

컴퓨터를 통한 문제해결

◆ 자료구조와 알고리즘 그리고 프로그래밍 언어

Program = Data Structure + Algorithm

◆ 자료구조를 통해 문제해결의 재료를 준비

◆ 알고리즘을 통해 주어진 문제를 풀기 위한 절차나 방법을 찾아 구현

1. 문제이해를 통한 분해(divide and conquer)

2. 추상화(abstraction)

어떤 사물이나 이벤트에서 사소한 것을 제외하고 중요한 속성에서 추출하는 것

3. 규칙 발견 및 확인(rule detection & verification)

4. 일반화(generalization)

◆ 자료구조수업을 진행하기에 앞서 알고리즘 그리고 프로그래밍 언어 연습

C프로그램 연습하기(1)

```
main()
{
    float cel, fah;
    int n=0;

    printf("섭씨온도 : ");
    scanf("%f", &cel);

    while (cel > -50) {
        fah = cel * (9/5.0) + 32;
        printf("화씨온도 : %.2f\n", fah);
        n++;
        printf("섭씨온도 : ");
        scanf("%f", &cel);
    }
    printf("%d data processed . . .\n", n);
}
```

C프로그램 연습하기(2)

```
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);
}
```

C프로그램 이해하기(3)

다음은 덧셈 연습용 프로그램이다. 다음 물음에 답하시오.

```
main()
{
    int i, answer, score=0, num, correct;
    for (i=0; i<5; i++) {
        correct = 0;
        num=0;
        while (num < 3 && !correct) {
            printf("%d + %d = ", i*i, i*10);
            scanf("%d", &answer);
            if (answer == (i*i)+(i*10)) {
                printf("Correct!!\n");
                correct = 1;
                score += (20-num*3);
            }
            else
                printf("Try Again!!\n");
            num++;
        }
    }
    printf("당신의 덧셈성적은 ? %d\n", score);
}
```

(1) 위의 프로그램에서 제공되는 덧셈문제는?

(2) 한 문제에 대하여 기회는 몇 번 주어지는가?

C프로그램 이해하기(3)

다음은 덧셈 연습용 프로그램이다. 다음 물음에 답하시오.

```
main()
{
    int i, answer, score=0, num, correct;
    for (i=0; i<5; i++) {
        correct = 0;
        num=0;
        while (num < 3 && !correct) {
            printf("%d + %d = ", i*i, i*10);
            scanf("%d", &answer);
            if (answer == (i*i)+(i*10)) {
                printf("Correct!!\n");
                correct = 1;
                score += (20-num*3);
            }
            else
                printf("Try Again!!\n");
            num++;
        }
    }
    printf("당신의 덧셈성적은 ? %d\n", score);
}
```

(3) 문제를 맞힌 경우 몇 번 만에 맞혔는지 출력하려면 프로그램 어디를 수정 해야 하나?

printf("%d번째 Correct!!\n ", num+1);

(4) 5문제를 다음과 같이 맞힌 경우 위의 프로그램에 의하여 산출되는 점수를 계산해보시오.

()안에 증가되는 score값을 쓰시오.

1번 문제 : 처음에 맞힘 ()

2번 문제 : 두 번째 맞힘 ()

3번 문제 : 맞히지 못함 ()

4번 문제 : 세 번째 맞힘 ()

5번 문제 : 맞히지 못함 ()

C 함수 작성(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
```

C 함수 작성(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)**);

}

C 함수 작성 실습

```
#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);

int small(int a[], int n)
{
    int min = a[0], i;

    for (i=1; i < n; i++) {
        if (a[i] < min) min = a[i];
    }
    return min;
}
```

```
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));
}
```

0	1	2	3	4	5	6
70	50	87	98	30	60	90