

ch3

1. C 언어에서의 배열에 대하여 다음 중 맞는 것은?

- (1) 3차원 이상의 배열은 불가능하다.
- (2) 배열의 이름은 포인터와 같은 역할을 한다.
- (3) 배열의 인덱스는 1에서부터 시작한다.
- (4) 선언한 다음, 실행 도중에 배열의 크기를 변경하는 것이 가능하다.

2. 다음 중 배열에 관한 문장 중 문법에 맞지 않는 것은?

- (1) `int a[]={1,2,3,4};`
- (2) `char b[10], c[20];`
- (3) `char **pa[10];`
- (4) `char *pb[10]="I am a student";`

3. `float a[100]`으로 선언된 배열의 시작주소를 1000번지라고 할때, 배열의 10 번째 요소의 주소는 몇 번지인가?

$$1000+10*\text{sizeof}(\text{float})=1040$$

- (1) 1000번지
- (2) 1010번지
- (3) 1020번지
- (4) 1040번지

4. 구조체에 관한 내용 중 틀린 것은?

- (1) 구조체의 배열이 가능하다.
- (2) 구조체 변수에 대하여 비교연산이나 산술연산을 할 수 있다.
- (3) 구조체 안에 배열을 넣을 수 있다.
- (4) 서로 다른 데이터 형도 묶을 수 있다.

5. `int i=10; int *p; p=&i; *p=8;`의 문장이 수행되면 i값은 얼마인가?

- (1) 11
- (2) 10
- (3) 9
- (4) 8

6. `int i=10; int *p; p=&i; (*p)--;`의 문장이 수행되면 i값은 얼마인가?

- (1) 11
- (2) 10
- (3) 9
- (4) 8

7. `int a[10]; int *p; p=a; *p++=5;` 의 문장이 수행되면 변경되는 배열의 요소는?

- (1) `a[0]`            (2) `a[1]`            (3) `a[2]`            (4) `a[3]`

8. `int a[10]; int *p; p=a; *++p=5;` 의 문장이 수행되면 변경되는 배열의 요소는?

- (1) `a[0]`            (2) `a[1]`            (3) `a[2]`            (4) `a[3]`

9. `int a[10]; int *p; p=a; (*p)++;` 의 문장이 수행되면 변경되는 배열의 요소는?

- (1) `a[0]`            (2) `a[1]`            (3) `a[2]`            (4) `a[3]`

10. `int a[10]; int *p; p=a; *(p+1)++;` 의 문장이 수행되면 변경되는 배열의 요소는?

- (1) `a[0]`            (2) `a[1]`            (3) `a[2]`            (4) `a[3]`

11. `int a[10]; int *p; p=a; (*(p+1))++;` 의 문장이 수행되면 변경되는 배열의 요소는?

- (1) `a[0]`            (2) `a[1]`            (3) `a[2]`            (4) `a[3]`

12. `struct { int fa; int fb; } a; int *p=&a;` 라고 할때 `fa`를 수정하는 문장으로 맞는 것은?

문제에 오류가 있습니다. 다음과 같이 수정하여 주십시오.

-> `struct { int fa; int fb; } a, *p=&a;` 라고 할때 `fa`를 수정하는 문장으로 맞는 것은?

- (1) `p.fa = 10;`  
(2) `p->a.fa = 10;`  
(3) `(*p).fa = 10;`  
(4) `p[0].fa = 10;`

13. 다음과 같은 문장을 수행하고 난 뒤의 `a[0]`의 값은?

```
void sub(int b[])  
{ b[0] = 0; }  
void main()
```

```
{ int a[]={1,2,3,4,5,6}; sub(a); }
```

(1) 0                    (2) 1                    (3) 2                    (4) 3

14. 다음과 같은 문장을 수행하고 난 뒤의 a.field1의 값은?

```
struct test { int field1; };  
void sub(struct test b)  
{ b.field1 = 1; }  
void main()  
{ struct test a={ 0 }; sub(a); }
```

(1) -1                    (2) 0                    (3) 1                    (4) 2

15. 다음과 같은 문장을 수행하고 난 뒤의 a.field1의 값은?

```
struct test { int field1; };  
void sub(struct test *b)  
{ b->field1 = 1; }
```

```
void main()  
{ struct test a={ 0 }; sub(&a); }
```

(1) -1                    (2) 0                    (3) 1                    (4) 2

16. 다음의 코드에서 잘못된 곳을 찾아라.

```
main()  
{ int i, a[10];  
  for(i=0;i<=10;i++)  
    a[i] = 0;  
}
```

->

```
main()  
{ int i, a[10];  
  for(i=0;i<10;i++)  
    a[i] = 0;  
}
```

17. 다음과 같이 정의된 배열의 10번째 요소의 주소는? 단 배열의 시작 주소는 1000번지로 한다.

- (a) char a[100];    1000+100\*sizeof(char)=1100  
(b) short b[100];    1000+100\*sizeof(short)=1200

- (c) `int c[100];`            `1000+100*sizeof(int)=1400`  
(d) `float d[100];`        `1000+100*sizeof(float)=1400`

18. 크기가 5인 정수 배열을 생성하고 {1,2,3,4,5}로 초기화하는 문장을 쓰시오.

```
int a[5]={1,2,3,4,5,6};
```

19. 구조체 배열을 만들어 보자. 정수 항목인 `number`와 문자열 항목인 `name`을 가지는 구조체를 정의하고 크기가 100인 이 구조체의 배열을 생성해보라.

```
struct {  
    int number;  
    char name[SIZE];  
} sa [100];
```

20. 10개의 정수 포인터의 배열을 정의하여 보라.

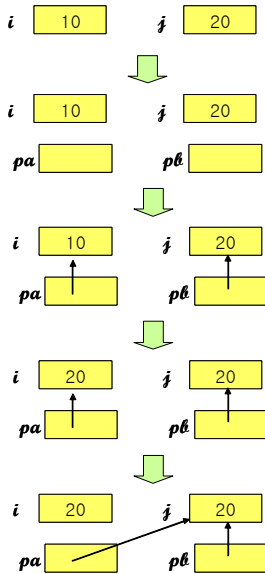
```
int *pi[10];
```

21. `i`가 정수 변수이고 `p`와 `q`가 정수를 가리키는 포인터라면 다음 중 어느 문장이 컴파일 오류를 발생할 것인지를 지적하고 그 이유를 밝히시오.

- (1) `p = &i;`  
(2) `p = *i;` -> `i`의 값이 `p`로 대입(경고)  
(3) `p = *&i;` -> `i`가 포인터가 아니므로 `*i`는 컴파일 오류  
(4) `i = *&p;`

22. 다음의 각각의 문장들을 수행한 다음에 변수들과 포인터들이 서로 연결된 모습을 그림으로 그려라.

```
void main() {  
    int i=10, j=20;  
    int *pa=&i, *pb=&j;  
    *pa = *pb;  
    pa = pb;  
}
```



23. 포인터 p1과 p2가 선언되어 있을때 다음의 문장의 의미를 구별하여 설명하여 보라.

- (1) `p1 = p2;`     $\rightarrow$ 포인터 p2가 가지고 있는 주소값을 포인터 p2에 대입
- (2) `*p1 = *p2;`  $\rightarrow$ 포인터 p2가 가리키는 값을 포인터 p1가 가리키는 곳에 대입

24. 포인터 p에 대한 연산중 p와 같은 것은?

- (1) `&*p`            (2) `*&p`            (3) `&&p`            (4) `**p`

25. 다음을 설명하여 보라.

- (1) `int **p;`                            정수의 포인터에 대한 포인터 p
- (2) `int *p[100];`                        정수 포인터 배열 p(크기 100)
- (2) `void (*f)(int);`                    정수 매개변수와 void 반환값을 가지는 함수에 대한 포인터 f

26. 배열 x를 {1,2,3,4,5,6}으로 초기화 한 후 포인터 p를 정의하고 포인터 p에 저장할 주소는 x[2]로 하고 \*(p+3)와 \*(p-2) 및 그 값의 곱을 구하는 프로그램 작성하라.

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

```
//배열 x를 {1,2,3,4,5,6}으로 초기화 한 후 포인터 p를 정의하고
```

```

// 포인터 p에 저장할 주소는 x[2]로 하고
//*(p+3)와 *(p-2)및 그 값의 곱을 구하는 프로그램을 작성하라.
main()
{
    int x[]={1,2,3,4,5,6};
    int *p=&x[2];

    printf("%d %d %d", *(p+3), *(p-2), *(p+3) * *(p-2));
}

```

27. 1개의 정수와 최대크기가 20인 문자열로 이루어진 구조체를 저장할 수 있도록 동적 메모리를 할당받고 여기에 정수 100과 문자열 "just testing"을 저장한 다음, 동적 메모리를 반환하는 프로그램을 작성하라.

```

#include <stdio.h>
#include <malloc.h>
#include <string.h>

```

```

//1개의 정수와 최대크기가 20인 문자열로 이루어진 구조체를 저장할 수 있도록
//동적 메모리를 할당받고
//여기에 정수 100과 문자열 "just testing"을 저장한 다음,
//동적 메모리를 반환하는 프로그램을 작성하라.

```

```

void main()
{
    struct test {
        int i;
        char str[20];
    };
    struct test *p;
    p=(struct test *)malloc(sizeof(struct test));
    if( p==NULL ) {
        printf("memory error");
        return;
    }
    p->i=100;
}

```

```

        strcpy(p->str,"just testing");
        free(p);
    }

```

28. C언어의 **typedef**을 이용하여 **complex**라고 하는 새로운 자료형을 정의하라. **complex** 자료형은 구조체로서 **float**형인 **real** 변수와 역시 **float**형인 **imaginary** 변수를 갖는다. **complex** 자료형으로 변수 **c1**과 **c2**를 선언하여 보라.

```

typedef struct {
    float real;
    float imaginary;
} complex;
complex c1, c2;

```

29. **person**이라는 구조체를 만들어보자. 이 구조체에는 문자배열로 된 이름, 사람의 나이를 나타내는 정수값, 각 개인의 월급을 나타내는 **float**값등이 변수로 들어가야 한다.

```

struct person {
    char name[SIZE];
    int age;
    float salary;
};

```

30. 29번의 구조체에 생년월일을 추가하고자 한다. 다음과 같은 구조체를 25번 구조체 내부에 포함시켜보라.

```

struct {
    int month;
    int day;
    int year;
};

#define SIZE 100

struct person {

```

```
char name[SIZE];
int age;
float salary;
struct {
    int month;
    int day;
    int year;
} birthday;
};
```