

1주차 2차시 알고리즘의 기술언어

【학습목표】

1. 변수와 지정문을 설명할 수 있다.
2. 조건문, 반복문, 배열을 구분할 수 있다.

학습내용1 : 변수와 지정문

1. 변수(Variable)란

말 그대로 변할 수 있는 수를 의미하며 데이터를 저장 할 수 있는 메모리 공간에 붙여진 이름을 의미 한다.

변수의 지정 : 정수(int), 실수(float), 참거짓(boolean), 문자(char)을 사용한다.

※ 표현하는 언어 따라 타입은 조금씩 차이가 있다.

Bool형

자료형은 논리 자료형이라고도 하며, 참과 거짓을 나타내는 데 쓰인다. 주로 참은 1, 거짓은 0에 대응하나 언어마다 차이가 있다. 숫자를 쓰지 않고 참과 거짓을 나타내는 영단어 true와 false를 쓰기도 한다.

2. 변수를 만드는 규칙

- a. 변수는 영문자 순자 언더바로 구성 된다.
- b. 첫 글자 반드시 영문자 이어야 한다.
- c. 띄어쓰기(공백)을 사용 할 수 없다.
- d. 대문자와 소문자는 다른 의미 이다. (표현 하는 언어 따라 다름)
- e. 예약어는 사용할 수 없다.

3. 상수란(Constant)

상수(Constant)란 수식에서 변하지 않는 값을 뜻한다. 이것은 변하는 값 변수와 반대이다.

예) $A=100$ 에서 A는 변수를 의미 하며 100은 상수를 의미 한다.

$A=100$ 컴퓨터의 주기억장치의 A번지(변수 A)에 100을 저장(대입) 시켜라.

$B=200$ 컴퓨터의 주기억장치의 B번지(변수 B)에 200을 저장(대입) 시켜라.

$C=A+B$ 컴퓨터의 주기억장치 A번지에 기억된 100과 B번지에 기억된 200을 인출하여 더한 후 C번지에 저장(대입) 시켜라.

수식	순서도 표기	의사 코드
A=100	A = 100	A ← 100;
B=200	B = 200	B ← 200;
C=A+B	C = A + B	C ← A + B;

순서도 표기 의미 코드

학습내용2 : 반복문과 조건문

1. 반복문

반복문은 for문과 while 문이 사용 된다.

1) for문

- for문은 반복문의 일종으로 가장 많이 사용 되며 반복되는 부분을 제어하기 위해 초기값, 조건식, 증감값 등의 세부분으로 구성된다.

형식)

```
for (초기값; 조건식; 증감값)
    명령문;
```

```
main(){
int i, s;
s=0;
for (i=1; i<=100; i++)
    s=s+i;
printf("1+2+... +99+100=%d\n", s);
}
```

간단한 예제

2) while 문

- while 문의 조건식을 비교하여 참이면 해당 명령문을 수행하고 다시 조건식을 비교하여 참이면 계속적으로 해당 명령문을 계속 수행한다. 반복 수행 도중 조건식이 거짓이면 while문을 벗어난다. 만일 처음부터 조건식이 거짓이면 해당문장을 수행하지 않고 while문을 빠져나온다.

형식

```
while(조건식);  
{  
    명령문;  
}
```

```
int main(void)  
{  
    int num=0;  
    while(num<=5);  
    {  
        printf("Hello Word !");  
        num++;  
    }  
    . . .  
    return 0  
}
```

간단한 예제

2. 조건문

1) if 문

조건문은 조건을 제시하고 그 조건이 참이면 조건문 안에 문장이 수행되고 아니면 다음문장을 수행한다.

형식)

```
if(조건식)  
    명령문1;  
else  
    명령문2;
```

예1)

```
main(){
int a;
scanf("%d",&a);
if(a>0)
    printf("양수");
else
    printf("음수");
}
```

간단한 예제

예2)

```
main(){
char hak;
int kor, eng, mat, tot;
float ave;
scanf("%d %d %d",&kor, &eng, &mat);
tot = kor + eng + mat;
ave = tot / 3;
if (ave >= 90) then
    hak = 'A';
else if(ave >= 80)
    hak = 'b';
else if(ave >= 70)
    hak = 'c';
}
```

간단한 예제

학습내용3 : 배열

1. 배열의 정의

배열이란 같은 데이터 형태의 구조화된 데이터 형 을 갖고 있는 기억 장소들을 모아 놓은 기억 장소들의 집합체이다. 메모리 내에서 연속적인 기억 장소들을 확보하며 각각의 기억 장소들을 배열 요소라 하고 앞에 붙여진 이름을 배열명이라 한다. 배열의 크기는 대괄호([])로 쓴다.

2. 일차원배열

형식

형식 배열명[];

* 예문 : int ab[10];

- 배열명은 ab이고 ab라는 이름의 정수형 기억장소가 10개 확보가 된다.

ab[0]	ab[1]	ab[2]	ab[3]	ab[4]	ab[5]	ab[6]	ab[7]	ab[8]	ab[9]
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

- 언어에 따라 1부터 할 수 도 있다.

ab[1]	ab[2]	ab[3]	ab[4]	ab[5]	ab[6]	ab[7]	ab[8]	ab[9]	ab[10]
-------	-------	-------	-------	-------	-------	-------	-------	-------	--------

* float kkk[5]

- 배열명은 kkk이고 kkk라는 이름의 실수형 기억 장소가 5개 확보가 된다.

kkk[0]	kkk[1]	kkk[2]	kkk[3]	kkk[4]
--------	--------	--------	--------	--------

- 언어에 따라 1부터 할 수 도 있다.

kkk[1]	kkk[2]	kkk[3]	kkk[4]	kkk[5]
--------	--------	--------	--------	--------

* 예문 : 배열의 간단한 예제

```
main(){
int ab[12]={31,28,31,30,31,30,31,31,30,31,30,31};
int a, dd;
dd=0;
for (a=0; a<12; a++)
    dd += ab[a];
printf("일년은 총 %d 일입니다. \n", dd);
printf("원하는 달은? ");
scanf("%d", &a);
printf("%d월은 %d일입니다" \n", a, ab[a-1]
}
```

3. 다차원 배열

배열은 첨자의 형태에 따라 2차원, 3차원 등의 다차원 배열을 선언할 수 있다.

* 예문

```
char bb[2][3];
```

```
int cc[3][3];
```

```
float kk[2][3][4];
```

```
char bb[2][3] = {'a','b','c'},{'d','e','f'};
```

a	b	c
d	e	f

```
int cc[3][4] = {{1,2,3},{4,5,6},{7,8,9}}
```

1	2	3
4	5	6
7	8	9

와 같이 기억 된다.

4. 함수

각종 프로그래밍 언어 들은 다양한 함수로 구성 된다. 함수란 데이터가 주어지면 이를 처리하여 사용자가 원하는 결과를 얻게 한다. 이렇게 특정 기능을 처리하기 위한 명령들을 묶어 놓은 단위를 함수라고 하는데 함수에는 메이커 측에서 제공하는 표준 라이브러리 함수와 사용자가 필요에 의해서 직접 만들어 쓰는 사용자 정의 함수가 있다.

1) 표준 라이브러리 함수

키보드 입력 scanf 함수

출력하기위한 printf 함수

2) 사용자 정의 함수

작은 프로그램은 main 함수 하나로 충분이 작성할 수 있지만 프로그램이 커지면 main 함수가 너무 커지게 되므로 프로그램을 기능별로 적당히 분할해야 한다. 프로그램을 기능별로 나누어서 모듈화를 하면 프로그램을 작성하기도 쉬워지고 수정도 용이 하게 된다. 프로그램을 기능별로 모듈화 하기 위해 함수가 사용되며 함수는 필요한 기능을 처리할 수 있도록 별도로 만들어 놓을 수 있다. 이를 사용자 정의 함수라 하며 필요시 적절히 호출하여 사용할 수 있다.

예제 프로그램

```
#include<stdio.h>
void hap(int n)
{
    int i;
    int sum;
    sum=0;
    for (i=1; i<=n; i++)
        sum=sum+i;
    printf("1 ~ %d = %d \\\n", n, sum);
}
void main()
{
    hap(10);
    hap(20);
    hap(30);
    printf("\\n\\n");
}
```

함수의 간단한 예제

위의 프로그램은 1~10, 1~20, 1~30의 합 을 구하는 프로그램인데 hap(30) 아래 줄에 hap(40); hap(50); 을 추가 하면 간단히 1~40, 1~50의 합 을 쉽게 구할 수 있게 된다.

① 함수를 정의하는 이유

- a. 모듈화에 의한 기능별 분류로 프로그램의 체계화.
- b. 유지보수 및 확장성의 용이성.
- c. 문제의 해결성 및 분활성.

【학습정리】

1. 변수와 지정문

1) 변수(Variable)란

말 그대로 변할 수 있는 수를 의미하며 데이터를 저장 할 수 있는 메모리 공간에 붙여진 이름을 의미 한다.

변수의 지정 : 정수(int), 실수(float), 참거짓(boolean), 문자(char)을 사용한다.

※ 표현하는 언어 따라 타입은 조금씩 차이가 있다.

2. 상수란(Constant)

상수(Constant)란 수식에서 변하지 않는 값을 뜻한다. 이것은 변하는 값 변수와 반대이다.

예) $A=100$ 에서 A는 변수를 의미 하며 100은 상수를 의미 한다.

$A=100$ 컴퓨터의 주기억장치의 A번지(변수 A)에 100을 저장(대입) 시켜라.

$B=200$ 컴퓨터의 주기억장치의 B번지(변수 B)에 200을 저장(대입) 시켜라.

$C=A+B$ 컴퓨터의 주기억장치 A번지에 기억된 100과 B번지에 기억된 200을 인출하여 더한 후 C번지에 저장(대입) 시켜라.

3. 반복문

반복문은 for문과 while 문이 사용 된다.

4. 조건문

1) if 문

조건문은 조건을 제시하고 그 조건이 참이면 조건문 안에 문장이 수행되고 아니면 다음문장을 수행한다.

5.배열

배열이란 같은 데이터 형태의 구조화된 데이터 형 을 갖고 있는 기억 장소들을 모아 놓은 기억 장소들의 집합체이다.

메모리 내에서 연속적인 기억 장소들을 확보하며 각각의 기억 장소들을 배열 요소라 하고 앞에 붙여진 이름을 배열명이라 한다. 배열의 크기는 대괄호([])로 쓴다. 배열에는 1차원배열과, 첨자의 형태에 따라 2차원, 3차원 등의 다차원 배열을 선언할 수 있다.

6. 함수

각종 프로그래밍언어 들은 다양한 함수로 구성 된다. 함수란 데이터가 주어진다면 이를 처리하여 사용자가 원하는 결과를 얻게 한다. 이렇게 특정 기능을 처리하기 위한 명령들을 묶어 놓은 단위를 함수라고 하는데 함수에는 메이커 측에서 제공하는 표준 라이브러리 함수와 사용자가 필요에 의해서 직접 만들어 쓰는 사용자 정의 함수가 있다.