

학습목표

- ▶ 라이브러리 함수에 대해 알 수 있다.
- ▶ 재귀 함수에 대해 알 수 있다.



EBS tip

1 라이브러리 함수

- ① C 언어 컴파일러에서 사용자가 많이 필요로 하는 것을 함수로 제공하는데 이를 라이브러리 함수 또는 표준 함수라 한다.
- ② 라이브러리 함수는 함수명과 괄호 안의 인수로 구성된다.

형식 : 함수명(인수)

③ 수치 라이브러리 함수의 종류

| 관련 헤더 | 함수명(인수) | 의미 |
|----------|-----------------------|--------|
| math.h | double sin(double x) | sin(x) |
| math.h | double cos(double x) | cos(x) |
| math.h | double tan(double x) | tan(x) |
| math.h | double sqrt(double x) | 제곱근 |
| math.h | int abs(int x) | 절대값 |
| stdlib.h | int rand(int x) | 난수 발생 |

예) 절대값을 구하는 프로그램(입력값 : -5)

| 프로그램 소스 코드 | 실행 결과 |
|--|------------|
| <pre>#include <stdio.h> #include <math.h> void main(){ int a, b; scanf("%d", &a); b=abs(a); printf("%d의 절대값은=%d", a, b); }</pre> | -5의 절대값은=5 |

[프로그램 해설]

- 절대값을 구하는 수치 함수를 사용하기 위해 math.h 헤더 파일을 포함시킨다.
- scanf() 문으로 값을 입력받아 abs() 함수에 입력하면 절대값을 구한 결과가 출력된다.

④ 난수 발생 프로그램

- C 언어에서는 난수 발생을 위해 rand() 함수와 srand() 함수를 제공한다.
- rand() 함수는 난수를 발생시키고, srand() 함수는 초기화를 한다.
- 난수를 발생하기 위해서는 <stdlib.h> 파일을 포함해야 한다.

라이브러리 함수를 사용하기 위해서는 헤더파일을 반드시 포함시켜야 한다.

- 절대값과 동일한 역할을 수행하는 사용자 정의 함수 정의

```
int fun_abs(int x){
    if ( x > 0 )
        return x;
    else
        return x * (-1);
}
```

예 1부터 10까지 범위의 난수 20개를 발생하는 프로그램

| 프로그램 소스 코드 | 실행 결과 |
|---|--|
| <pre>#include <stdio.h> #include <stdlib.h> #include <time.h> void main(){ int m, num, cnt; srand((unsigned)time(NULL)); cnt = 0; for(m = 1; m <= 20; m++){ num = (rand() % 10) + 1; printf("%3d", num); cnt++; if (cnt % 5 == 0) printf("\n"); } }</pre> | <pre>6 1 2 5 10 9 7 6 8 1 9 2 4 1 1 4 1 10 1 7</pre> |

[프로그램 해설]

- `srand((unsigned)time(NULL))`를 이용하여 난수 발생 계열을 초기화한다. 발생 계열을 초기화하지 않으면 계속해서 동일한 난수가 발생될 수 있다.
- `rand()`에 의해 생성된 값을 10으로 나눈 나머지는 0부터 9까지의 숫자에 해당하므로 1부터 10까지의 난수를 발생하기 위해서는 발생된 값에 1을 더한다.
- 그리고 난수가 발생할 때마다 `cnt` 값을 1씩 더하여 난수가 발생된 수를 누적한다. 또, 발생된 난수를 한 줄에 5개씩 인쇄하기 위해 `cnt` 값을 5로 나누어 나머지가 0이면 다음 줄로 이동하기 위해 `printf("\n");` 문을 사용하였다.
- 따라서 한 줄에 5개씩 20개의 난수를 발생하여 출력한다.

⑤ 문자열 라이브러리 함수

| 관련 헤더 | 함수명(인수) | 의미 |
|----------|--------------|-----------|
| string.h | strlen(x) | 문자열 x의 길이 |
| string.h | strcat(a, b) | 두 문자열의 접속 |
| string.h | strcmp(a, b) | 두 문자열의 비교 |
| string.h | strcpy(a, b) | 문자열의 복사 |
| stdio.h | gets(x) | 문자열 입력 함수 |
| stdio.h | puts(x) | 문자열 출력 함수 |



EBS tip

- `rand()` : 정수형의 난수값을 반환한다.
- `srand(seed)` : seed 값에 따라 난수값을 초기화한다.
- 실행할 때마다 다른 난수가 발생하도록 하기 위해 `srand`의 seed 값으로 항상 바뀌는 `time` 함수를 이용한다.
- `time` 함수를 사용하기 위해서는 프로그램에 `<time.h>` 헤더 파일을 포함한다.



예 문자열 함수를 활용한 프로그램

| 프로그램 소스 코드 | 실행 결과 |
|---|-------------------------|
| <pre>#include <stdio.h> #include <string.h> void main(){ char s[]="ALGORITHMS"; int cnt, a; a = strlen(s); printf("%2d\n", a); for(cnt = 0; cnt <= a; cnt++){ if ((cnt % 2) == 0) printf("%c ", s[cnt]); } }</pre> | <pre>10 A G R T M</pre> |

[프로그램 해설]

- strlen() 함수를 사용하기 위해 <string.h> 헤더 파일을 include한다.
- strlen(s) 함수는 문자열 처리 함수 중에 하나로 문자열 s의 길이를 리턴한다.
- 따라서 문자열 길이를 저장하는 변수 a = 10이 된다.
- cnt를 2로 나누었을 때 나머지가 0인 경우의 첨자에 해당하는 문자만 출력한다.

- C 언어에서 문자열 길이는 \0(null)을 제외한 문자의 개수가 된다.

2 재귀 함수

- ① 재귀 함수란 자기 자신으로부터 호출되는 함수를 말한다.
- ② 자기 자신을 호출하는것을 재귀 호출(recursive call)이라고 한다.
- ③ 재귀 호출 함수는 재귀 호출이 종료되는 조건을 반드시 설정해야 한다.

예 재귀 호출을 이용한 1부터 n까지의 합을 구하는 프로그램

| 프로그램 소스 코드 | 실행 결과 |
|--|---|
| <pre>#include <stdio.h> int sum(int n); void main(){ int num; printf("Inpu Number =>"); scanf("%d", &num); printf("sum of 1 - %d : %d\n", num, sum(num)); } int sum(int n){ if (n == 1) return 1; else return n + sum(n-1); }</pre> | <pre>Inpu Number =>10 sum of 1 - 10 : 55</pre> |

[프로그램 해설]

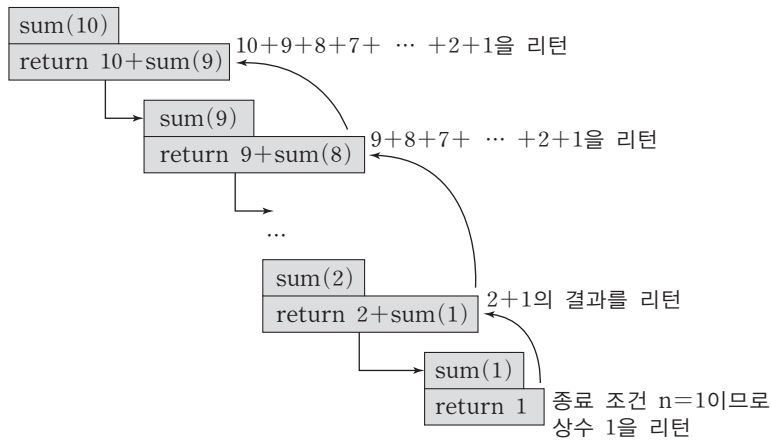
- 1부터 n까지의 합을 구하는 과정을 재귀적으로 분석하면, $\text{sum}(n)=n+\text{sum}(n-1)$ 이 된다. 예를 들어, $n=10$ 인 경우 $\text{sum}(10)=10+\text{sum}(9)$ 와 같이 구할 수 있다.
- 재귀 호출이 종료되는 조건을 제시하기 위해 $n=1$ 일 때 $\text{sum}(1)=1$ 이 되도록 설정한다.

● 반복문을 이용한 1부터 n까지의 합을 구하는 사용자 정의 함수

```
int sum(int n){
    int k, tot = 0;
    for (k = 1; k <= n; k++){
        tot = tot + k;
    }
    return tot;
}
```



- sum 함수에서 $n + \text{sum}(n-1)$ 의 과정을 통해 자기 자신을 호출하므로 재귀 함수이다.
- 실행 과정은 다음과 같이 나타낼 수 있다.



3 변수의 영역 규칙(scope rules)

변수는 선언된 위치에 따라 변수의 수명과 유효 범위가 다르다.

(1) 지역 변수

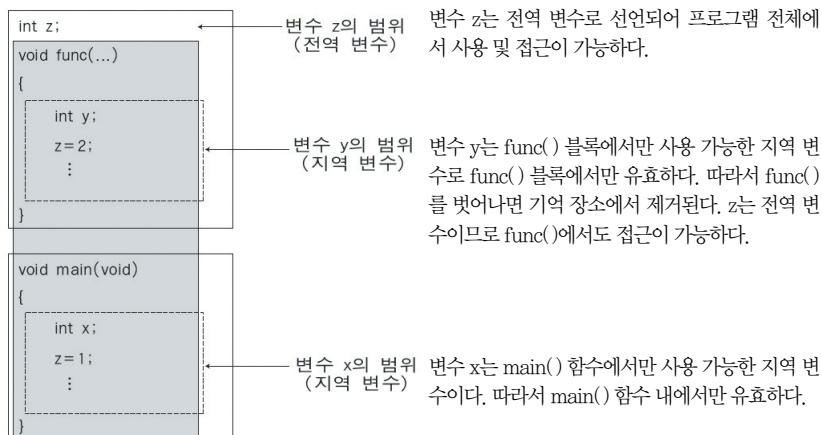
- ① 지역(local) 변수는 하나의 블록에서만 정의되어 사용되는 변수이다.
- ② 지역 변수와 전역 변수의 이름이 같으면 지역 변수가 유효하다.
- ③ 중첩된 블록의 지역 변수들의 이름이 같으면 가장 안쪽에 정의된 지역 변수가 유효하다.

(2) 전역 변수

- ① 전역(global) 변수는 함수 밖에서 선언하여 전체 프로그램에서 사용할 수 있는 변수이다.
- ② 전역 변수는 많은 함수에서 사용하는 공용성이 높은 변수에 주로 사용한다.

(3) 프로그램에서의 사용 예

- ① 전역 변수로 선언된 변수 : 선언 위치부터 프로그램 끝까지 유효하다.
- ② 함수 내에 지역 변수로 선언된 변수 : 함수 내부에서 유효하다.



예 지역 변수와 전역 변수

| 프로그램 소스 코드 | 실행 결과 |
|--|--------|
| <pre>#include <stdio.h> int a, b, c; int fun(int b, int c){ a = a + 10; b = b + 10; c = c + 10; return a; } void main(void){ int a, b, c; a = 3; b = 5; c = 7; a = fun(b, c); printf("%2d %2d %2d\n", a, b, c); }</pre> | 10 5 7 |

[프로그램 해설]

- 함수 외부의 변수 a, b, c는 전역 변수로 선언되었다. 따라서 프로그램 전체에서 접근 및 사용이 가능하다.
- main() 함수에서 변수 a, b, c는 지역 변수로 선언하여 사용된다. 따라서 전역 변수와 이름이 동일하지만, 선언된 지역 내에서만 사용할 수 있다.
- fun()에서 선언된 변수 b, c는 fun() 내에서만 유효하다. 즉 변수 b와 c의 값이 함수 fun() 내에서 바뀌어도 main() 함수에서 b와 c의 값에는 영향을 주지 않는다.
- 따라서 출력값은 전달받은 10이 main() 함수의 a에 저장되어 10 5 7이 된다.



EBS tip

- 지역 변수와 전역 변수의 이름이 같으면 지역 변수가 유효하다.
- 프로그램 작성시 가급적 지역 변수명과 전역 변수명은 다르게 설정하는 것이 바람직하다.



기출 모의고사

정답 및 해설 p. 15

2009학년도 대수능

[1~2] 다음은 자료를 관리하는 프로그램이다. 물음에 답하시오.

```
#include <stdio.h>
int k[10];
int Fr, Re;
void fun_P(int m) {
    k[Re] = m;
    Re = Re + 1;
}
int fun_G() {
    int b;
    b = k[Fr];
    Fr = Fr + 1;
    return b;
}
void main() {
    int a;
    Fr = 0; Re = 0;
    fun_P(7);
    fun_P(5);
    a = fun_G();
    fun_P(a);
    fun_P(3);
    a = Fr;
    while(a != Re) {
        printf(" %d", k[a]);
        a = a + 1;
    }
}
```

1 위 프로그램에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

보기

- ㄱ. fun_P()는 반환값이 없다.
- ㄴ. fun_G()는 정수형의 연산 결과를 반환한다.
- ㄷ. 변수 Fr은 fun_G()에서 사용되는 지역 변수이다.

- ① ㄱ ② ㄷ ③ ㄱ, ㄴ
- ④ ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ

2 위 프로그램의 실행 결과는?

- ① 3 5 7 ② 5 7 3
- ③ 7 5 3 ④ 7 5 5 3
- ⑤ 7 5 7 3

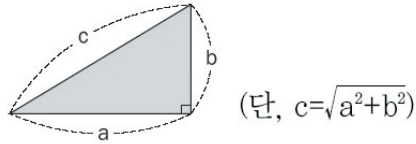
3 입력값이 5일 때 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
int r(int x);
void main(){
    int a,s;
    scanf("%d", &a);
    s=r(a);
    printf("%d", s);
}

int r(int x){
    if (x==1)
        return 1;
    else
        return r(x-1)+x;
}
```

- ① 5 ② 10 ③ 15
- ④ 20 ⑤ 25

- 4 다음은 밑변 a , 높이 b 를 입력받아 직각 삼각형의 빗변 c 를 구하는 프로그램을 작성하려고 한다. 이에 대한 설명으로 옳지 않은 것은?



```
#include <stdio.h>
#include <math.h>
void main() {
    변수 선언 부분 (가)
    scanf("%f,%f", &a, &b);
    c=a*a+b*b;
    d=sqrt(c); (나)
    printf("%6.2f", d);
}
```

- ① (가)의 변수 선언은 실수형이어야 한다.
- ② 자료를 입력받기 위해 함수를 사용하였다.
- ③ 변수 c 의 값이 4.0이라면 d 의 값은 16.0이다.
- ④ (나)에서 사용한 함수를 라이브러리 함수라 한다.
- ⑤ (나)의 함수가 포함된 헤더 파일을 전처리기에 포함시켜야 한다.

- 5 다음의 프로그램의 실행 결과는?

```
#include <stdio.h>
#include <string.h>
void main(){
    char s[10] = "KOREAART";
    int cnt, n;

    n = strlen(s);
    for(cnt = 0; cnt <= n; cnt++){
        if (cnt % 2 == 1)
            printf("%c", s[cnt]);
    }
}
```

- ① KOREA ② KRAR ③ OEAA
- ④ OEAT ⑤ REAA

- 6 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main(){
    int k;
    scanf("%d", &k);

    if (k < 0)
        k=k*(-1);
    printf("d\n", k);
}
```

- ① exp() ② int() ③ log()
- ④ abs() ⑤ sqrt()