



## 실 습 문 제

## 제2장

## 1. [컴파일 과정 실습]

본문의 예제 프로그램을 컴파일하고 실행하여서 결과를 확인한 다음에, C언어의 문법을 일부러 위반한 다음, 어떤 오류가 발생하는지를 관찰한다.

```
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

실행결과

```
Hello World!
```

(1) 비주얼 C++를 이용하여 다음과 같은 프로젝트를 만들고 프로젝트 안에 소스 파일을 추가한다. 소스 파일을 더블 클릭하여 위의 프로그램을 오류 없이 입력한다. 소스 파일의 이름을 입력할 때 확장자 .c를 붙이는 것을 잊지 말자.

- \* 프로젝트 이름: hello
- \* 프로젝트 폴더: c:\cprog\hello
- \* 소스 파일 이름: hello.c

(2) Build->Build hello.exe 메뉴를 이용하여 컴파일, 링크를 수행하여 실행 파일을 만든다. 이어서 Build->Execute hello.exe메뉴를 이용하여 실행을 시켜서 출력 화면처럼 나오는 지를 확인한다. 컴파일과 실행을 나타내는 단축 아이콘을 찾아보자.

(3) 소스 코드에서 줄과 줄 사이의 여백을 다음과 같이 없애면 어떻게 되는지를 관찰하여 보라. 빈 줄은 왜 두는 것인가?

```
/* 첫 번째 프로그램 */
#include <stdio.h>
int main(void)
{ printf("Hello World!\n"); return 0; }
```

(4) 소스 코드에서 단어 사이의 공백을 다음과 같이 없애면 어떻게 되는지를 관찰하여 보라. 공백을 하나씩 추가하면서 띄어쓰기를 할 곳을 찾아보자.

```
/* 첫번째프로그램 */
#include<stdio.h>
intmain(void){printf("Hello World!\n");return0;}
```

(5) 입력 과정에서 다음과 같은 타이핑 오류가 발생하였다면 어떤 오류 메시지가 발생하는가? 오류 메시지를 해석하여서 그 의미를 유추해보라. 오류의 위치와 오류 메시지의 위치가 일치하는가?

원래 문장	변경 문장	발생 오류
/* 첫 번째 프로그램 */	/* 첫 번째 프로그램	
/* 첫 번째 프로그램 */	첫 번째 프로그램 */	
/* 첫 번째 프로그램 */	/* 첫 번째 프로그램 */	
#include <stdio.h>	include <stdio.h>	
#include <stdio.h>	#include <stdio,h>	
int main(void)	Int main(void)	
int main(void)	int main(void);	
printf("Hello World\n");	printf("Hello World\n")	
printf("Hello World\n");	printf("Hello World\n);	
printf("Hello World\n");	print("Hello World\n");	

2. [문장의 순차적인 실행]

1번 예제를 다음과 같이 확장하여 보자.

```
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello\n");
    printf("World!\n");

    return 0;
}
```

여기에 실행 결과를 쓰시오.



- (1) 위의 프로그램을 컴파일하고 수행시켜 본다. 어떤 결과가 출력되는가? 문장 다음에 항상 ;를 붙이는 것을 잊지 말자. ;기호는 우리말의 마침표와 같은 것이다. 우리말은 마침표를 붙이지 않아도 별 문제가 없지만, C언어에서는 문장의 마침표가 없으면 컴파일러가 컴파일을 할 수가 없다.
- (2) C에서는 함수 안에 들어 있는 문장들이 순차적으로 실행된다. 다음과 같이 출력되도록 위의 프로그램을 변경하여 보자.

```
Hi!
Hello
World!
```

- (3) 프로그램 안의 문장들의 위치를 다음과 같이 변경하면 실행 결과가 어떻게 변경되는가?

```
printf("World!\n");
printf("Hello\n");
```

- (4) /\* 와 \*/ 사이에 있는 주석을 /\* 확장된 첫 번째 프로그램 \*/로 변경하여 보자. 컴파일 과정이나 수행 시에 달라지는 것이 있는가?
- (5) 주석을 다음과 같이 여러 줄로 만들어 보자. 컴파일 과정이나 수행 시에 달라지는 것이 있는가? 주석 안에 자신의 이름과 작성 날짜 등을 넣어보자.

```
/*
    확장된 첫 번째 프로그램
    작성자: 홍길동
    작성일자: 2007.3.12
*/
```

### 3. [printf() 함수]

모니터 화면에 다음과 같이 사각형 모양을 출력하는 프로그램을 작성한다. 먼저 아래 코드의 빈 칸을 채우도록 하자.

```
/* 문장들은 순차적으로 실행된다. */
#include <stdio.h>

int main(void)
{
```

```

printf("*****\n"); /* ***** 출력 */
_____ ; /* * * * 출력 */
_____ ; /* * * * 출력 */
_____ ; /* ***** 출력 */
}

```

### 실행결과

```

*****
* * *
* * *
*****

```

### 힌트

C에서는 함수 안에 들어 있는 문장들이 순차적으로 실행된다. 따라서 출력 예시에서 먼저 맨 위에 있는 \*\*\*\*\*를 출력하는 문장을 쓴 다음에, \* \* \*을 출력하는 문장을 그 아래에 쓰면 된다. 똑같은 방식으로 나머지 2개의 줄을 출력하면 된다. C에서 텍스트(문자열)를 화면에 출력하려면 printf() 함수를 이용한다. 만약 \*\*\*\*\*를 출력하고 싶으면 printf("\*\*\*\*\*\n"); 하면 된다. 여기서 \n은 \*\*\*\*\*을 출력한 다음, 커서를 다음 줄의 처음으로 보내라는 의미이다. 이후 출력은 그 위치부터 다시 시작된다.

(1) 위의 프로그램을 참조하여 다음과 같이 출력되는 프로그램도 작성하여 본다.

```

*****
* * *
* * *
* * *
*****

```

(2) 위의 프로그램을 참조하여 다음과 같이 출력되는 프로그램도 작성하여 본다.

```

*
* *
* *
* *
* *
* *
*

```



(3) `printf()`가 들어 있는 모든 문장에서 `\n`을 생략하면 실행결과가 어떻게 변화되는가?

```
printf("*****");
```

(4) `printf()`가 들어 있는 모든 문장에서 `\n`을 두 번 적어주면 실행결과가 어떻게 변화되는가? `\n`의 역할은 무엇인가?

```
printf("*****\n\n");
```

(5) 다음과 같이 코드를 변경하여 보자. 어떤 실행 결과가 얻어지는가? 어떤 결론을 유추할 수 있는가?

```
printf("*****\n*   *\n*   *\n*****\n");
```

(6) 줄을 바꾸기 위하여 다음과 같이 하면 어떻게 결과가 나오는가?

```
printf("*****
*   *
*   *
*****");
```

줄을 바꾸고자 하는 위치에서 엔터키를 눌러준다.

#### 4. [디버거의 사용]

디버거는 프로그램이 실행되는 과정을 직접 볼 수 있어서 논리적인 오류를 수정하고 프로그램 실행의 이해를 돕는데 많은 역할을 한다. 다음과 같은 실행 결과를 얻기 위하여 프로그램을 작성하였다. 여기에는 논리적인 오류가 있다. 디버거를 사용하여 논리적인 오류를 찾아내도록 하자.

실행결과

```
2 × 1 = 2
2 × 2 = 4
2 × 3 = 6
```

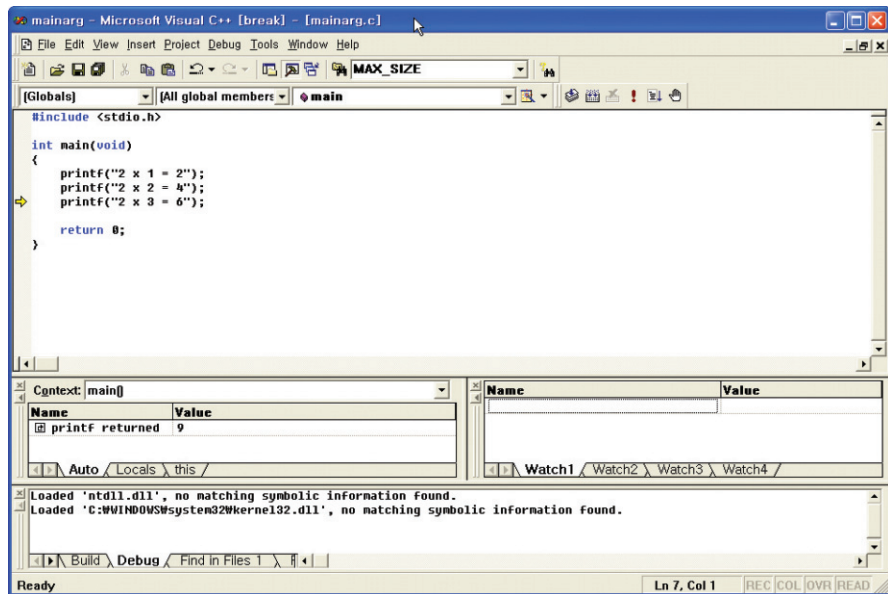
```
#include <stdio.h>

int main(void)
{
    printf("2 x 1 = 2");
    printf("2 x 2 = 4");
    printf("2 x 3 = 6");

    return 0;
}
```

- (1) 위의 프로그램을 컴파일하여 실행시켜 보자. 어떤 출력 화면이 나오는가? 실행 결과를 여기에 쓰시오.

- (2) 논리적인 오류를 찾아내기 위하여 디버거를 실행하여 보자. 비주얼 C++에서는 컴파일이 끝난 상태에서 F10 키를 누르면 한 문장씩 차례대로 실행한다. F10키를 세 번 눌러서 다음과 같은 화면이 나오도록 하라. 이 상태에서 프로그램 출력 화면을 확인해보라.



- (3) 이 프로그램에 들어있는 논리적인 오류는 무엇인가? 논리적인 오류를 수정하여 문제에서 주어진 실행화면이 나오도록 하라.