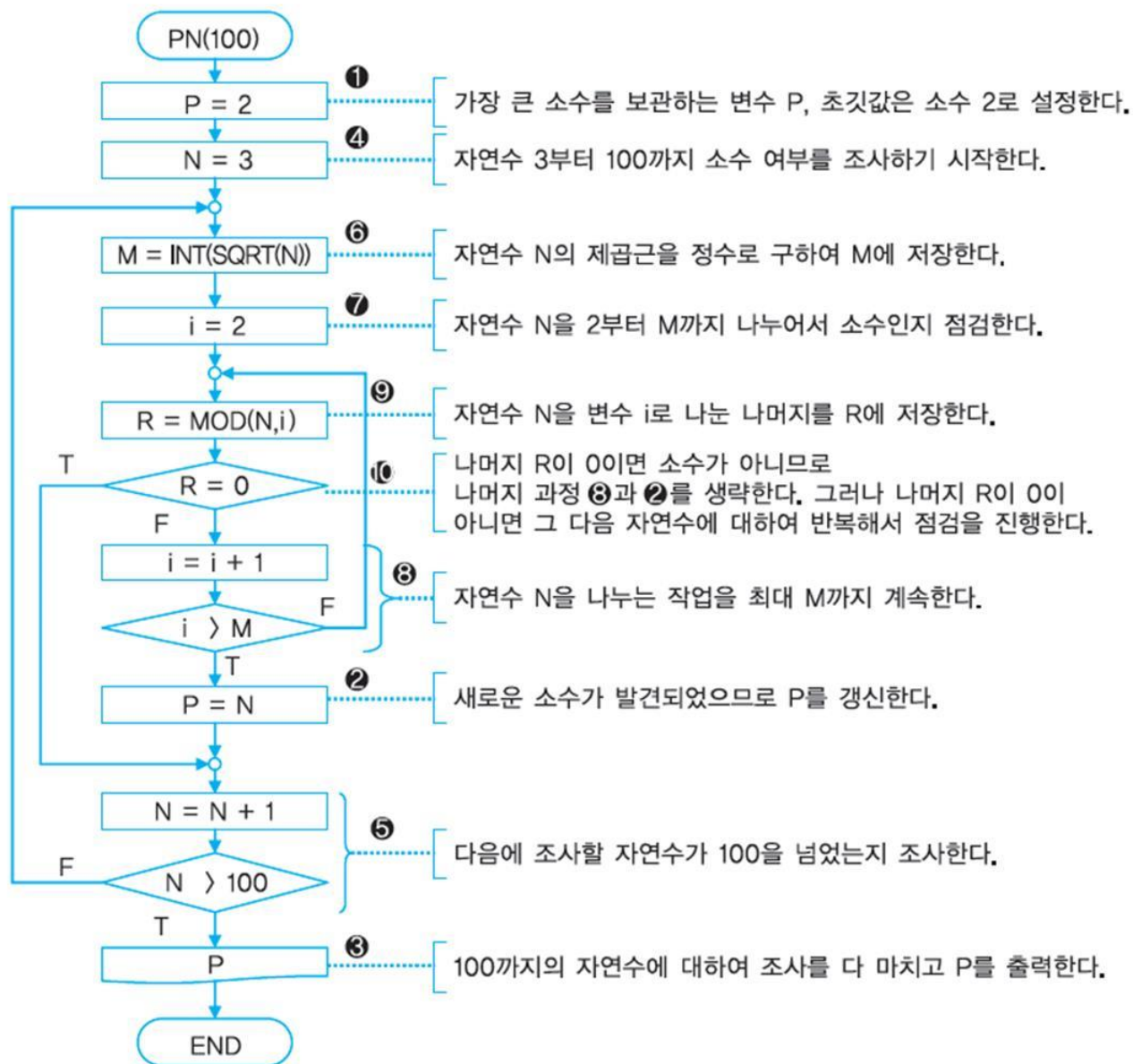


Section 4 소수 판별

(문제) 1부터 100 사이에서 가장 큰 소수를 구하는 알고리즘을 제시하라.

- 소수(Prime Number)란 1과 자기 자신 이외의 수로는 나누어떨어지지 않는 1보다 큰 자연수를 가리킨다. 예를 들어 2, 3, 5, 7, 11, 13, 17,...은 소수이다.
- 만일 자연수 N 이 소수라면, 2부터 N 의 제곱근 \sqrt{N} 까지의 자연수들 중에서 N 을 나누어 떨어지게 하는 수는 존재하지 않는다.
- N 의 제곱근(\sqrt{N})은 시스템 함수 SQRT(N)을 호출하여 계산한다.

(순서도)



Section 5 소인수 분해

(문제) 자연수 N 을 입력받아 소인수 분해하여 그 결과를 출력하는 알고리즘을 제시하라.

- 입력받은 값 N 은 1000 이하의 자연수라고 가정한다.
- 입력받은 정수 N 이 2보다 작으면 알고리즘을 종료한다.
- 입력받은 정수 N 이 소수이면 '소수'라고 출력한다.
- 입력받은 정수 N 이 소수가 아니면 소인수 분해한 결과를 출력한다.
- 단계별로 소인수 분해한 결과를 배열에 저장해 두었다가 나중에 한꺼번에 출력한다.
- 132는 $2 \times 2 \times 3 \times 11$ 과 같이 소인수 분해된다.
- MOD()는 나누기에서 나머지를 구하는 함수이다.
- 다양한 입력별 알고리즘의 실행 예시는 다음과 같다.

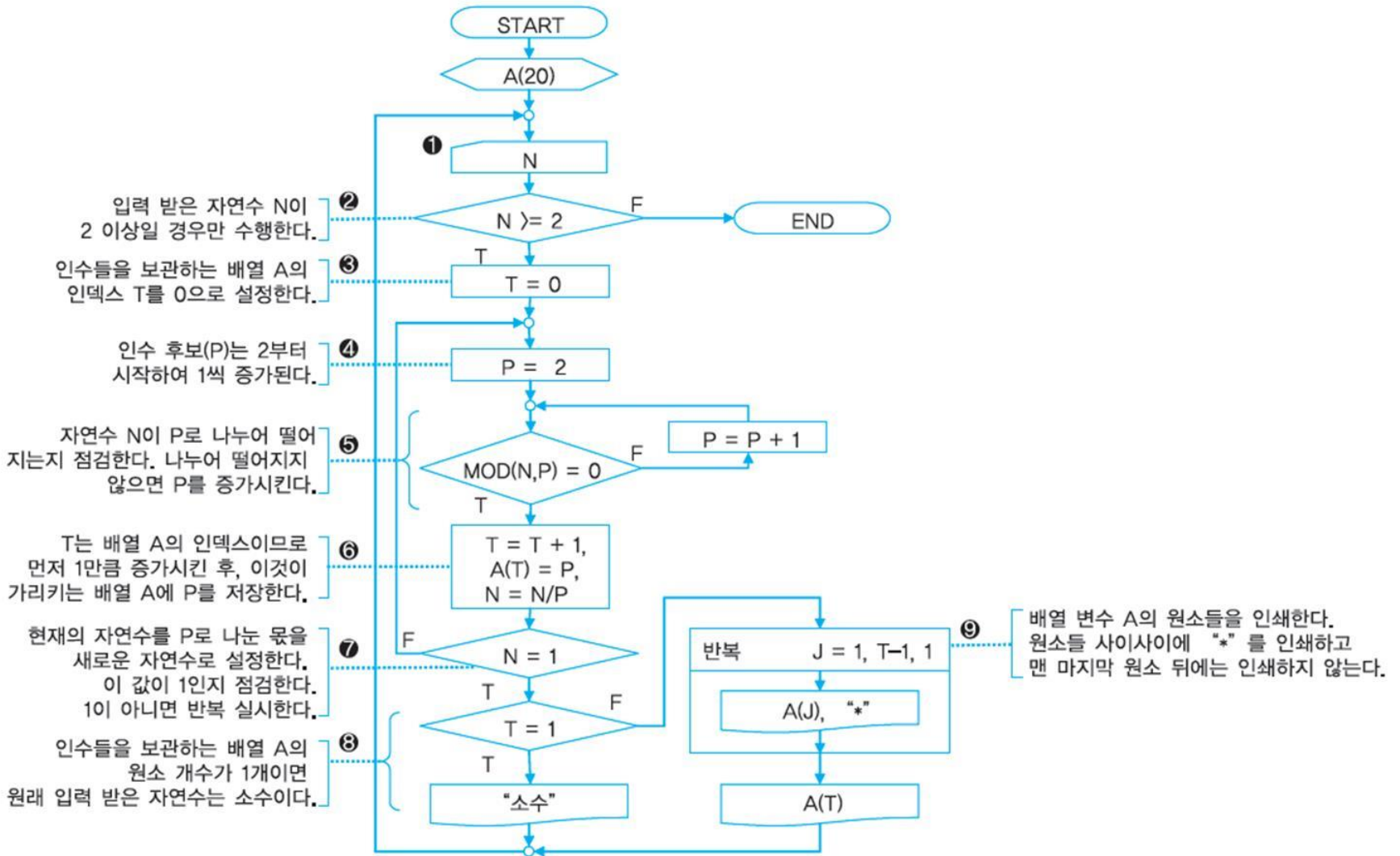
입력 : 132 → 출력 : "2*2*3*11"

입력 : 20 → 출력 : "2*2*5"

입력 : 37 → 출력 : "소수"

입력 : 0 → 알고리즘 종료

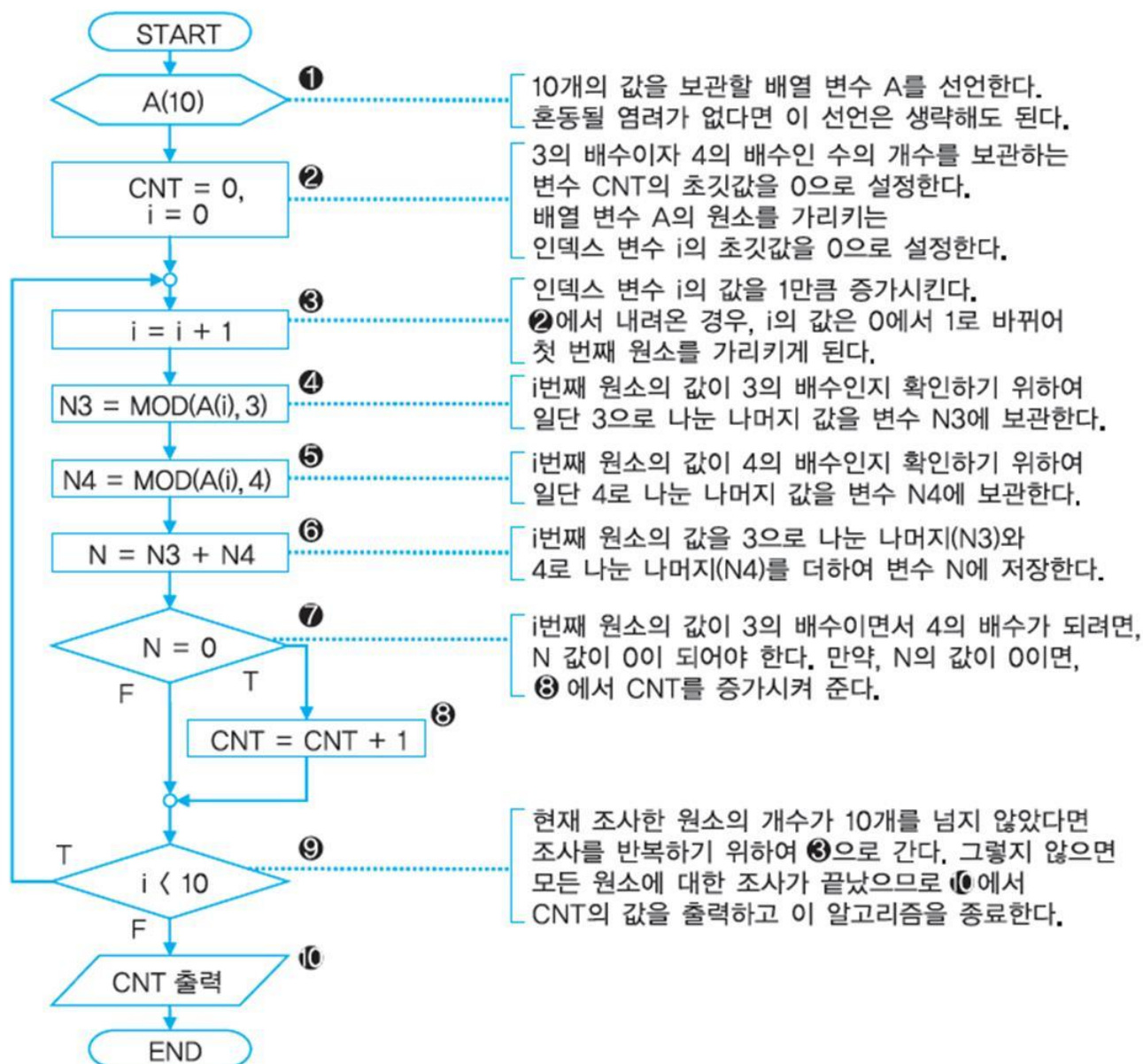
(순서도)



Section 6 배수와 공배수

(문제) 배열 A에 21, 17, 4, 51, 24, 75, 40, 27, 48, 72가 A(1)부터 시작하여 순차적으로 입력되어 있다고 가정할 때, 3의 배수이면서 4의 배수인 수의 개수를 구하는 알고리즘을 제시하라(단, A를 B로 나눈 나머지를 구해 주는 시스템 함수 MOD(A,B)를 활용함).

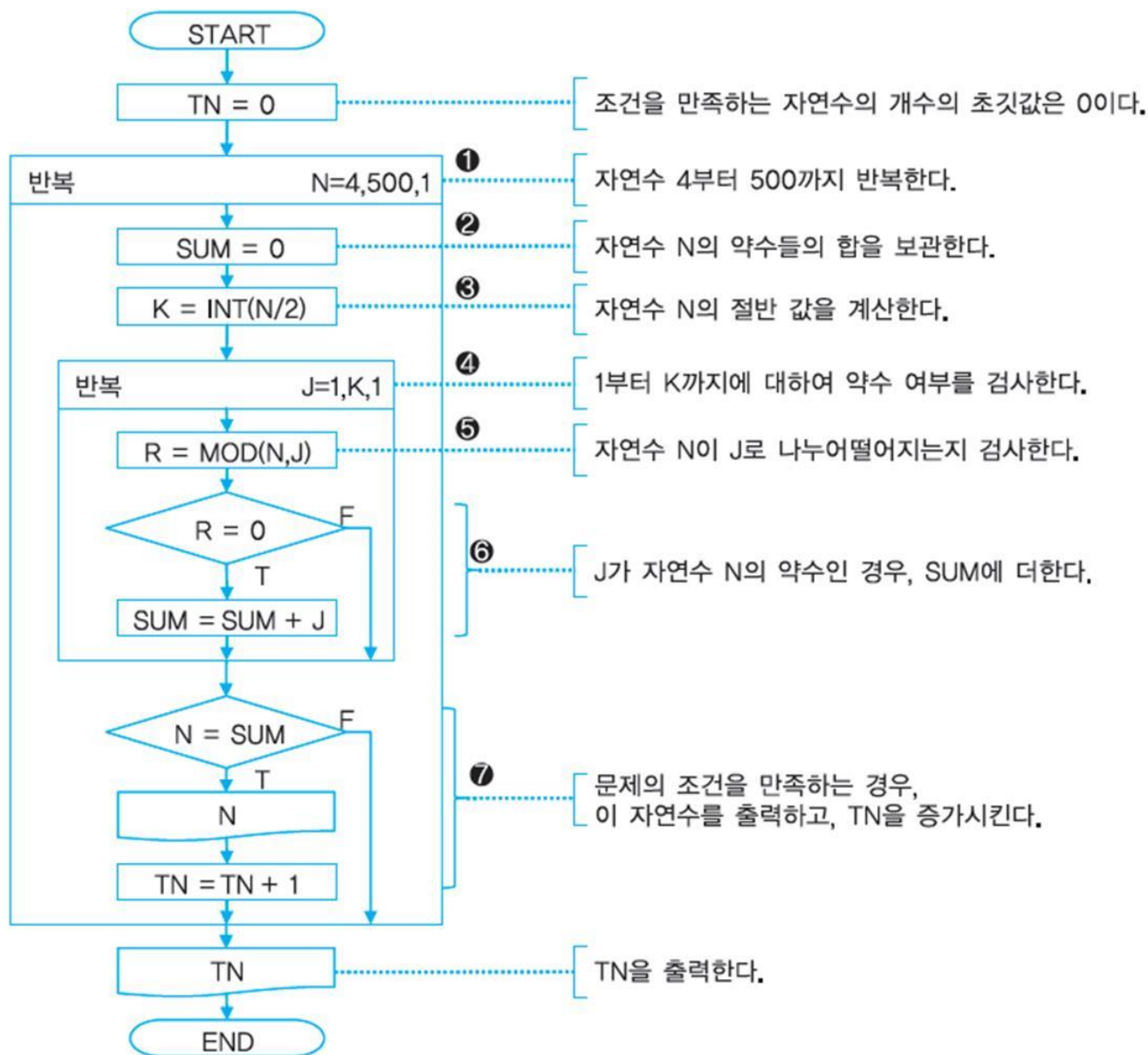
(순서도)



Section 7 약수와 완전수

(문제) 4부터 500까지의 자연수 중에서 완전수를 찾아 출력하고 그 개수를 구하는 알고리즘을 제시하라.

(순서도)



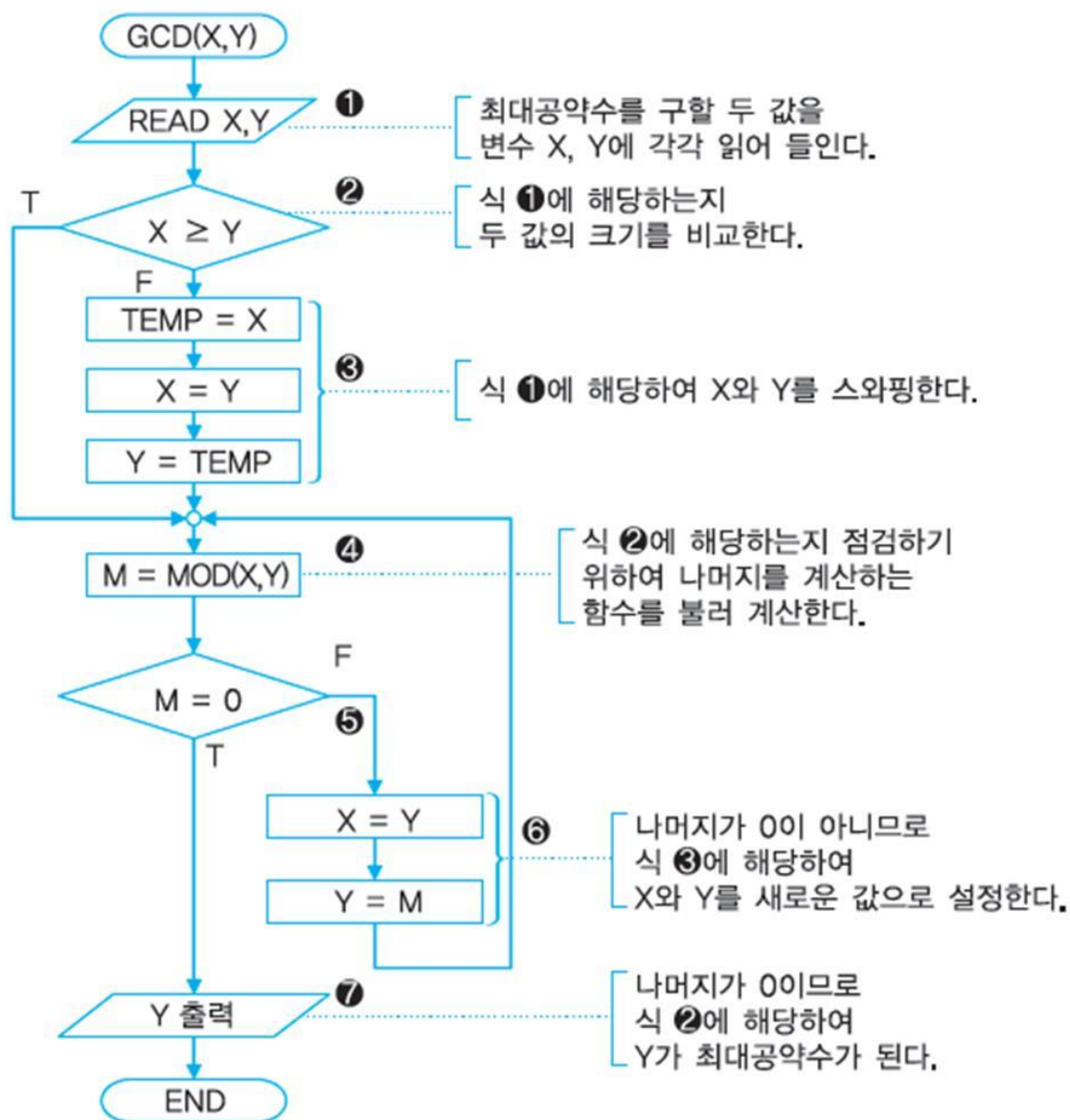
Section 8 최대공약수와 최소공배수

(문제) 다음과 같은 유클리드 호제법에 의하여 두 정수 A, B 의 최대공약수 (GCD)와 최소공배수(LCM)를 구하는 알고리즘을 제시하라.

- | | |
|--|---|
| ① $\text{GCD}(A, B) = \text{GCD}(B, A)$ | $A < B$ 일 경우 |
| ② $\text{GCD}(A, B) = B$ | $A \geq B$ 이면서 $\text{MOD}(A, B) = 0$ 일 경우 |
| ③ $\text{GCD}(A, B) = \text{GCD}(B, \text{MOD}(A, B))$ | 그 외(즉, $A \geq B$ 이면서 $\text{MOD}(A, B) \neq 0$ 일 경우) |

단, $\text{MOD}(A, B)$ 는 정수 A 를 정수 B 로 나눈 나머지를 구하는 시스템 함수이다.

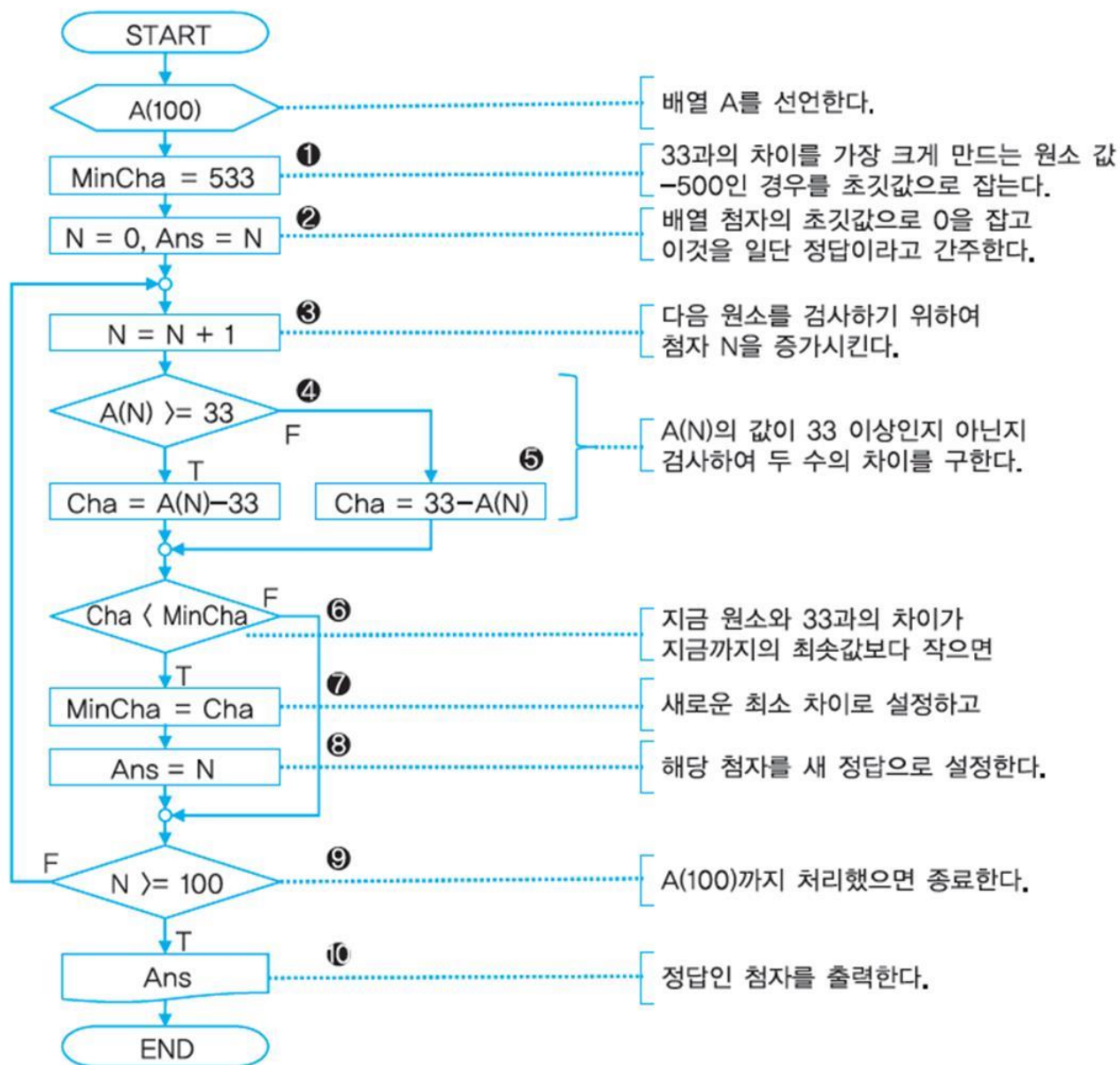
(순서도)



Section 9 근사값

(문제) 배열 $A(100)$ 의 원소 100개는 절대값이 500 이하이다. 이 중에서 정수 33에 가장 가까운 근사값을 찾아 해당 원소의 첨자를 출력하는 알고리즘을 제시하라.

(순서도)

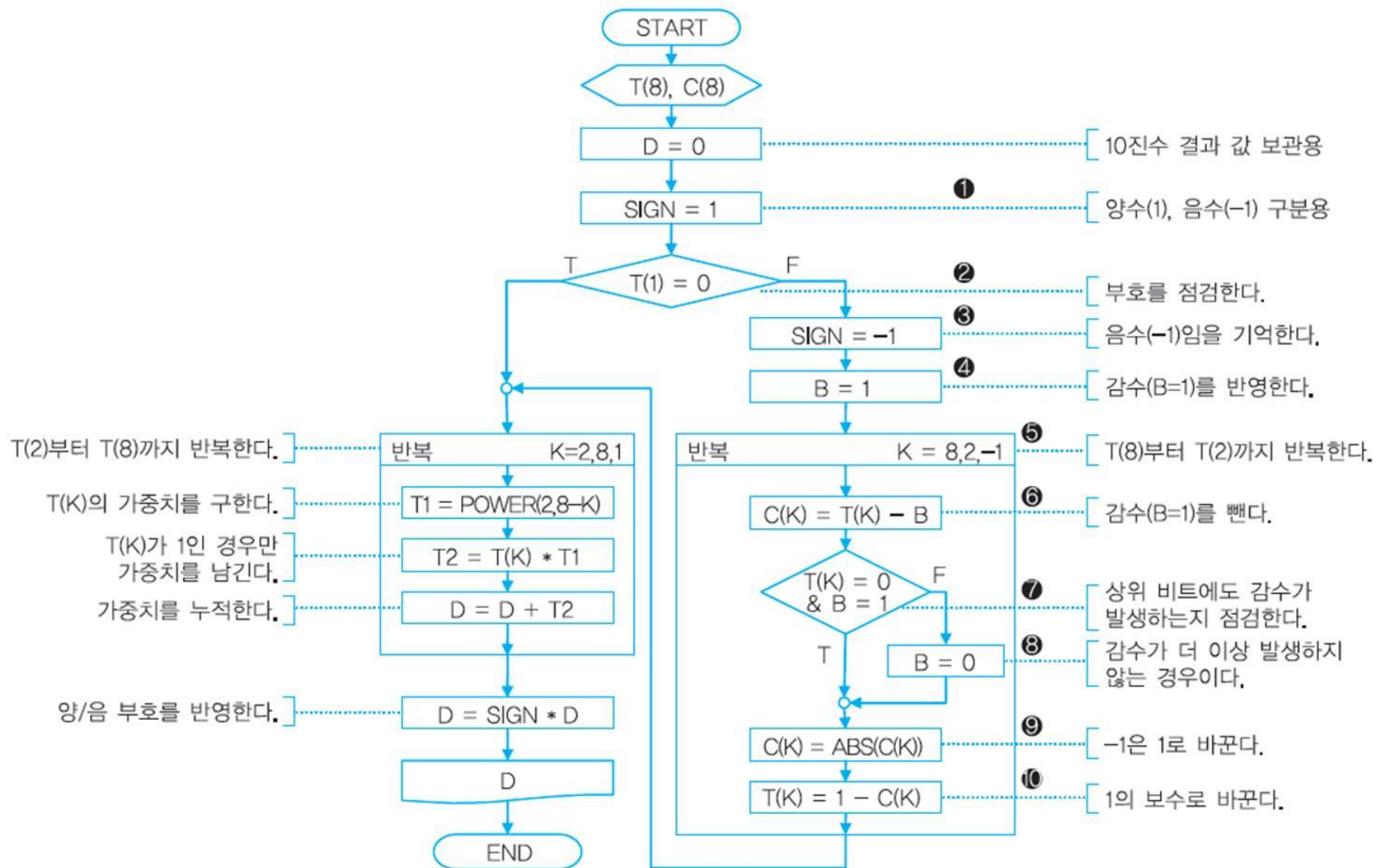


Section 10 10진수와 2진수의 변환

(문제) 다음 조건을 고려하여 2진수를 10진수로 변환하는 알고리즘을 제시하라.

- 변환할 2진수는 크기가 8인 배열 T에 저장되어 제공된다.
- 첫 번째 비트 T(1)은 2진수의 부호를 나타낸다. T(1)이 0이면 양수, 1이면 음수이다.
- 첫 번째 비트 T(1)의 값이 1인 음수의 경우, 2의 보수에 의하여 크기가 표현된다.
- 함수 ABS(X)는 X의 절댓값을 계산해주는 함수이다.
- 함수 POWER(X,Y)는 X의 Y제곱, 즉 X^Y 을 계산해주는 함수이다.

(순서도)



Section 11 10진수와 16진수의 변환

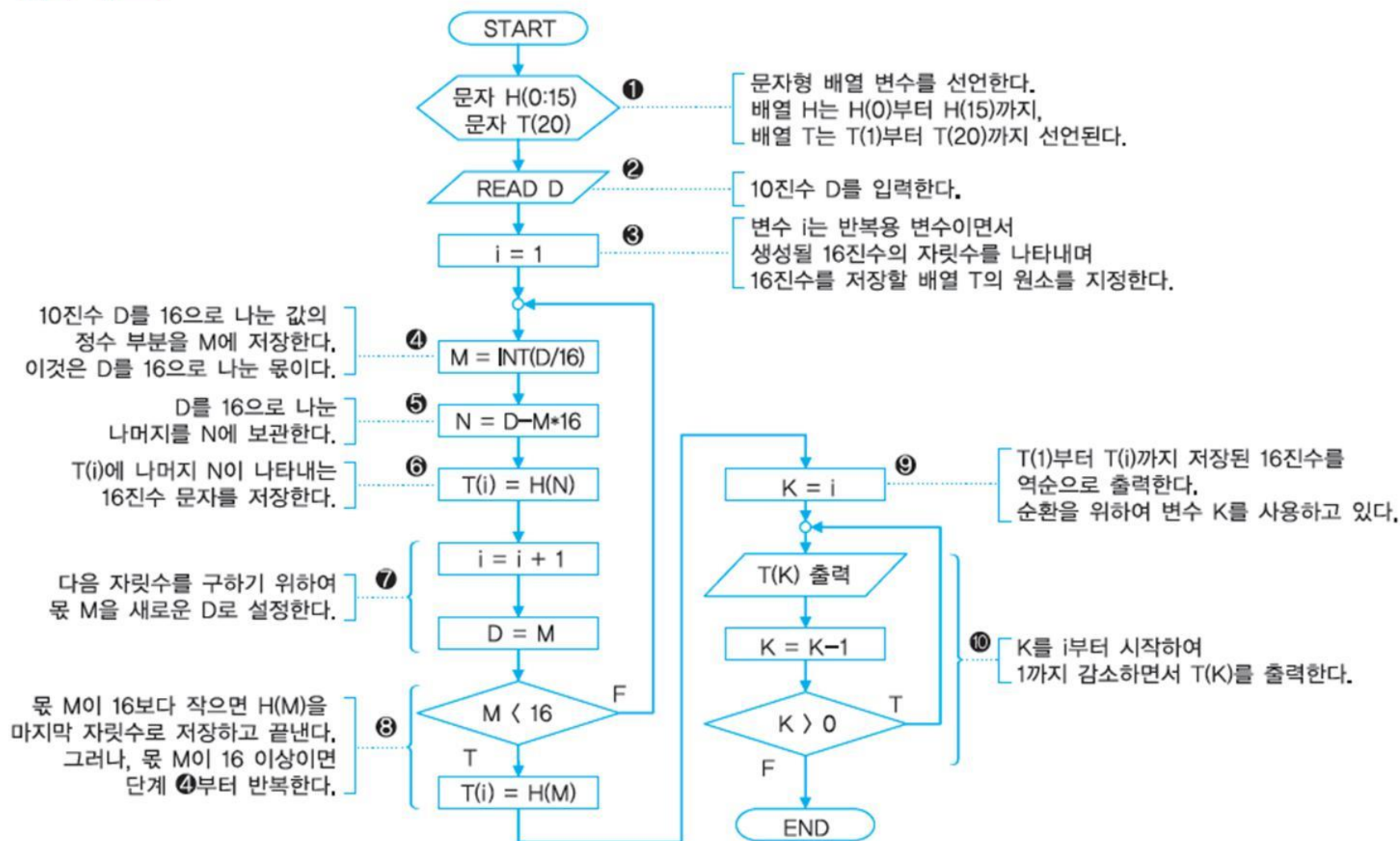
(문제) 다음과 같은 조건에서 10진수를 받아들여 16진수로 변환하여 출력하는 알고리즘을 제시하라.

- 입력으로 주어지는 10진수는 변수 D에 저장하여 처리한다.
- 16진수를 나타내는 각 숫자에 해당하는 문자는 문자형 배열 변수 H(16)에 저장되어 제공된다.

H(0)= "0", H(1)= "1", H(2)= "2",, H(14)= "E", H(15)= "F" (인덱스를 0부터 매김)

- 16진수는 최대 20자리까지 보관이 가능하며 문자형 배열 변수 T(20)을 이용하여 한 자리씩 저장된다.

(순서도)

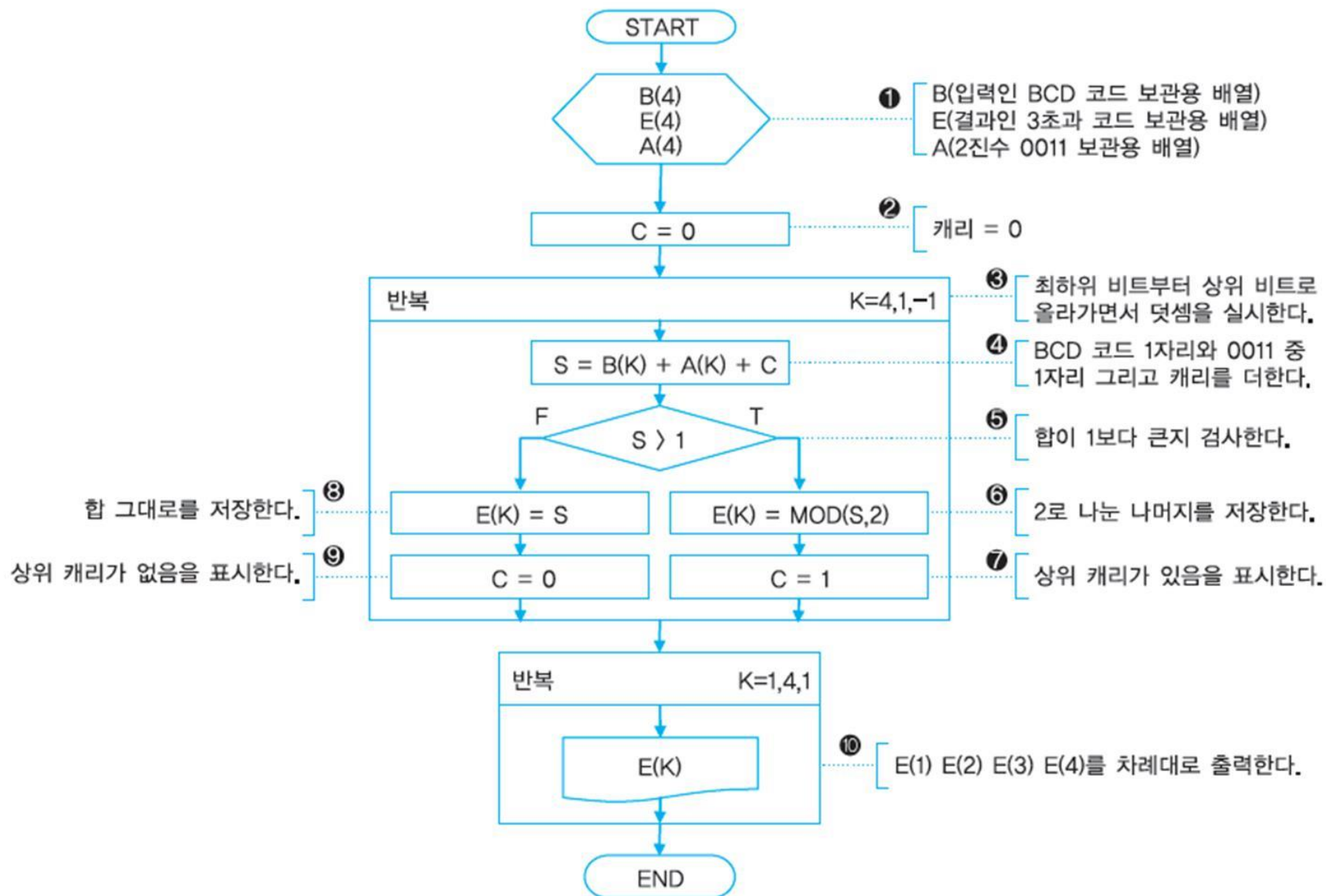


Section 12 BCD 코드와 3초과 코드의 변환

(문제) 다음과 같은 조건에서 4비트의 BCD 코드를 받아들여 4비트의 3초과 코드로 변환하여 출력하는 알고리즘을 제시하라.

- BCD 코드는 크기가 4인 배열 B(4)에 1비트씩 저장되어 입력으로 제공된다.
- 3초과 코드는 크기가 4인 배열 E(4)에 1비트씩 저장된 후 나중에 출력된다.
- 2진수 0011은 크기가 4인 배열 A(4)에 1비트씩 저장된다.

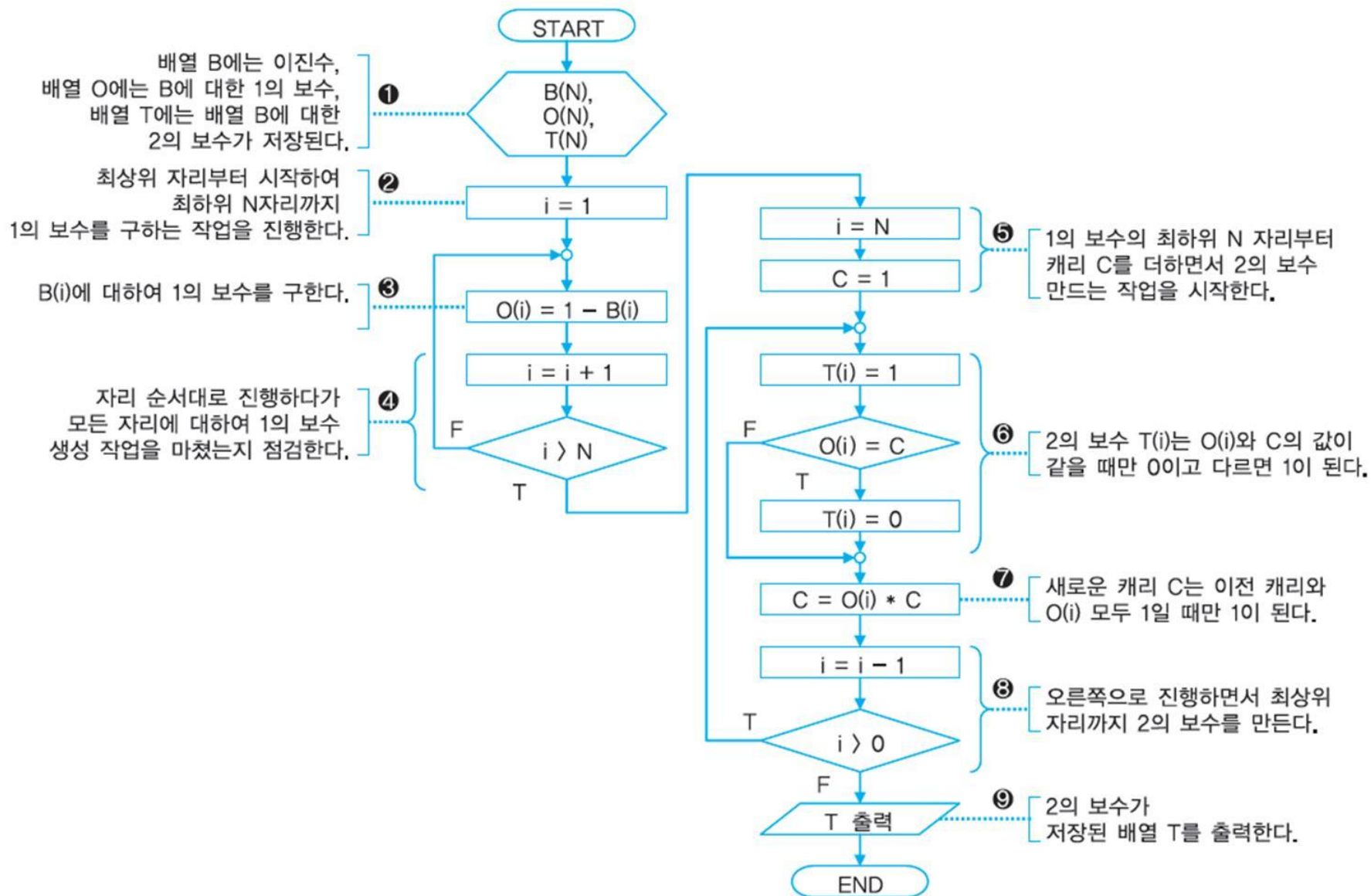
(순서도)



Section 13 1의 보수와 2의 보수

(문제) 어떤 2진수에 대하여 2의 보수를 구하는 알고리즘을 제시하라.

(순서도)



Section 14 패리티 비트 검증

(문제) 다음 조건으로 8비트 패리티 비트 오류 검사를 실시하는 알고리즘을 제시하라.

- 입력으로 총 9비트를 받아들인다.
- 첫 번째 비트는 패리티 검사 방식을 나타내는데, 그 값이 1이면 홀수 패리티, 0이면 짝수 패리티로 검사함을 의미한다.
- 나머지 8개의 비트는 0 또는 1의 값을 갖는 데이터 영역이다.
- 홀수 패리티는 8비트 데이터 영역에 저장된 1의 개수가 홀수 개인 경우에 오류가 없음을 의미한다.
- 짝수 패리티는 8비트 데이터 영역에 저장된 1의 개수가 짝수 개인 경우에 오류가 없음을 의미한다.
- 패리티 검사를 오류가 없으면 "오류 미발견"이라고 출력하고, 오류가 있으면 "오류 발견"이라고 출력한다.

(순서도)

