

순서도 그리기 연습

1. 자연수 a 의 모든 약수 구하기
2. 자연수 a 와 b 의 최대공약수 구하기
3. 유클리드 호제법으로 자연수 a 와 b 의 최대공약수 구하기
4. 자연수 a 와 b 의 최소공배수 구하기
5. 자연수 a, b, c 의 최소공배수 구하기
6. 자연수 N 을 입력 받아, 다음과 같이 출력하기 ($N = 17$ 일 때의 예)
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17
7. 자연수 a 의 완전수, 과잉수, 부족수를 판별하는 함수 `isPerfect(a)`
8. 자연수 a 의 소수 여부를 판별하는 함수 `isPrime(a)`
9. 에라토스테네스의 체 기법을 사용하여 2부터 n 사이의 소수 출력하는 함수 `erathos(n)`
10. 자연수 a 의 계승(factorial)을 계산하는 함수 `fact(a)`
11. n 개의 피보나치(Fibonacci) 수 출력하는 함수 `fib(n)`
12. a, x, e 를 입력 받아 항(term)의 값을 계산하는 함수 `evalTerm(a, x, e)`
예를 들어, `evalTerm(4, 2, 3)`을 입력하면, $4 \times 2^3 = 32$ 를 반환

14. N개의 정수로 이루어진 리스트 a를 버블 정렬 기법을 사용하여 정렬하는 함수
bubbleSort(a, n)

<버블 정렬>

정수를 정렬할 때, 입력되는 데이터의 개수가 3개일 때는 $3(=2+1)$ 개의 if문이 필요하고, 데이터의 개수가 4개일 때는 $6(=3+2+1)$ 개의 if문이 필요하게 된다. 또한, 데이터의 개수가 5개가 되면 10개, 6개가 되면 15개의 if문이 필요하게 되므로, 이러한 방식으로 정렬 프로그램을 작성하는 것은 매우 비효율적이다. 따라서 while문이나 for문과 같은 반복문을 사용하여 프로그램을 작성하는 것이 효율적이다. 반복문을 사용하여 매 반복시마다 가장 큰 원소를 리스트의 가장 뒤로 보내는 정렬 방법을 버블 정렬(Bubble Sort)라고 한다.

>>>

정렬 전 리스트 : [5, 4, 3, 2, 1]

1 회 반복 후 결과 : [4, 3, 2, 1, 5]

2 회 반복 후 결과 : [3, 2, 1, 4, 5]

3 회 반복 후 결과 : [2, 1, 3, 4, 5]

4 회 반복 후 결과 : [1, 2, 3, 4, 5]

>>>
