**C프로그래밍 최종정리**

**변수**

**문자** :   
char : 1byte 문자

**정수**  
short : 2byte   
int : 4byte  
long : 4byte  
long long : 8byte

스택의 첫 값은 sign값으로 이용되어, 음수, 양수를 나타낸다.  
때문에, 4바이트의 부호를 가지는 정수형 변수의 경우, -2^15 ~ 2^15-1의 범위를 가진다.

char에는 문자 하나입력만됨. ㅎㅎ  
여러문자 입력하고싶으면 string 헤더 들고와서 쓰자

printf : printformatted  
%형식지정자  
%c : character. 문자.  
%d : Decimal. 십진정수.  
%f : Float. 실수.  
%x : hexadecimal. 16진수.

증감연산자 전위 후위

**타입 변환 연산자 : cast연산자**

암시적 타입 변환 (암시적 형변환)

타입 변환 연산자가 사용되지 않고 자동으로 타입 변환이 됨.

Ex. 정수 + 실수 = 실수  
 정수 + 정수 연산은 당연히 정수값이 나오기에,   
 출력시 %f와 같은 형식지정자를 사용하려 한다면 주의해야 한다.

타입변환을  
(int)a 식으로 할 수 있음. 이름뭔지 까먹음(수정할 것)

**연산자의 우선순위**

1. 괄호()[]
2. ++, --, +부호, -부호, !, ~, &(참조), \*(포인터), sizeof
3. \*(곱하기), /, %
4. +(더하기), -(빼기)
5. <<, >> 쉬프트연산자
6. <, <=, >=, >
7. ==, !=
8. &(비트곱)
9. ^(비트차)
10. |(비트합)
11. %%
12. ||
13. ?:(삼항연산)
14. =, +=, -=, \*=, /=, %=, <<=, >>=, &=, ^=, |= 잡다한놈들

**제어문 : 선택문**

If, switch –

**If**

다들 아는대로   
if ( 조건문; )  
 명령문; //이런식으로, 바로 아랫줄에는 보통 중괄호 없이도 작동하긴함.

근데

If ( 조건문; ) {  
 명령문;  
 } //이게 정석. 굳이 겉멋부린다고 중괄호 없이 코딩하는건   
 진짜 욕 얻어먹고 오래 살기 좋은 행동 1위로 유명하다.  
 (협업 시 가독성 하락, 유지보수 난이도 상승. 꽝이다.)

여러 명령문을 써야 하면 당연히 여러 개 써서 복합문 만들어야 함.

If( 0 ) 는 false, if( 1 ) 은 true로 여겨진다.   
(항상 false / true. If( 0 ) 는 명령문 존재자체가 무의미함.)

**If~else**

조건식 이후 else로 이외 상황에서의 명령문을 추가함.

If ( )   
{ }  
else   
{ }

**switch 이어서**

switch( ) 괄호 속 조건식은 정숫값만 허용된다.   
정수이며, 상수여야 한다.

**break**

switch문 강제로 빠져나오게 함.

**반복문**

변수를 조건식에 사용하며, 특정 횟수가 지정된 경우 : for문을 주로 사용  
특정 조건 만족까지 반복 실행하는 경우 : while / do while 주로 사용.

**For문**

For( 초기식; 조건식; 반복식 ) { 명령문; }  
초기식 : 처음 시작될 때 실행됨. (주로 초기화를 위해 사용됨.)  
조건식 : 만족하면 for문 빠져나옴.  
반복식 : 명령문 실행 이후 실행. 매 반복마다 실행.

**배열**

자료형 배열명[배열의 길이];  
 배열의 길이는 7 >> 0~6까지의 7개의 자리가 생김.

**인덱스**

요소의 순번.

위에서 말한 ‘배열의 길이’가 인덱스의 예시.  
배열 요소가 N개라면 마지막 요소의 인덱스는 N-1로 설정됨.

배열 이름, 첨자[] 연산자, 인덱스를 이용해 개별 요소 접근가능. Arr[0] 이런거.

초기화는 int Arr[10] = { 0, 1, 2, 3, 4, 5, 6, 7, …. , N번째 요소 } 이런식으로 진행.   
N번째 요소는 10-1로, 9번째 요소가 되어야 할 것.

문자열은 당연하게도 { ‘a’, ‘b’, ‘c’, ‘d’ }와 같이 초기화 해 줘야함.

**2차원 배열**

자료형 배열명[배열 요소 개수][배열 요소 속 요소의 개수];

그냥 int Arr[3][3] 이런거임. 이러면 3x3의 배열 구조가 생김. 실제로 메모리 구조가 그러한 격자 모양인 건 아니지만, 그렇게 연상하면 편함.

텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

실제로는 선형 구조를 가짐.

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

바이트마다 주소가 부여되어 있음.

이외에 다차원 배열이라고, Arr[3][3][3][3][3][3][3] 그냥 맘대로 배열 만들 수 있다. 그냥 7-4그림 아래에 배열이 더 추가되는 것 뿐이다.

**배열의 크기와 길이**

배열의 크기 : sizeof(배열)  
 : sizeof(요소의 타입) \* 요소의 개수

배열 메모리 구조를 받아오거나, 해당 자료형을 받아와 개수를 곱해주는 방식을 사용하여 크기를 구할 수 있다.

배열의 길이 : sizeof(배열) / sizeof(요소의 타입)

배열의 크기를 자료형 타입으로 나누면, 메모리의 선형 구조인 배열을 메모리 단위로 나눌 수 있는 것이기에, 배열의 길이를 구할 수 있다.

**strlen함수**

strlen함수로 문자열의 길이를 쉽게 받아올 수 있다.

str arr[ ] = “어쩌구저쩌구”;  
int len = strlen(arr);

이런식이다.

arr[ ] 처럼 배열 크기를 지정하지 않은 문자열은, 초기화값에 따라 자동으로 맞춰진다.

하지만, MBCS(멀티 바이트 캐릭터 셋), 두개 이상의 바이트를 사용해 하나의 문자를 표현하는 문자 인코딩 방식을 사용하는 문자열은 곧바로 strlen함수를 사용하여 길이를 구할 수 없다.

따라서 문자 배열에서 MBCS문자만 골라내 따로 출력하거나, 유니코드로 변환하여 출력하여야 한다.

**가변 길이 배열**

배열의 길이에 변수를 이용하는 방식이다.

Int arr[N] =

이런 방식.

오류를 일으키기 쉬워 비주얼 스튜디오에서는 의도적으로 지원하지 않음.

가변 배열 크기는 1024 \* 1024 \* 4, 즉 2의 22승까지 가능함.