

# Program = Data Structure + Algorithm

## ◆ 자료구조와 알고리즘

## ◆ 알고리즘의 정의와 표현

- 정의 : 주어진 문제를 풀기 위한 절차나 방법  
실행 명령어들의 순서적 나열

- 표현(기술방법)

Flowchart

Pseudo Code

Program Language

- 프로그램 문장 구성 =  
선언문 + 실행명령문
- 대표적인 컴퓨터 실행 명령문
  - 대입문
  - 조건문
  - 반복문
  - 입출력문
  - 함수호출문

# 알고리즘 작성 연습

## [연습 1]

세 수를 읽어 들어서 그 중 가장 큰 값을 찾아 출력하는 알고리즘을 작성하시오(교과서p.58)

input : n1, n2, n3

output : large

**START**

read(n1, n2, n3)

if (n1 > n2) large = n1

else large = n2

if (n3 > large) large = n3

print(large)

**END**

# 알고리즘 작성 연습

## [연습 2]

50개의 수(data)를 읽어 들여 그 수가 0보다 크면 pos\_sum에 더하고  
0보다 작으면 nega\_sum에 더하여  
그 결과 pos\_sum과 nega\_sum을 출력하시오.

input : data

output : pos\_sum, nega\_sum

- 반복제어변수(LCV) – count  
loop control variable

- 일반화(상수 -> 변수)  
generalization

START

count = 0, pos\_sum = 0, nega\_sum = 0

while (count < 50) do

    read(data)

    if (data > 0) pos\_sum = pos\_sum + data

    else if (data < 0) nega\_sum = nega\_sum + data

    count = count + 1

endwhile

print(pos\_sum, nega\_sum)

END

# 알고리즘 작성 연습

## [예제1]

n 개의 양의 정수를 입력 받아 홀수의 합과 짝수의 합을 구하는 알고리즘을 작성하시오.

### (1) 문제분석

- Input : n, num
- Output : oddsum, evensum
- Process : %(나머지 연산자)를 이용  
    num %2 -> 0이면 짝수  
            -> 1이면 홀수

### (2) 알고리즘 작성

START

```
oddsum = 0, evensum = 0
count = 0
read(n)
while (count < n) do
    read(num)
    if (num % 2 == 0)
        evensum = evensum + num
    else
        oddsum = oddsum + num
    endif
    count = count + 1
endwhile
print(evensum, oddsum)
```

END

# 알고리즘 작성 연습

## [예제2]

양의 정수를 읽어 끝자리 숫자부터 출력하는 알고리즘을 작성하시오.

### (1) 문제분석

· Input : num(num > 0)

· Output : r

예) num01 1 2 3 40이면

4 3 2 10이 출력되도록 한다.

### Process

$r = \text{num} \% 10$	4	3	2	1
$\text{num} = \text{num} / 10$	123	12	1	0

### (2) 알고리즘 작성

START

    read(num)

    while(num > 0) do

        r = num % 10

        num = num / 10

        print(r)

    endwhile

END

# 알고리즘 작성 연습

## [예제3]

0-100점인 n명 학생의 성적을 입력받아  
평균과 최고점을 출력하는 알고리즘을 작성  
하시오.

### (1) 문제분석

- Input : n, score
- Output : avg, max
- Process
  - ① n명을 세는 변수 count를 0으로 초기화 한다.
  - ② 평균을 구하기 위해서는 우선 합을 구해야 한다. (avg = 0)
  - ③ 최고점을 0으로 초기화해야 한다. (max = 0)

### (2) 알고리즘 작성

START

avg = 0, max = 0

read(n)

for i ← 0 to n-1 do

read(score)

avg = avg + score

if(score > max) max = score

endfor

avg = avg / n

print(avg, max)

END

# 알고리즘 작성 연습

## [예제4]

다음의 ssum과 psum을 계산하는 알고리즘을 작성하시오.

$ssum = 1 + (1+2) + \dots + (1+2 + \dots + n)$

$psum = 1 + (1*2) + \dots + (1*2 * \dots * n)$

### (1) 문제분석

- Input : n
- Output : ssum, psum
- Process
  - ① ssum과 psum을 0으로 초기화 한다.
  - ② 1부터 n까지 반복하여 부분 합 temp1과 부분 곱 temp2를 구한다.  
(즉 반복문 안에 반복문을 사용해야 한다.)

### (2) 알고리즘 작성

START

  read(n)

  ssum = 0, psum = 0

  for i ← 1 to n do

    temp1 = 0

    temp2 = 1

    for j ← 1 to i do

      temp1 = temp1 + j

      temp2 = temp2 \* j

    endfor

    ssum = ssum + temp1

    psum = psum + temp2

  endfor

  print(ssum, psum)

END

# 알고리즘 작성 연습

## [예제5]

음수가 나올 때까지 양수를 읽은 후, 입력된 데이터 수와 최대 값을 구하는 알고리즘을 작성하시오(단 처음부터 음수를 입력하는 경우 “No data!!” 라고 출력하고 끝낸다).

### (1) 문제분석

- Input : num
- Output : size, max
- Process

- ① max는 0으로 초기화한다.  
데이터의 수 size는 0으로 초기화한다.
- ② 데이터 num은 우선 1개 읽어 양수인 경우 처리 과정을 시작한다.
- ③ 예로 num <- 70 15 27 99 85 92 10 -1  
인 경우 size는 7이고 max는 99가 된다.

### (2) 알고리즘 작성

START

max = 0, size = 0

read(num)

while(num > 0) do

size = size + 1

if (num > max) max = num

read(num)

endwhile

print( “데이터 수 : ” , size)

if(size == 0) print("No data!!")

else print( “최대 값 : ” , max)

END