

# 2022 C언어 기초

정보처리기사 실기 대비(개편) – 배점 약 10점

C언어: 1972년 미국 벨 연구소, 데니스 리치에 의해 개발된 시스템 기술용 언어/논리적, 구조적, 간략한 문법 표현

C언어->	<div>01 C언어의 기초문법구조</div> <div>02 변수와 상수</div> <div>03 기억클래스</div> <div>04 연산자</div> <div>05 함수</div> <div>06 제어문_선택구조</div> <div>07 제어문_반복구조</div>	<div>08 배열</div> <div>09 포인터 응용</div> <div>10 구조체</div> <div>11 공용체</div> <div>12 전처리기</div> <div>13 메모리 동적 할당과 해제</div> <div>14 파일입출력</div>	범위->	<div>01 C언어의 기초문법구조</div> <div>02 변수와 상수</div> <div>04 연산자</div> <div>06 제어문_선택구조</div> <div>07 제어문_반복구조</div> <div>08 배열</div> <div>09 포인터 응용</div>
-------	---	--	------	--

## ▶ 프로그래밍 개발 단계(번역 단계)

프로그램 목적 정의 및 프로그램 설계 > 소스코드 작성 > 컴파일 링킹 로딩 > 프로그램 실행 > 테스트와 디버깅 > 유지보수

C언어 소스코드	컴파일	C언어 결과값	.obj	목적파일
.C	= 링킹 =>	.exe	.h	라이브러리 함수
문법	로딩	0, 1		

## ▶ C 언어 프로그램의 기본 구조(~.C) : 행 번호가 없고 대소문자를 구분한다.

도입 부분: 프로그램설명 주석 / 전처리기(매크로) / 사용자정의 함수 선언문 / 사용자정의 자료형 선언(구조체)

Main()함수: . exe가 실행되면 Main()함수가 먼저 실행됨 / 1개必

변수선언문(육각형,준비기호)과 실행문(직사각형,마름모꼴)으로 보통 나뉨

호출 함수: 선언문의 내용 부분

## ▶ C 프로그램의 구성 요소

예약어 reserved word:	키워드 자료형 관련, 기억 관련, 제어 관련, 기타 int, char, float, static, if~else, for, while, include
명칭 identifier:	식별자 변수명, 배열명, 함수명, 매크로
상수 constant:	정수 상수, 실수 상수, 문자 상수, 문자열 상수
연산자 operator:	45개의 연산자
설명문 comments:	주석 비실행문, //한줄 주석 /* 여러 줄 주석 */

## ▶ C 기초문법구조와 printf()함수

C언어 함수 중심 프로그래밍: main() 함수, 표준 함수(h헤더파일, 링킹, 인클루드), 사용자정의 함수

main() 함수: 프로그램 실행의 시작과 끝, 반드시 1개 있어야 한다.

printf() 함수: f은 format , #include <stdio.h> 속에 있다.

stdio.h: 표준입출력 헤더파일, printf()를 사용하기 위해 링킹

양식: void main() 헤더, { } 바디

%d	십진수 decimal	25	정수형
%f	부동소수점(실수) floating point	3.14	실수형
%c	문자상수 char	'A' 따옴표	문자형
%s	문자열상수 string	"PASS" 쌍따옴표	문자열형
%o: 8진, %x: 16진, %u: 부호없는 10진, %e: 지수			

%d + %d, 1, 2 인수 확장

## ▶ 변수Variable 와 상수Constant

C프로그램: 자료(상수) + 명령어 // 생성->저장->처리

자료인 상수를 메모리 영역인 변수에 담아야 명령어에서 처리할 수 있다.

변수 = 메모리 계획

상수 Constant:

항상 고정된 값을 갖는 자료, 변경 불가능  
정수형(10진수, 8진수, 16진수), 실수형(소수형식, 지수형식),  
문자형(키보드문자+escape문자, ASCII코드값으로 저장됨),  
문자열형

일반	123(10)	123(8)	123(16)
C	123	0123	0x123

%c와 %s:

%s(문자열상수)에는 문자열 끝에 \0 추가(null값)

ex) 'A'는 1byte, "A"는 2byte (+ \0)

변수 variable:

프로그램 실행 중, 변경할 수 있는 값이 저장되는 기억공간(=메모리)

변수의 4요소:

(시작)주소(Address), 이름(Name), 자료형(Data Type), 값(Value)

#### ▶ 변수의 선언, 대입, 초기화

변수에 값 넣기:

변수 선언	int A;	육각형
변수 대입	A=10	직사각형
초기화(선언과 대입을 동시수행)	int A=10;	육각형

변수의 선언문:

변수명name과 자료형datatype을 결정하여, 기억공간을 할당(계획, 준비)하는 것 (컴파일러가)

자료형 data type

정수형	<b>int(4)</b> , short, long, unsigned
실수형	<b>float(4)</b> , <b>double(8)</b> , long double
문자형	<b>char(1)</b> , unsigned char
열거형	enum
형 없음	void

변수명의 정의 규칙:

영어 대소문자 구분, 숫자 ok(숫자 시작X), 밑줄 ok

변수의 대입문:

변수(기억공간, address의 memory)에 자료(값, value)을 대입하는 것

=는 대입연산자, L-value와 R-value 있음

A = A+1 ➡ 누적

#### ▶ 변수와 scanf()

scanf()함수:

콘솔화면에서 키보드로부터 자료(상수)를 주어진 입력 양식으로 입력시키는 함수

#include <stdio.h>에 있음 // scanf("%d", &입력대상1, &입력대상2)

&는 주소연산자, 주소를 찾아감~ 미리 저장할 공간(변수 선언문) 필요

#### ▶ 연산자 operator

풍부한 연산자를 제공 -> 3항연산자, 비트연산자(6개)

연산 대상(피연산자)로 구분: 1차 연산자(단항): > 2차 연산자(이항): > 3차 연산자(삼항): > 대입연산자 > 나열연산자  
, 는 나열연산자

연산자 우선순위: 단항( [] ! & )>산술( \* / + - )>관계( > >= == )>논리( && || )>삼항( ? : )>대입( += )>나열( , )  
++prefix(전위), postfix--(후위)

논리부정 연산자: 논리부정 NOT, ! -> 논리곱 AND, && -> 논리합 OR, || // 출력 byte단위: 참1 거짓0

삼항연산자: 항1(조건식) ? 항2(참) : 항3(거짓)

비트 연산자(6개): & | ^ ~ << >>

기타 연산자(대부분 단항): sizeof() case & \*

sizeof(): 괄호 안의 있는 자료형의 크기를 byte단위 메모리 크기로 보여줌

☞ 형변환 연산자(cast): // 정수와 정수 연산 => 정수 / 실수와 실수 연산 => 실수

소숫점(실수) 결과 얻기: 변수 앞 괄호 안에 변형할 형 넣음

ex) "%f", (double)x / (double)y

#### ▶ 제어문 – 선택구조

C언어: 구조화된 프로그래밍 structured program

블록 제어구조{, 순차/선택/반복 구조 >> 순차가 기본, 중첩됨

제어구조

순차 제어구조 Sequence	int a; a = 10+20	변수 선언문, 변수 대입문
선택 제어구조 select	if문, switch~case문	
반복 제어구조 looping	while문(반복 0회 ↑), do~while문(반복 1회 ↑), for문	
제어명령문	break; continue; goto 레이블;	

중괄호 스코프: 블록 구조라고 함. ex) C, Java.... / 제어구조: 순서도에서는 화살표 ->

if문:                    ① 단순 if문:            실행대상(명령문)이 한 개  
                           ② if~else:            if 조건식으로 참과 거짓(else)를 구분해서 진입, 실행대상(명령문) 2개 => 3행 비교  
                           ③ 중첩 if            if, else if, else if, else if..... => switch~case문과 비교할 수 있다.(switch가 적절)

switch~case문        C언어는 행번호가 없고, 코드의 위치를 레이블로 나타낸다.  
                           switch( 변수 또는 상수 ){ case: ~; break; .... default: ~; }

#### ▶ 제어문 - 반복구조

유한반복과 무한반복: break;여부 또는 조건문  
                           증감 등으로 count하는 변수: 반복 제어 변수  
                           반복문 위에서 제어변수 초기화; 반복문 내부에 제어변수 증감식;

무한반복문 사용하기:  
                           while문    ⇨ while(1){ 무한반복대상 }  
                           for문       ⇨ for( ; ; ){ 무한반복대상 }  
                           ~~~> 필요할 때 if문 사용, break;

반복문 반복수행대상에서 sum + i;에서 i의 역할:    ① 반복횟수, 제어변수 ② 누적항

⇨ goto 레이블명:    무조건 분기, 구조적 포함관계와 상관없이 이동 / 일부 상황에서 사용함  
                           START: 여기 시작/여기 도착  
                           ~~~~~ ? goto END : goto START;  
                           END: 여기 도착

#### ▶ 배열 Array

사용자정의 자료형(확장자료형)을 제공한다.(배열, 구조체) ≠기본자료형  
 한 번의 선언으로 여러 개의 메모리 공간 확보(연속적 나열)  
 동질형자료형의 연속적 모임 ,

배열 변수 선언문:    int num[3];                    // 순서도에서는 대괄호 또는 소괄호 사용  
                           메모리 공간을 연속적으로 확보: 번지수 연속으로~~  
                           형식: 자료형 배열명[개수]; // 개수는 정수상수  
 배열의 초기화:        배열 요소의 범위: 배열명[0] ~ 배열명[첨자-1]  
                           첨자 = index, 0부터 시작  
                           선언과 동시에 초기화=>요소의 개수 생략 // int a[]={1, 2, 3};

float avg; avg = (int+int)/2 는 정수값 ((float)int+int)/2 캐스팅, 형변환 printf("%.2f", avg) 소수 이하 2번째 자리
int pil[5]; float avg; int i, total=0;                    // 변수 ii의 역할: 반복제어, 배열첨자 for(i=0; i<5; ii++){            // 변수 total: 누적 총계(반드시 초기화) scanf("%d", &pil[i]); total+=pil[i] } avg = total / 5.0    // 5.0이 실수니까 total이 float로 형변환되서 나눠짐

문자열 배열:        str[5] = "PASS" // 한 개, 잘라져서 들어감 [P, A, S, S, \0]  
                           \0 : escape문자, Null문자, 문자열의 끝을 나타냄, 1byte이다.  
                           문자열배열의 길이, sizeof(str) == 5이다.  
                           printf("%s\n", str); %s는 Null이 나올 때까지 출력하고 멈춤  
                           str는 &str[0] 축약 // =why?=> 배열명은 배열의 첫 요소[0]의 주소&다.  
 2차원 배열:        2차원은 논리적구조(표), 배열명[행 개수][열 개수] // 초기화 같이 할 때 s[][n] 가능

▶ 포인터 pointer 변수

메모리 내 스택영역 내 변수를 pointing함 / 주소 참조 / 포인터 변수를 사용한다.

포인터 Pointer:      특정한 데이터가 저장된 주소값(시작 주소)을 저장하는 변수  
                          하나의 주소값은 1바이트 크기의 메모리공간을 표현함

포인터 변수 선언문: 자료형\* 포인터 변수명; // int \*p1(에디터) == int\* p1(시험용) 포인터형 자료형 변수

포인터 변수Variable 선언과 대입:

포인터자료형 변수	기본자료형 변수
int* p;           (4byte) p=&num; <주소지칭	int num;       (4byte) num=10;

&	주소 연산자	모든 변수에 대한 주소값을 구하는 연산자
*	포인터 연산자, (간접 연산자, 내용 연산자)	only 포인터 변수의 자료(내용)을 구하는 연산자

```
int num; num=100;
int* p1; p1=&num;                        // 포인터 변수가 int*형
printf("일반변수 %d\n", num);
printf("포인터변수 %d\n", *p1);   <- 포인터 변수가 가리키는 *내용
```

포인터변수와 배열: 배열의 이름은 배열의 첫 요소의 주소임 => 배열 이름을 포인터변수에 입력 => 배열 첫 요소를 포인팅

```
int i; int A[]={1, 2, 3, 4, 5};
int* p; p = A;    // A == &A[0]
for(ii=0; ii<5; ii++){
    printf("%5d", *(p+i))   // A[i]를 포인터변수를 기준으로 주소로 접근해서 표현
}
```

포인터변수와 연산: p=변지, \*p=내용 / (\*p)++;