

PART 15-2. 함수 포인터와 void 포인터

◆ 연습문제 ◆

1. 다음과 같이 선언된 함수가 있을 때 대입 연산이 가능하도록 함수 포인터를 선언하세요.

```
double div(int, int);  
void prn(char *);  
int *sava(int);
```

- ① fp = div;
- ② fp = prn;
- ③ fp = save;

정답 : ① fp = div; -> double (*fp)(int, int);
 ② fp = prn; -> void (*fp)(char *);
 ③ fp = save; -> int *(*fp)(int);

해설 : 없음

2. 다음의 배열과 포인터가 있을 때 포인터 vp로 세 번째 배열 요소의 값 30을 출력하세요.

```
int ary[5] = {10, 20, 30, 40, 50};  
void *vp = ary;
```

정답 : printf("%d", ((int *)vp)[2]);

해설 : 없음

3. 다음 프로그램의 실행 결과를 예상해보세요.

```

#include <stdio.h>

int add(int a, int b) {
    return (a + b);
}
int sub(int a, int b) {
    return (a - b);
}
int mul(int a, int b) {
    return (a * b);
}

int main(void) {
    int (*pary[3])(int, int) = {add, sub, mul};
    int i, res = 0;

    for (i = 0; i < 3; i++) {
        res += pary[i](2, 1);
    }

    printf("%d", res);

    return 0;
}

```

정답 : 6

해설 : 없음

4. 다음의 실행 결과와 같도록 빈 칸을 채워 프로그램을 완성하세요.

```

#include <stdio.h>

int main(void) {

```

```

int a = 10, b = 20;
int *pa = &a, *pb = &b;
void *pt;
(      )
(      )
(      )
printf("%d, %d", *pa, *pb);

return 0;
}

```

실행 결과

20, 10

정답 : pt = pa;
 pa = pb;
 pb = pt;

해설 : 없음

5. main 함수가 다음과 같을 때 exchange 함수와 func 함수를 만들어 프로그램을 완성하세요. exchange 함수는 주소를 인수로 받아서 가리키는 두 값을 바꿉니다. func 함수는 인수로 받은 두 값을 출력한 후에 exchange 함수를 사용하여 두 값을 바꾸고 다시 출력합니다.

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

```

void exchange(double *, double *);
void func(void (*fp)(double *, double *), double, double);

```

```

int main(void) {
    double a = 10, b = 20;

```

```
func(exchange, a, b);

return 0;
}
```

실행 결과

```
a:10.0, b:20.0
a:20.0, b:10.0
```

정답

```
#include <stdio.h>

void exchange(double *, double *);
void func(void (*fp)(double *, double *), double, double);

int main(void) {
    double a = 10, b = 20;

    func(exchange, a, b);

    return 0;
}

void func(void (*fp)(double *, double *), double a, double b) {
    printf("a:%.1lf, b:%.1lf\n", a, b);
    fp(&a, &b);
    printf("a:%.1lf, b:%.1lf\n", a, b);
}

void exchange(double *pa, double *pb) {
    double temp;
```

```
temp = *pa;  
*pa = *pb;  
*pb = temp;  
}
```

해설 : 없음