

C Camp

Software
Department



Contents



전 시간 복습

반복문 loop

-while loop

-for loop

Mind



이해 하려 들지 말고, 먼저 **관찰** 하자.

복습



C programming을 한다 함은..

함수를 디자인 하는 것..

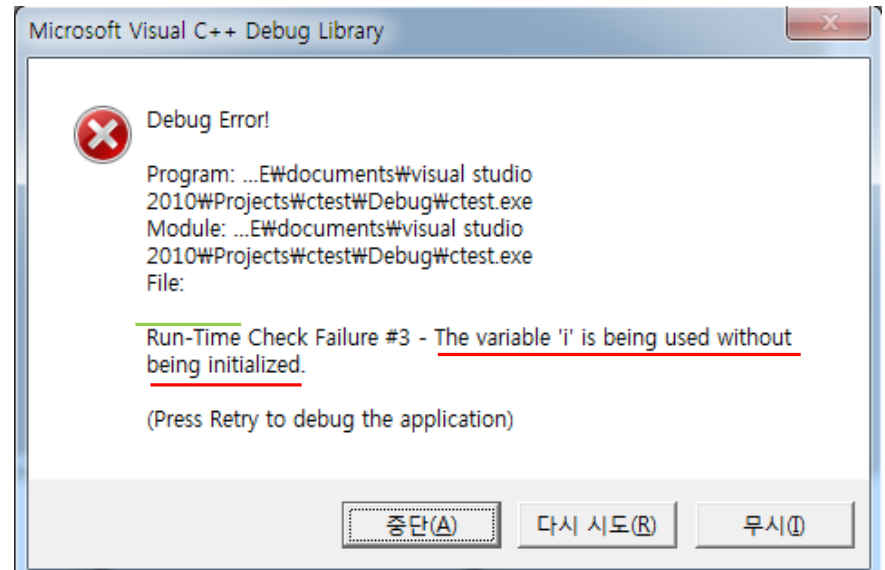
우리는 이미 main이라는 함수를
만들어 왔다!

```
void main ( )  
{  
  
}
```

복습



```
int i;  
printf( "%d" , i);
```



쓰레기 값 1324709813274198473928471

오해하지 마세요



```
int num = 5;
```

```
printf( "num" );
```

```
-> printf( "num : %d \n" , num);
```

```
printf( " abcd num 1234 32490  ");
```

오해하지 마세요



int num = 10; //메모리에 2진수로 저장.

char ch = 'a' ;

printf("10진수로 : %d \n" , num);

printf("8진수로 : %o \n" , num);

Printf("16진수로: %x \n" , num);

복습



연산자는 값을 **반환**한다.

```
int num1 , num2 ;
```

```
num2 = num1 = 1 + 2;
```

//이 문장이 가능한 이유!

풀어서 쓰면.

```
num2 = num1 = 3;
```

```
num2 = 3;
```

```
num2 = 3;
```

//1 + 2의 연산 결과 3을 반환

// num1 = 3 연산 결과 3을 반환

//num2 = 3 연산 결과 3을 반환하고

//더 이상 저장 할 곳 없으니 끝.

복습



형 변환

- 자동 형 변환

-대입 연산 시 양 쪽의 데이터 형이 같지 않다면
대입 대상을 기준으로 변환한다.

ex) floatNum = 1 + 2 ;

intNum = 2.2 + 1.1 ;

-연산 시 데이터의 손실이 예상된다면, 손실이 안 나는 쪽으로
형 변환

ex) result = intNum * floatNum

복습



형 변환

- 강제 형 변환

-float + float 의 연산 결과가 float인 것과 마찬가지로,
int 와 int 의 결과는 int형이다..

때문에 int 나누기 int 의 결과는 int.

때문에 우리의 예상과는 달리. 몫만 나온다.
(소수점 이하는 버려짐)

ex) printf("%f" , 3/2); //0.00000이 나온다. (int를 %f로 출력..)
printf("%d" , 3/2); //1이 나옴을 확인 할 수 있다.

해결책-> printf("%f" , (float)3/2);
printf("%f" , 3.0 / 2); //float 과 int의 연산이 이루어지려 하기때문
printf("%f" , 3.0 / 2.0); //데이터의 손실을 막기 위해 나머지
//int하나까지 자동으로 형 변환이 이루어져
//원하는 결과가 나온다.

loop



"Hello world" 를 5줄 출력하려면..?

(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
}
```

loop



컴퓨터는 멍청하지만,
단순 반복은 최고예요...

while loop

(전역 범위)

```
#include <stdio.h>

void main(){
    int num = 0;

    while ( num < 5 ){
        printf("hello world ! \n" );
        num = num + 1; //이 줄을 num += 1; 로 바꿔보세요
                      //num++; 로도 바꿔 보세요
    }
}
```

(전역 범위)

```
#include <stdio.h>

void main(){

    printf("hello world ! \n");
    printf("hello world ! \n");
    printf("hello world ! \n");
    printf("hello world ! \n");
    printf("hello world ! \n");

}
```

while loop



```
while ( 조건 ){  
    //조건이 참일 때  
    //실행 되는 영역!  
}
```

cf) if..else if..if

if문처럼, 반복의 대상이 하나라면.. {}생략 가능

while loop

(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    int num = 0;
```

```
    while ( num < 5 ){
```

```
        printf("hello world ! \n" );
```

```
        num = num + 1; //이 줄을 num += 1; 로 바꿔보세요
```

```
        //num++; 로도 바꿔 보세요
```

```
    }
```

```
}
```

반복을 위한 변수의 초기화

반복 조건 검사

(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
    printf("hello world ! \n");
```

```
}
```

반복의 조건을 거짓으로
만들게 하기 위한
증가 및 감소 연산

※루프 탈출? 조건이 없다면..?

즉, 반복의 조건을 무너뜨리기 위한
연산이 없다면..?

Practice



문제 1.

- (1) 1부터 10까지 출력하세요.
- (2) 1부터 10까지의 합을 출력하세요

문제 2.

구구단 2단을 출력하세요.

while loop



(전역 범위)

```
#include <stdio.h>

void main(){
    int num ;

    num = 0;

    while ( num < 10 ){
        printf( "%d\n", num + 1 );
        num++;
    }
}
```

(전역 범위)

```
#include <stdio.h>

void main(){
    int num = 1;
    int total = 0;

    while ( num <= 10 ){
        total = total + num ;
        num++;
    }

    printf( "total : %d \n" , total);
}
```

while loop



(전역 범위)

```
#include <stdio.h>

void main(){
    int dan, num = 1;

    printf("몇 단 ? ");
    scanf("%d", &dan );

    while ( num < 10 ){
        printf("%d x %d = %d #n" , dan , num , dan * num);
        num ++;      // num = num + 1;
    }
}
```

while loop



```
while (1) {  
    ..  
    ..  
    ..  
}
```

//무한루프

// 흐름을 break 해주는 코드가 없으면 안 끝남

//ctrl + c 로 강제 종료.

while loop



While loop의 중첩!

(전역 범위)

```
#include <stdio.h>

void main()
{
    int i = 1;
    int j = 1;

    while ( i < 4 )
    {
        while ( j < 4 )
        {
            printf("%d + %d = %d \n", i , j , (i + j) );
            j++;
        }

        j = 1;

        i++;
    }
}
```

Practice



while문을 중첩시켜서 구구단을 출력하세요
2단부터 19단까지 전부!

for loop



(전역 범위)

```
#include <stdio.h>

void main(){
    int num = 0;

    while ( num < 5 ){
        printf("hello world ! \n" );
        num = num + 1; //이 줄을 num += 1; 로 바꿔보세요
                     //num++; 로도 바꿔 보세요
    }
}
```

+2

※이 반복문이 몇 회 반복되는지 알려면?
어디어디를 봐야 하죠?

for loop



(전역 범위)

```
#include <stdio.h>

void main(){
    int i ;

    for ( i = 0 ; i < 5 ; i ++ ){
        printf("hello world \n");
    }
}
```

※이 반복문이 몇 회 반복되는지 알려면?
어디만 보면 되죠?

for loop



반복을 위한 변수의 초기화

반복 조건 검사

```
for ( 초기식 ; 조건식 ; 증감식 ) {  
    //조건이 참일 때  
    //실행 되는 영역!  
}
```

반복의 조건을 거짓으로
만들게 하기 위한
증가 및 감소 연산

for loop 's flow..



(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    int i ;
```

```
    for ( (1) i = 0 ; (2) i < 5 ; (4) i ++ ){
```

```
        (3) printf("hello world \n");
```

```
    }
```

```
}
```

(1) (2) (3) (4) i : 1
(2) (3) (4) i : 2
(2) (3) (4) i : 3
(2) (3) (4) i : 4
(2) (3) (4) i : 5
(2)

for loop vs while loop



(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    int num = 0;
```

```
    while ( num < 5 ){
```

```
        printf("hello world ! \n" );
```

```
        num = num + 1;
```

```
    }
```

```
}
```

(전역 범위)

```
#include <stdio.h>
```

```
void main(){
```

```
    int i ;
```

```
    for ( i = 0 ; i < 5 ; i ++ ){
```

```
        printf("hello world \n");
```

```
    }
```

```
}
```

Practice



문제 1.

(1) 1부터 10까지 출력하세요.

(2) 1부터 10까지의 합을 출력하세요

문제 2.

for loop을 중첩시켜서 구구단을 출력하세요
2단부터 19단까지 전부!

Thank You.

