

알고리즘의 작성

알고리즘 작성단계

- (1) 문제 파악
- (2) 입력과 출력변수 도입하여 문제 재정의
- (3) 알고리즘구상
임의의 입력변수에 데이터(특정 값)를 넣어 출력이 나올 때까지의 과정을 **simulation** 하면서 그 과정을 알고리즘 명령문으로 기술해 본다.
- (4) 기술한 알고리즘의 검토
여러 테스트 케이스에 대하여 알고리즘을 simulation해 보면서 알고리즘 명령문을 수정-보완해 완성한다

알고리즘의 구현

1. 작성된 알고리즘 검토

일반적인 경우와 예외적인 경우로 나누어 모든 경우를 다 포함하는지
더 세분화 해야 할 단계는 없는지
조건이 문제에 맞게 정해져 있는지
정해진 횟수만큼 반복하는지

2. 주어진 알고리즘을 보고 변수의 형과 배열의 크기 등을 고려하여 선언문 작성

3. 프로그래밍언어 구문을 고려하여 실행문 작성

4. 변수의 형에 따른 오류 등 컴퓨터 계산방식에 따라 검토

예) $avg = sum/n$

$avg = (\text{double}) sum / n$

5. 사용자 인터페이스 고려 : 입력 받기 전에 입력할 내용을 안내하는 출력문 쓰기

알고리즘 작성과 구현 연습(1)

(예-1) n개 자료의 값을 읽어 0이 아닌 값을 곱하여
출력하는 알고리즘을 작성하시오

입력 : n, data
출력 : mult

n : 7

data : 3 4 0 1 10 0 5

1. mult는 1로 초기화
2. 다음의 과정을 n번 반복
 데이터 읽기
 0 이아닌 값은 mult에 곱하기

Start

```
read(n)  
mult = 1
```

```
count = 0  
while (count < n) do  
    read(data)  
    if (data !=0) then  
        mult = mult * data  
    endif  
    count = count + 1  
endwhile
```

```
print(mult)
```

End

알고리즘 작성과 구현 연습(1)

Start

```
mult = 1
read(n)
```

```
for i=0 to n-1 do
    read(x)
    if (x !=0) then
        mult = mult * data
    endif
endfor
```

```
print(mult)
```

End

* n개의 데이터가 모두 0인 경우?

```
main()
```

```
{
```

```
int n, i, x;
```

```
long mult = 1;
```

```
printf("Enter the number of data to be processed : ");
scanf("%d", &n);
```

```
printf("Enter %d numbers : ", n);
```

```
for (i=0; i < n; i++) {
```

```
    scanf("%d", &x);
```

```
    if (x != 0) {
```

```
        mult *= x;
```

```
    }
```

```
}
```

```
printf(" The result = %ld\n", mult);
```

```
}
```

알고리즘 작성과 구현 연습(1)

// 예외적인 경우 고려

Start

mult = 1

read(n)

allzero = 1

for i=0 to n-1 do

read(x)

if (x !=0) then

mult = mult * data

allzero = 0

endif

endfor

if (allzero==1) mult = 0

print(mult)

End

main()

{

int n, i, x, allzero=1;

long mult = 1;

printf("Enter the number of to be processed : ");

scanf("%d", &n);

printf("Enter %d numbers : ", n);

for (i=0; i < n; i++) {

scanf("%d", &x);

if (x != 0) {

mult *= x;

allzero = 0;

}

}

if (allzero) mult = 0;

printf("Answer = %d\n", mult);

}

알고리즘 작성과 구현 연습(2)

(예-2) 배열에 n개의 성적데이터를 읽어

평균을 출력하고 평균과 평균 이상인 데이터의 수를 출력

1. 입력과 출력

Input : n, scores(array)

Output : avg, over_count

2. 배열의 최대 크기 결정

#define DMAX 100

3. 초기치 설정 : sum = 0, over_count = 0, i=0

4. 알고리즘구상

- (1) 배열에 읽으면서 합을 구한다.(반복)
- (2) 평균을 구한다.
- (3) 배열에 저장된 데이터를 보면서
평균보다 큰 데이터의 수를 구한다.
(반복 + 조건)
- (4) 결과를 출력한다

알고리즘 작성과 구현 연습(2)

START

// PART 1

sum = 0

read(n)

for i=0 to n-1 do

 read(scores(i))

 sum = sum + scores(i)

endfor

avg = sum / n

print(avg)

//PART 2

over_count = 0

for i=0 to n-1 do

 if (scores(i) > avg) over_count++

endfor

print(over_count)

END

#define DNUM 100

main()

{

 int scores[DNUM], n, i, sum, over_count;

 double avg;

 sum = 0;

 printf("The number of data to be processed");

 scanf("%d", &n);

 printf("Enter %d data : \n", n);

 for (i=0; i < n; i++) {

 scanf("%d", &scores[i]);

 sum = sum + scores[i];

 }

 avg = (double) sum / n;

 printf("Average = %.2f\n", avg);

 over_count = 0;

 for (i=0; i < n; i++)

 if (scores[i] > avg) over_count++;

 printf("over average = %d\n", over_count);

}

알고리즘 작성과 구현 연습(3)

(예-3) n명의 학생의 학번과 성적을 이차원 배열에 읽어 들어 학번을 입력하면 그의 성적과 등수를 출력하는 C 프로그램을 작성하시오.

Input: n, scores(n, 2), sid

scores는 $n * 2$ 의 이차원 배열이다.

Output : jumsu, rank

717	88
625	92
810	80
707	75
530	98
424	70
877	65
701	85
628	70
505	78

n = 10, sid = 810

jumsu = 80

rank = 5

1. 2차원 배열 scores에 데이터를 읽어 들인다.

2. 임의의 sid를 입력 받아 해당 학생의 성적 jumsu를 출력해 준다.
즉, sid가 scores(i, 0)이면 jumsu는 scores(i, 1)이 된다.

3. 학생의 등수 rank는 각 학생의 성적 scores(i, 1)가 jumsu보다 큰 학생의 수 + 1이 된다.

알고리즘 작성과 구현(3)

Start

```
read(n)
print("Input id and score : ")
for i <- 0 to n-1 do
    read(scores(i, 0), scores(i, 1))
endfor
print("Input student id to be searched...")
read(sid)
for i <- 0 to n-1 do
    if (sid = scores(i, 0))
        jumsu = scores(i, 1)
        break
    endif
endfor
rank = 0;
for i <- 0 to n-1 do
    if(scores(i, 1) > jumsu) rank = rank + 1
endfor
print(sid, "의 점수 : ", jumsu)
print(sid, "의 등수 : ", rank + 1)
```

End

main()

```
{
    int i, scores[NUM][2], sid, jumsu, rank, n;
    printf("Enter n :");
    scanf("%d", &n);
    printf("Input id and score : \n");
    for (i=0; i < n; i++)
        scanf("%d %d", &scores[i][0], &scores[i][1]);
    printf("Input student id to be search : ");
    scanf("%d", &sid);
    for (i=0; i < n; i++)
        if (sid == scores[i][0]) {
            jumsu = scores[i][1];
            break;
        }
    rank = 0;
    for (i=0; i < n; i++)
        if (scores[i][1] > jumsu) rank++;
    printf("%d번의 점수 : %d\n", sid, jumsu);
    printf("%d번의 등수 : %d\n", sid, rank+1);
}
```

C함수 작성법(1)

1. 함수 원형(Function Prototype)

2. 함수 정의(Function Definition)

3. 함수 호출(Function Call)

- Parameter
- Call by value

```
int my_pow(int x, int y);
```

```
main()  
{
```

```
    int k;
```

```
    for (k=2; k < 6; k++)
```

```
        printf("%d ** %d = %d\n", k, k+1, my_pow(k, k+1));
```

```
}
```

```
int my_pow(int x, int y)
```

```
{
```

```
    int i, ans=1;
```

```
    for (i=0; i < y; i++)
```

```
        ans = ans * x;
```

```
    return ans;
```

```
}
```

2 ** 3 = 8

3 ** 4 = 81

4 ** 5 = 1024

5 ** 6 = 15625

C함수 작성 예

Start

```
mult = 1
allzero = 1
count = 0
while (count < n) do
    read(x)
    if (x != 0) then
        mult = mult * x
        allzero = 0
    endif
    count = count + 1
endwhile
if (allzero) mult = 0
print(mult)
```

End

```
long example(int n)
{
    int i, x, allzero=1;
    long mult = 1;

    for (i=0; i < n; i++) {
        scanf("%d", &x);
        if (x != 0) {
            mult *= x;
            allzero = 0;
        }
    }
    if (allzero) mult = 0;
    return mult;
}
```

C 함수 작성법(2)

함수 호출시 배열을 파라미터로 넘겨주기

- 배열은 같은 형의 데이터의 모임
- 배열이름을 넘겨준다.
- 배열이름은 배열의 첫 데이터의 주소

data = &data[0]

int pcount(int a[])

```
{  
    int j, result=0;  
    for (j=0; j < 10; j++)  
        if (a[j] > 0) result++;  
    return result;  
}
```

a[0] ← data[0]

a[1] ← data[1]

.

.

.

.

a[9] ← data[9]

main()

{

int data[10], k;

printf("Enter 10 data : ");

for (k=0; k < 10; k++)

scanf("%d", &data[k]);

printf("양의 갯수는 %d\n", pcount(data));

}

C함수 구현 연습(1)

```
#define DNUM 100
```

```
int pcount(int a[], int n);
```

```
double avg(int a[], int n);
```

```
int large(int a[], int n);
```

```
int small(int a[], int n);
```

```
main()
{
    int data[DNUM], k, n;
    printf("처리할 데이터의 수 : ");
    scanf("%d", &n);
    printf("%d개의 정수를 입력하세요. ", n);
    for (k=0; k < n; k++)
        scanf("%d", &data[k]);
    printf("양의 갯수는 %d\n", pcount(data, n));
    printf("데이터 평균 = %.2f\n", avg(data, n));
    printf("최대값 = %d\n", large(data, n));
    printf("최소값 = %d\n", small(data, n));
}
```

```
int small(int a[], int n)
```

```
{
    int imin = a[0], i;
    for (i=1; i < n; i++) {
        if (a[i] < imin) imin = a[i];
    }
    return imin;
}
```

```
int large(int a[], int n)
```

```
{
    int i, max;
    max = a[0];
    for (i=1; i < n; i++)
        if (a[i] > max) max = a[i];
    return max;
}
```

C함수 구현 연습(1)

```
int pcount(int a[], int n)
{
    int j, result=0;
    for (j=0; j < n; j++)
        if (a[j] > 0) result++;
    return result;
}
```

```
double avg(int a[], int n)
{
    int i, sum=0;

    for (i=0; i < n; i++)
        sum = sum + a[i];
    return (double)sum / n;
}
```

C함수 구현 연습(2)

(예-4) n명 학생의 학번과 0-100점 사이의 학생 성적을 이차원 배열에 읽어 들어
성적의 평균과 최고점을 출력하고
학번을 입력하면 그의 등수를 출력하는 알고리즘을 작성하시오.

main()에서는 이차원배열 scores에 데이터를 입력 받고 함수 호출한 결과를 출력한다.

성적의 평균을 구하는 함수 find_avg를 작성한다.

성적의 최고점을 구하는 함수 find_max를 작성한다.

학번을 가지고 등수를 구하는 함수 find_rank를 작성한다.

Start(main)

read(n)

print("Input id and score : ")

for i = 0 to n-1 do

read(scores(i, 0), scores(i, 1))

endfor

print("평균 : ", find_avg(scores, n))

print("최대값 : ", find_max(scores, n))

print("Input student id to be searched...")

read(sid)

print(sid, "의 등수 : ", find_rank(scores, n, sid))

End

C함수 구현 연습(2)

Start(find_avg(sdata, n))

sum = 0

for i = 0 to n-1 do

sum = sum + sdata(i,1)

return sum/n

End

Start(find_max(sdata, n))

max = sdata(0,1)

for i = 1 to n-1 do

if (sdata(i, 1) > max) max = sdata(i,1)

return max

End

Start(find_rank(sdata, n, sid))

for i = 0 to n-1 do

if (sid = sdata(i, 0))

jumsu = sdata(i, 1)

break

endif

endfor

rank = 0;

for i = 0 to n-1 do

if(sdata(i, 1) > jumpsu)

rank = rank +1

endfor

return rank+1

End

	0	1
0	717	88
1	625	92
2	810	80
3	707	75
4	530	98
5	424	70
6	877	65
7	701	85
8	628	70
9	505	78

C함수 구현 연습(2)

```
#define NUM 30
double find_avg(int sdata[][2], int n);
int find_max(int sdata[][2], int n);
int find_rank(int sdata[][2], int n, int sid);
main()
{
    int i, scores[NUM][2], sid, n;
    printf("Enter n :");
    scanf("%d", &n);
    printf("Input id and score : \n");
    for (i=0; i < n; i++)
        scanf("%d %d", &scores[i][0], &scores[i][1]);

    printf("평균 = %.2f\n", find_avg(scores,n));
    printf("최대값 = %d\n", find_max(scores, n));

    printf("Input student id to be search : ");
    scanf("%d", &sid);
    printf("%d번의 등수 : %d\n", sid, find_rank(scores, n, sid));
}
```

	0	1
0	717	88
1	625	92
2	810	80
3	707	75
4	530	98
5	424	70
6	877	65
7	701	85
8	628	70
9	505	78

C함수 구현 연습(2)

```
double find_avg(int sdata[][2], int n)
{
    int i;
    double avg=0;
    for (i=0; i < n; i++)
        avg += sdata[i][1];
    avg = avg / n;
    return avg;
}
```

```
int find_max(int sdata[][2], int n)
{
    int i, max;
    max = sdata[0][1];
    for (i=1; i < n; i++)
        if (sdata[i][1] > max)
            max = sdata[i][1];
    return max;
}
```

```
int find_rank(int sdata[][2], int n, int sid)
{
    int i, jumsu, rank;

    for (i=0; i < n; i++)
        if (sid == sdata[i][0]) {
            jumsu = sdata[i][1];
            break;
        }

    rank = 0;
    for (i=0; i < n; i++)
        if (sdata[i][1] > jumsu) rank++;

    return rank+1;
}
```

	0	1
0	717	88
1	625	92
2	810	80
3	707	75
4	530	98
5	424	70
6	877	65
7	701	85
8	628	70
9	505	78