

2023 정보처리기사 필기 정오표

[1권]

78쪽 오른쪽 단 5번 문제 보기 ①

수정 전 : ① 화면 설계 도구에는 **파워포인트**, 스토리보드, 와이어프레임, 목업 등이 있다.

수정 후 : ① 화면 설계 도구에는 **프로토타입**, 스토리보드, 와이어프레임, 목업 등이 있다.

121쪽 아래쪽 정답 줄 6번 문제 정답 수정

수정 전 : 6.②

수정 후 : 6.①

141쪽 왼쪽 단 위쪽 41번 문제 삭제

수정 전 :

41. 다음 중 가장 약한 결합도(Coupling)는?

- ① Common Coupling
- ② Content Coupling
- ③ External Coupling
- ④ Stamp Coupling

188쪽 왼쪽 단 7번 문제 해설 2번째 줄 식 부분

수정 전 : (- (/ (* A (+ B C))) D) E)

수정 후 : (- (/ (* A (+ B C)) D) E)

252쪽 왼쪽 가장자리 1번째 전문가의 조언

수정 전 : • **검증**(Validation)과 **확인**(Verification)의 개념을 묻는 문제가 출제되었습니다. **검증**은 사용자, **확인**은 개발자 중심이라는 것을 기억하세요.

수정 후 : • **확인**(Validation)과 **검증**(Verification)의 개념을 묻는 문제가 출제되었습니다. **확인**은 사용자, **검증**은 개발자 중심이라는 것을 기억하세요.

493쪽 오른쪽 단 5번 문제 보기 ②

수정 전 :

수정 후 :

학번
2020222

학번
20202222

[2권]

26쪽 기출문제 따라잡기 QR 코드 번호

수정 전 : 문제 7 2413257

수정 후 : 문제 4 2413254

187쪽 오른쪽 단 30번 문제 해설 중 아래쪽 표

수정 전 :

네트워크	호스트 수	IP 주소 범위
1	32	192.168.1.0 ~ 63
2	32	192.168.1.64 ~ 127
3	32	192.168.1.128 ~ 191
4	32	192.168.1.192 ~ 255

수정 후 :

네트워크	호스트 수	IP 주소 범위
1	64	192.168.1.0 ~ 63
2	64	192.168.1.64 ~ 127
3	64	192.168.1.128 ~ 191
4	64	192.168.1.192 ~ 255

300쪽 오른쪽 단 8번 문제 해설 수정

수정 전 : 공개키 암호화 기법에서 키의 개수는 $n(n-1)/2$ 이므로, 10명이 암호를 사용할 때 키의 개수는 $10(9)/2$ 로 총 45개의 키가 필요하겠네요.

수정 후 : 공개키 암호화 기법에서 키의 개수는 $2n$ 이므로, 10명이 암호를 사용할 때 키의 개수는 2×10 으로 총 20개의 키가 필요하겠네요.

[기술문제집]

42쪽 암기 247 아래쪽 작업 2, 작업 3 설명 부분

수정 전 :

- 작업 2 : 1에 도착하여 작업 1이 완료될 때까지 대기한 후 작업 1이 완료되는 6에서 실행을 시작하여 9에 작업이 완료된다. 그러므로 대기 시간은 5이고, 반환 시간은 9이다.
- 작업 3 : 2에 도착하여 작업 2가 완료될 때까지 대기한 후 작업 2가 완료되는 9에서 실행을 시작하여 13에 작업이 완료된다. 그러므로 대기 시간은 7이고, 반환 시간은 13이다.

수정 후 :

- 작업 2 : 1에 도착하여 작업 1이 완료될 때까지 대기한 후 작업 1이 완료되는 6에서 실행을 시작하여 9에 작업이 완료된다. 그러므로 대기 시간은 5이고, 반환 시간은 8이다.
- 작업 3 : 2에 도착하여 작업 2가 완료될 때까지 대기한 후 작업 2가 완료되는 9에서 실행을 시작하여 13에 작업이 완료된다. 그러므로 대기 시간은 7이고, 반환 시간은 11이다.

59쪽 6번 문제 정답

수정 전 : ④

수정 후 : ①

76쪽 오른쪽 단 64번 문제 지문

수정 전 :

```
#include <stdio.h>
main() {
    int a, b;
    for (a = 0; a < 2; a++)
        for (b = 0; b < 2; b++)
            printf("%d", !a && !b);
}
```

수정 후 :

```
#include <stdio.h>
main() {
    int a, b;
    for (a = 0; a < 2; a++)
        for (b = 0; b < 2; b++)
            printf("%d", !a && b);
}
```

108쪽 오른쪽 단 가운데 6번 해설 부분

수정 전 :

수정 후 :

⑥ temp[i]에 i의 값을 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	temp[]			
	[0]	[1]	[2]	[3]
0	0	0	0	0
1	0			
2		1		
3			2	
4				3

⑦ temp를 반환한다. 인수나 반환값으로 배열을 이름에 지정하면 배열의 시작 주소가 전달된다. 즉 temp 배열의 시작주소가 반환된다.

⑧ ⑦번에서 전달받은 정수형 배열 temp의 시작 주소를 iarr이 받는다.

	[0]	[1]	[2]	[3]
iarr	1	1	2	3

⑥ temp[i]에 i의 값을 저장한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	temp[]			
	[0]	[1]	[2]	[3]
0	0	0	0	0
1	0			
2		1		
3			2	
4				3

⑦ temp를 반환한다. 인수나 반환값으로 배열을 이름에 지정하면 배열의 시작 주소가 전달된다. 즉 temp 배열의 시작주소가 반환된다.

⑧ ⑦번에서 전달받은 정수형 배열 temp의 시작 주소를 iarr이 받는다.

	[0]	[1]	[2]	[3]
iarr	0	1	2	3

136쪽 왼쪽 단 65번 문제 보기 ②

수정 전 : ② 2128개의 주소를 표현할 수 있다.

수정 후 : ② 2^{128} 개의 주소를 표현할 수 있다.

140쪽 오른쪽 단 76번 전문가의 조언 해설 부분

수정 전 :

① 20개의 요소를 갖는 문자형 배열 str1을 선언하고 "KOREA"로 초기화한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[20]
str1	'K'	'O'	'R'	'E'	'A'	\0	...	

② 20개의 요소를 갖는 문자형 배열 str2를 선언하고 "LOVE"로 초기화한다.

	[0]	[1]	[2]	[3]	[4]	...	[20]
str2	'L'	'O'	'V'	'E'	\0	...	

③ 문자형 포인터 변수 p1을 선언하고 NULL로 초기화한다.

④ 문자형 포인터 변수 p2를 선언하고 NULL로 초기화한다.

⑤ p1에 str1 배열의 시작 주소를 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[20]			
p1	●	→	str1	'K'	'O'	'R'	'E'	'A'	\0	...	

⑥ p2에 str2 배열의 시작 주소를 저장한다.

	[0]	[1]	[2]	[3]	[4]	...	[20]			
p2	●	→	str2	'L'	'O'	'V'	'E'	\0	...	

⑦ p2는 str2를 가리키므로 str2[2]의 값인 'V'를 str1[1]에 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[20]
str1	'K'	'V'	'R'	'E'	'A'	\0	...	

⑧ p1은 str1을 가리키므로 str1[4]의 값인 'A'를 str2[3]에 저장한다.

	[0]	[1]	[2]	[3]	[4]	...	[20]
str2	'L'	'O'	'V'	'A'	\0	...	

⑨ str1의 문자열 뒤에 str2의 문자열을 이어붙인다.

• strcat(문자배열A, 문자배열B) : A 배열에 저장된 문자열의 마지막에 이어서 B 배열에 저장된 문자열을 이어붙인다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	...	[20]
str1	'K'	'V'	'R'	'E'	'A'	'L'	'O'	'V'	'A'	\0	...	

수정 후 :

① 20개의 요소를 갖는 문자형 배열 str1을 선언하고 "KOREA"로 초기화한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[19]
str1	'K'	'O'	'R'	'E'	'A'	\0	...	

② 20개의 요소를 갖는 문자형 배열 str2를 선언하고 "LOVE"로 초기화한다.

	[0]	[1]	[2]	[3]	[4]	...	[19]
str2	'L'	'O'	'V'	'E'	\0	...	

③ 문자형 포인터 변수 p1을 선언하고 NULL로 초기화한다.

④ 문자형 포인터 변수 p2를 선언하고 NULL로 초기화한다.

⑤ p1에 str1 배열의 시작 주소를 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[19]			
p1	●	→	str1	'K'	'O'	'R'	'E'	'A'	\0	...	

⑥ p2에 str2 배열의 시작 주소를 저장한다.

	[0]	[1]	[2]	[3]	[4]	...	[19]			
p2	●	→	str2	'L'	'O'	'V'	'E'	\0	...	

⑦ p2는 str2를 가리키므로 str2[2]의 값인 'V'를 str1[1]에 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	...	[19]
str1	'K'	'V'	'R'	'E'	'A'	\0	...	

⑧ p1은 str1을 가리키므로 str1[4]의 값인 'A'를 str2[3]에 저장한다.

	[0]	[1]	[2]	[3]	[4]	...	[19]
str2	'L'	'O'	'V'	'A'	\0	...	

⑨ str1의 문자열 뒤에 str2의 문자열을 이어붙인다.

• strcat(문자배열A, 문자배열B) : A 배열에 저장된 문자열의 마지막에 이어서 B 배열에 저장된 문자열을 이어붙인다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	...	[19]
str1	'K'	'V'	'R'	'E'	'A'	'L'	'O'	'V'	'A'	\0	...	

166쪽 오른쪽 단 69번 문제 해설 중 아래쪽 표

수정 전 :

네트워크	호스트 수	IP 주소 범위
1	32	192.168.1.0 ~ 63
2	32	192.168.1.64 ~ 127
3	32	192.168.1.128 ~ 191
4	32	192.168.1.192 ~ 255

수정 후 :

네트워크	호스트 수	IP 주소 범위
1	64	192.168.1.0 ~ 63
2	64	192.168.1.64 ~ 127
3	64	192.168.1.128 ~ 191
4	64	192.168.1.192 ~ 255