



1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
#include <stdio.h>
struct A {
    int n;
    int g;
};
main( ) {
    struct A st[2];
    for (int i = 0; i < 2; i++) {
        st[i].n = i;
        st[i].g = i + 1;
    }
    printf("%d", st[0].n + st[1].g);
}
```

$st[0].n = 0$ $st[1].n = 1$
 $st[0].g = 1$ $st[1].g = 2$

답 : 2

[해설]

```
#include <stdio.h>
struct A {
    int n;
    int g;
};
main( ) {
    ① struct A st[2];
    ② for (int i = 0; i < 2; i++) {
    ③     st[i].n = i;
    ④     st[i].g = i + 1;
    }
    ⑤ printf("%d", st[0].n + st[1].g);
}
```

구조체 A를 정의한다.

A의 멤버로 정수형 변수 n을 선언한다.

A의 멤버로 정수형 변수 g를 선언한다.

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

① A 구조체 형태로 배열 st를 선언한다.

	int n	int g
st[0]		
st[1]		

② 반복 변수 i 가 0부터 1씩 증가하면서 2보다 작은 동안 ③, ④번을 반복 수행한다.

③ $st[i].n$ 에 i 의 값을 저장한다.

④ $st[i].g$ 에 $i+1$ 의 값을 저장한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

• 1회전 ($i = 0$)

	int n	int g
st[0]	0	1
st[1]		

• 2회전 ($i = 1$)

	int n	int g
st[0]	0	1
st[1]	1	2

• i 가 2가 되면서 for문을 빠져나가 ⑤번으로 이동한다.

⑤ $0+2$ 의 결과인 2를 정수로 출력한다.

결과 2

시나공

2. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
#include <stdio.h>
main() {
    struct insa {
        char name[10];
        int age;
    } a[] = { "Kim", 28, "Lee", 38, "Park", 42, "Choi", 31 };
    struct insa* p;
    p = a;
    p++;
    printf("%s\n", p->name);
    printf("%d\n", p->age);
}
```

답 :

Lee

38

[해설]

```
#include <stdio.h>
main() {
    ① struct insa {
    ②     char name[10];
    ③     int age;
    ④ } a[] = { "Kim", 28, "Lee", 38, "Park", 42, "Choi", 31 };
    ⑤ struct insa* p;
    ⑥ p = a;
    ⑦ p++;
    ⑧ printf("%s\n", p->name);
    ⑨ printf("%d\n", p->age);
}
```

① 구조체 insa를 정의한다.

구조체의 멤버를 지정할 때는 [변수명].[멤버이름]으로 지정하지만, 포인터 변수를 이용해 구조체의 멤버를 지정할 때는 [변수명]->[멤버이름]으로 지정한다.

② insa의 멤버로 10개의 요소를 갖는 문자형 배열 name을 선언한다.

③ insa의 멤버로 정수형 변수 age를 선언한다.

④ insa 구조체의 형태로 a 배열을 선언하고 초기화한다. 배열의 크기를 지정하지 않았으므로 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

3. 다음 C 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
#include <stdio.h>
struct jsu {
    char nae[12];
    int os, db, hab, hhab;
};

int main() {
    struct jsu st[3] = { {"데이터1", 95, 88}, {"데이터2", 84, 91}, {"데이터3", 86, 75} };
    struct jsu* p;
    p = &st[0];
    (p + 1)->hab = (p + 1)->os + (p + 2)->db;
    (p + 1)->hhab = (p + 1)->hab + p->os + p->db;
    printf("%d", (p + 1)->hab + (p + 1)->hhab);
}
```

Handwritten notes:

- $st[1].hab = 84 + 75 = 159$
- $st[1].hhab = 159 + 95 + 86 = 342$
- $159 + 342 = 501$

답 : 501

[해설]

```
#include <stdio.h>
① struct jsu {
    char nae[12];
    int os, db, hab, hhab;
};

int main() {
    ② struct jsu st[3] = { {"데이터1", 95, 88}, {"데이터2", 84, 91}, {"데이터3", 86, 75} };
    ③ struct jsu* p;
    ④ p = &st[0];
    ⑤ (p + 1)->hab = (p + 1)->os + (p + 2)->db;
    ⑥ (p + 1)->hhab = (p + 1)->hab + p->os + p->db;
    ⑦ printf("%d", (p + 1)->hab + (p + 1)->hhab);
}
```

구조체 jsu를 정의한다. 구조체를 정의한다는 것은 int나 char 같은 자료형을 하나 만든다는 의미다. 구조체의 멤버를 지정할 때는 [변수명].[멤버이름]으로 지정하지만, 포인터 변수를 이용해 구조체의 멤버를 지정할 때는 [변수명]->[멤버이름]으로 지정한다.

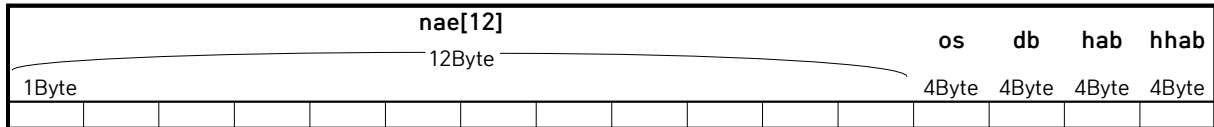
12개의 요소를 갖는 문자 배열 nae를 선언한다.

정수형 변수 os, db, hab, hhab를 선언한다.

① 구조체 jsu의 구조

	nae[0]	nae[1]	nae[2]	...	nae[11]
char nae[12]					
int os					
int db					
int hab					
int hhab					

※ 위의 구조체는 다음과 같이 메모리의 연속된 공간에 저장된 후 사용됩니다.



모든 C 프로그램은 반드시 main() 함수에서 시작한다

❶ 구조체 jsu 자료형으로 3개짜리 배열 st를 선언하고 초기화한다.

	char nae[12]	int os	int db	int hab	int hhab
st[0]	st[0].nae[0]~st[0].nae[11]	st[0].os	st[0].db	st[0].hab	st[0].hhab
st[1]	st[1].nae[0]~st[1].nae[11]	st[1].os	st[1].db	st[1].hab	st[1].hhab
st[2]	st[2].nae[0]~st[2].nae[11]	st[2].os	st[2].db	st[2].hab	st[2].hhab



	char nae[12]	int os	int db	int hab	int hhab
st[0]	'데' '이' '터' 1 '\0'	95	88		
st[1]	'데' '이' '터' 2 '\0'	84	91		
st[2]	'데' '이' '터' 3 '\0'	86	75		

※ 문자열을 저장하는 경우 문자열의 끝을 의미하는 널 문자('\0')가 추가로 저장되며, 출력 시 널 문자는 표시되지 않습니다. 또한 영문, 숫자는 1Byte, 한글은 2Byte를 차지합니다.

❷ 구조체 jsu의 포인터 변수 p를 선언한다.

❸ p에 st 배열의 첫 번째 요소의 주소를 저장한다. 주소는 임의로 정한 것이다.

p	1000	메모리															
주소	0000																
:																	
:		nae	os	db	hab	hhab	nae	os	db	hab	hhab	nae	os	db	hab	hhab	
→	1000	“데이터1”	95	88			“데이터2”	84	91			“데이터3”	86	75			
:		p				p+1				p+2							
:		&st[0]				&st[1]				&st[2]							
:																	
:	9999																

❹ p+1이 가리키는 곳의 hab에 p+1이 가리키는 곳의 os 값과 p+2가 가리키는 곳의 db 값을 더한 후 저장한다. p가 st[0]을 가리키므로 p+1은 st[1]을 p+2는 st[2]를 가리킨다. 따라서 st[1]의 os 값 84와 st[2]의 db 값 75를 더한 값 159를 st[1]의 hab에 저장한다.

메모리															
주소															
	0000														
⋮															
	nae	os	db	hab	hhab	nae	os	db	hab	hhab	nae	os	db	hab	hhab
1000	“데이터1”	95	88			“데이터2”	84	91	159		“데이터3”	86	75		
	p					p+1					p+2				
⋮	&st[0]					&st[1]					&st[2]				
9999															

❺ p+1이 가리키는 곳의 hhab에 p+1이 가리키는 곳의 hab 값과 p가 가리키는 곳의 os와 db 값을 모두 더한 후 저장한다. st[1]의 hab 값 159, st[0]의 os와 db 값 95와 88을 모두 더한 값 342를

st[1]의 hhab에 저장한다.

메모리															
주소															
0000															
⋮															
1000	nae	os	db	hab	hhab	nae	os	db	hab	hhab	nae	os	db	hab	hhab
	“데이 터 1”	95	88			“데이 터 2”	84	91	159	342	“데이 터 3”	86	75		
⋮	p					p+1					p+2				
	&st[0]					&st[1]					&st[2]				
9999															

⑥ p+1이 가리키는 곳의 멤버 hab와 hhab의 값을 더한 후 정수로 출력한다. 159와 342를 더한 501이 출력된다.

결과 501

