

사용자 정의 함수

[1] 사용자 정의 함수의 개요

- 사용자가 직접 만들어 사용하는 함수이다
- C언어에서 함수 이름은 해당 함수가 시작되는 주소를 의미

[2] C문제

재귀 함수를 이용해 팩토리얼 연산을 수행하는 C프로그램이다.
실행 결과를 확인하시오

```
#include <stdio.h>
int factorial(int n);

main() {
    int (*pf)(int);
    pf = factorial;
    printf("%d", pf(3));
}

int factorial(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}
```

풀이

모든 C프로그램은 반드시 main() 함수에서 시작

```
#include <stdio.h>
int factorial(int n);

main() {
    int (*pf)(int); // 정수형 함수 포인터 pf를 선언
    pf = factorial; // factorial 함수의 시작 주소를 함수 포인터 pf에 저장
    printf("%d", pf(3)); // factorial(int 3)
}

int factorial(int n) { // n = 3 , 2, 1
    if (n <= 1) // 3 <= 1 : false, 2<=1 :false, 1<=1 : true
        return 1;
    else
        return n * factorial(n - 1); // 재귀호출 ,
} // 3*factorial(2), 2*factorial(1), factoria(1) = 1
결과 : 6
```

전문가의 조언

- **프로토타입**: 함수의 형태만 미리 선언하여 컴파일러에게 사용될 함수에 대한 정보를 미리 알리는 것이라고 한다
- 사용자 정의 함수를 main() 함수 이전에 정의하면 별도로 선언하지 않아도 됩니다
- main() 함수 이전에 다른 함수가 있으면 해당 함수를 인식하기는 하지만
- 실행은 main() 함수를 찾아서 거기서부터 실행한다
- 아래의 예제를 확인하자~~!!

```
#include <stdio.h>
int factorial(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}

main() {
    int (*pf)(int);
    pf = factorial;
    printf("%d", pf(3));
}
결과 : 6
```

참고

```
#include <stdio.h>

main() {
    int (*pf)(int);
    pf = factorial; // 오류 발생!!
    printf("%d", pf(3));
}

int factorial(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}

error: 'factorial' undeclared (first use in this function)
```

기출 따라잡기 문제[1]

```
#include <stdio.h>
void align(int a[]) {
```

```

    int temp;
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4 - i; j++)
            if (a[j] > a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}
main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    for (int i = 0; i < 5; i++)
        printf("%d ", a[i]);
}

```

풀이

```

#include <stdio.h>
void align(int a[]) {
    int temp;
    for (int i = 0; i < 4; i++) // i=0, 1, 2, 3
        for (int j = 0; j < 4 - i; j++) // j<4, j<3, j<2, j<1
            if (a[j] > a[j+1]) {
                temp = a[j]; // 자리교환
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}
main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a); // 함수 호출
    for (int i = 0; i < 5; i++) // i=0, 1, 2, 3, 4
        printf("%d ", a[i]); // a[0] a[1] a[2] a[3] a[4]
}

```

기출 따라잡기 문제[2]

```

#include <stdio.h>
int r1() {
    return 4;
}
int r10() {
    return (30 + r1());
}
int r100() {
    return (200 + r10());
}
int main() {

```

```
    printf("%d\n", r100());
    return 0;
}
```

풀이

```
#include <stdio.h>
int r1() { // 4
    return 4;
}
int r10() { // 34
    return (30 + r1()); // 30 + r1() => 30 + 4 = 34
}
int r100() { // 234
    return (200 + r10()); // 200 + r10() => 200 + 34 = 234
}
int main() {
    printf("%d\n", r100()); // 234
    return 0;
}
결과 : 234
```

기출 따라잡기 문제[3]

```
#include <stdio.h>
char n[30];
char* getname() {
    printf("이름 입력 : ");
    gets(n);
    return n;
}

main() {
    char* n1 = getname();
    char* n2 = getname();
    char* n3 = getname();
    printf("%s\n", n1);
    printf("%s\n", n2);
    printf("%s\n", n3);
}
```

풀이

```
#include <stdio.h>
char n[30];
char* getname() {
    printf("이름 입력 : ");
```

```

    gets(n); // 문자열 입력 함수
    return n; // 입력값 출력
}

main() {
    char* n1 = getname(); // 입력받은 문자열 n1변수에 저장
    char* n2 = getname(); // 입력받은 문자열 n2변수에 저장
    char* n3 = getname(); // 입력받은 문자열 n3변수에 저장
    printf("%s\n", n1);
    printf("%s\n", n2);
    printf("%s\n", n3);
}

```

기출 따라잡기 문제[4]

5를 입력했을 때 그 실행 결과를 쓰시오

```

#include <stdio.h>
int func(int a) {
    if (a <= 1) return 1;
    return a * func(a - 1);
}

int main() {
    int a;
    scanf("%d", &a);
    printf("%d", func(a));
}

```

풀이

```

#include <stdio.h>
int func(int a) { // func(5)
    if (a <= 1) return 1; // 5<=1 , 4<=1, 3<=1, 2<=1, 1<=1(true)
    return a * func(a - 1); // 재귀호출, 5*func(4),
} // 5*func(4), 4*func(3), 3*func(2), 2*func(1)
// 2*1=2 , 3*2=6, 4*6=24, 5*24=120

int main() {
    int a;
    //scanf("%d", &a); // 입력함수 a변수에 저장
    printf("%d", func(5)); // func(5) , 120
}
결과 : 120

```

기출 따라잡기 문제[5]

```

#include <stdio.h>
int isPrime(int number) {
    for (int i = 2; i < number; i++)
        if (number % i == 0) return 0;
    return 1;
}

int main() {
    int number = 13195;
    int max_div = 0;
    for (int i = 2; i < number; i++)
        if (isPrime(i) == 1 && number % i == 0) max_div = i;
    printf("%d", max_div);
}

```

약수
나머지 0

풀이

- 소수 : 1과 자기 자신만을 약수로 가지는 수

```

#include <stdio.h>
// 소수이면 1을 출력, 소수가 아니면 0을 출력
int isPrime(int number) {
    for (int i = 2; i < number; i++) // i=2,3,4,
        if (number % i == 0) return 0; // i가 number의 약수
    // true이면 반복문을 빠져나오고 0을 출력
    return 1; // 1을 출력
}

int main() {
    int number = 13195;
    int max_div = 0;
    for (int i = 2; i < number; i++) // i=2,3,4,5,..., 13194
        if (isPrime(i) == 1 && number % i == 0) max_div = i;
    // isPrime(i) == 1 true : 소수
    // number % i == 0 : 13195의 약수
    printf("%d", max_div); // 2가지 조건을 만족하는 값의 최대값 출력
}

```

결과 : 29

Handwritten calculation for 13195:

$$\begin{array}{r}
 5 \overline{) 13195} \\
 \underline{13} \\
 7 \\
 \underline{70} \\
 203 \\
 \underline{203} \\
 0
 \end{array}$$

29
 $5 \times 7 \times 13 \times 29$

기출 따라잡기 문제[6]

```

#include <stdio.h>
main() {
    int res = mp(2, 10);
    printf("%d", res);
}

int mp(int base, int exp) {
    int res = 1;
}

```

```
    for (int i = 0; i < exp; i++)  
        res *= base;  
    return res;  
}
```

풀이

```
#include <stdio.h>  
main() {  
    int res = mp(2, 10); // mp()함수호출  
    printf("%d", res);  
}  
int mp(int base, int exp) { // base=2, exp=10  
    int res = 1;  
    for (int i = 0; i < exp; i++) // i=0,1,2,3,4,5,...,9  
        res *= base; // res = res * base;  
    // 1 * 2 = 2, 2 * 2 = 4, 4 * 2 = 8,...  
    return res;  
}  
결과 : 1024
```