

C언어기초개념 정리

메모리의 기본 : 비트

→ 8비트 = 1바이트 → ^{다변히 더는 단위}

정수형 : int 형 (32비트)

→ CPU에 따라 int형의 비트수 바뀔수 있음

[정수 → 음수, 0, 양수]

부호있는 signed 정수형 (-, 0, +)

부호없는 Unsigned 정수형 (0, +) → ex) 속도'와 같이 음수가 존재하지 않는 것들

오버플로우 : 변수가 나타낼수 있는 범위 넘었을 때의 오류

언더플로우 : 변수가 나타낼수 있는 범위보다 작을 때의 오류

보수관계 (반대는 의미로 이해)

1의 보수 : 비트간의 합이 1이 되도록.
2의 보수 : 음수 표현 (1의 보수 + 1)

(ex) 0 1 1 0 1의 보수
1 0 0 1
1 0 0 0 2의 보수

문자형 : 글자, 숫자, 기호 등 ...

나 공통규격 : 아스키코드 (ASCII)
for Information Interchange

제어문자 (ctrl = \)

\a : 경고 벨 : 뽀-소리
\b : 백스페이스 : 앞으로 한칸씩 이동
\t : 수평탭 : 커서를 다음 탭 위치로!
\n : 줄바꿈 : 엔터 나
\r : 커다름표 : 출력하기
\f : 작은따옴표 : 출력하기
\r : 수직탭 : 커서 다음 수직탭 이동
\ : 역슬래시 : 출력하기

니가 젤 헛갈려하는거 → #include <stdio.h>

기본개념 : %d : 10진수
%o : 8진수
%x : 16진수

정수일 때 %3d = "3" 전체 자릿수
--0, -13

연산자

대입 : = : 대입
산술 : + - * / % : 사칙연산
관계 : > < <= >= : 비교

논리 : && (AND) : 둘다 참이어야 참
|| (OR) : 둘중 하나라도 참이면 참

! (NOT) : 반대 (0이 아니면 1, 1이 아니면 0)
ex) (_ ? _) → 틀

조건 : ? : if 조건문과 같은 맥락
(a < b) 참 : 거짓
조건문 : 3진마법칙
참 : 출력, 거짓 : 출력

코드
#include <stdio.h>
int main(void)
{
char code1 = 'A'; // 문자 상수로 초기화
char code2 = 65; // 아스키 코드로 초기화
printf("code1 = %d\n", code1);
printf("code2 = %c\n", code2);
}

출력결과
code1 = 65
code2 = A

standard input/output
3. 헤더 파일 선언
#include <stdio.h>
int main()
{
char code = A;
printf("%d %d %d\n", code, code+1, code+2); // 출력값 : 65 66 67
printf("%c %c %c\n", code, code+1, code+2); // 출력값 : A B C
return 0;
}

복합대입연산자

x += y → x = x + y (다 같은 맥락)

조건연산자 (ex)

1) 절댓값 계산 : (x > 0) ? x : -x;
2) 최댓값 계산 : (x > y) ? x : y;
3) 최솟값 계산 : (x < y) ? x : y;

비트연산자

& : AND : 둘다 1이면 1, 아니면 0
| : OR : 둘중 하나라도 1이면 1, 아니면 0
^ : XOR : 둘이 같으면 0, 다르면 1
<< (>>) : 비트를 왼쪽 (오른쪽) 으로 이동
~ : NOT : 0은 1로, 1은 0으로.

왼쪽 이동 : 원래 값의 2배 / 오른쪽 이동 : 1/2 배

예시로 이해하자

```
int x=10, y=10;
printf("x=%d\n", x); // x=10
printf("x+y의 값=%d\n", x+y); // x+y의 값=20
printf("x=%d\n", x); // x=10

printf("y=%d\n", y); // y=10
printf("y+y의 값=%d\n", y+y); // y+y의 값=20
printf("y=%d\n", y); // y=10
return 0;
```

증감 후치 : ++, --
선치 : ++, --

● 진법 별로 사용되는 기호의 종류 (범위)

16진수

● 진수 : ●개의 기호로 데이터를 표현해야 한다.

* 2진수 : 0 → 1 → 10 → 11 → 100 → 101 → 110 → 111 → ...

10진수

2진수

0123456789 ABCDEF

● 정수의 가장 왼쪽에 존재하는 비트 : "부호비트" (+, -) ⇒ 0: 양수 / 1: 음수

● "n개의 비트로 표현할 수 있는 데이터의 수는 2^n개이다"

ex) 11100 → 2^4 + 2^3 + 2^2

01001 → 2^4 + 2^3

● Switch 문 예시

#include <stdio.h>

int main (void)

```
{
    int num;
    printf("1이상5이하정수입력:");
    scanf_s("%d", &num);
```

switch (num)

```
{
    case 1 = (num=1)
```

case 10 → 땀땀땀이

printf("1은 ONE 인가");

break;

case 2:

switch (

default:

printf("I don't know in");

}

return 0;

}

● 조건문

if (조건식 1)

문장 1;

else if (조건식 2)

문장 2;

else

문장 3;

● break 문 쓰는 예시 코드

#include <stdio.h>

int main (void)

```
{
    int i;
```

int sum=0;

for(i=1; i<=100; i++)

```
{
    sum+=i;
```

if(sum>100)

{ break;

}

printf("합은 %d", sum);

return 0;

}

→ sum 값을 계속 증가시키고 있는데 sum이 100이 되면 더 이상 증가하지 않게 break 문을 써서 빠져나온다.

상호교체가능

for 문 (위에 대해 반복하라)	while 문 (아래는 동안 반복하라)
<ul style="list-style-type: none"> 정해진 횟수 반복. 1씩 증가시키면서 1부터 100까지 더하라 사용자로부터 숫자를 20번 입력받고 평균 구하라 50번 '*' 출력해라 	<ul style="list-style-type: none"> 무한반복하기 반복횟수 고려하기 싫을 때 (내부적으로 결정됨) 사용자가 그만하길 할 때 까지 입력받아 평균 구하라. 1부터 순차적으로 값이 증가 할 때 50 될 때까지 반복하라.

⇒ 필요시 언제든지 빠져나올 수 있음. (by break;)

" 건너뛸 수 있음. (by continue;)

● for 문 (정해진 횟수 반복)

들) for (i=0; i<5; i++)

```
{
    초기식 조건식 종감식
    printf("Hello World\n");
}
```

초기식
↑
종감식
↑
(i의 값을 1씩 증가시키면서
⇒ i의 값이 4가 될 때까지
"Hello World" 출력해라.)

● 참고 예시

1) for (printf("반복시작"), i=1; i<=5; i++)

// 콤마 연산자로 문장 넣어도 가능

2) for (i=1; i<=5 && p>3; i++)

// 복합수식도 가능

3) for (i=1; sum=0; i<=5; i++)

// 다른 변수도 초기화 → 초기식이 2개

4) i=1; // 변수 선언을 (초기화) 따로 함

for (; i<=5; i++) // 초기식 x

5) for (;;) // 무한반복

• for문 ~ continue; 쓰기

#include <stdio.h>

```
int main(void)
{
    int sum=0;
    for (int i=2; i<=10; i=i+2)
    {
        sum += i;
    }
    printf("偶数 1~10의 합 = %d\n", sum);
    return 0;
}
```

(=)

#include <stdio.h>

```
int main(void)
{
    int sum=0;
    for (int i=1; i<=10; i++)
    {
        if (i % 2 == 0)
        {
            sum += i;
        }
        else // (i % 2 != 0) 이면 아무것도 안하고 처음으로 돌아간다
        {
            continue;
        }
    }
    printf("偶数 1~10의 합 = %d\n", sum);
    return 0;
}
```

● 중첩 for문 (반복문 안에 또 반복문)

Ex) #include <stdio.h>

→ 출력결과

```
*****
*****
*****
*****
*****
```

```
int main(void)
{
    int x, y;
    for (y=0; y<5; y++)
    {
        for (x=0; x<10; x++)
        {
            printf("#")
        }
        printf("\n")
    }
    return 0;
}
```

여기부터 확인

● do ~ while 문 (예시)

```
#include <stdio.h>
int main(void)
{
    int i=0;
    do {
        printf("%d\n", i);
        i = i+1;
    } while (i<3);
}
```

⇒ i가 3보다 작을때까지만 실행한다.

● while문

int i=0; (초기식)

while (i<5) → 조건식

반복내용

printf("Hello World! \n");

i++; (증감식)

이거 없으면 무한루프 발생

● while문 예시

```
#include <stdio.h>
int main(void)
{
    int i, n, sum; // 변수선언
    printf("정수 입력: ");
    scanf("%d", &n);
    i = 1;
    sum = 0;
    while (i <= n)
    {
        sum += i;
        i++;
    }
    printf("1~%d의 합 = %d\n", n, sum);
    return 0;
}
```

※ 조건식에 원만하면 < / = < 이거나 쓰기
오류 → (!=) × (==) ×

● while (i) = while (i != 0)

이 0이 아닌 모든 값은 참으로 간주됨

● for문 예시

```
#include <stdio.h>
int main(void)
{
    int x, y;
    for (y=1; y<=5; y++)
    {
        for (x=0; x<10; x++)
        {
            printf("#")
        }
        printf("\n")
    }
    return 0;
}
```

중첩 for문 사용

● while문 (소문자 → 대문자)

```
#include <stdio.h>
int main(void)
{
    char letter;
    while (1)
    {
        printf("소문자를 입력하세요: ");
        scanf(" %c", &letter);
        if (letter == '0')
            break; // 대문자 Q가 입력되면 반복 종료
        if (letter < 'a' || letter > 'z')
            continue; // 소문자가 아니면 즉시 다음 반복 시작
        letter -= 32; // 소문자 → 대문자
        printf("변환된 대문자 = %c\n", letter);
    }
    return 0;
}
```

※ 205P
※ 206P (최대 입력값)
※ 207P (한글기)
※ 208P (숫자 입력)
※ 209P (06-11)