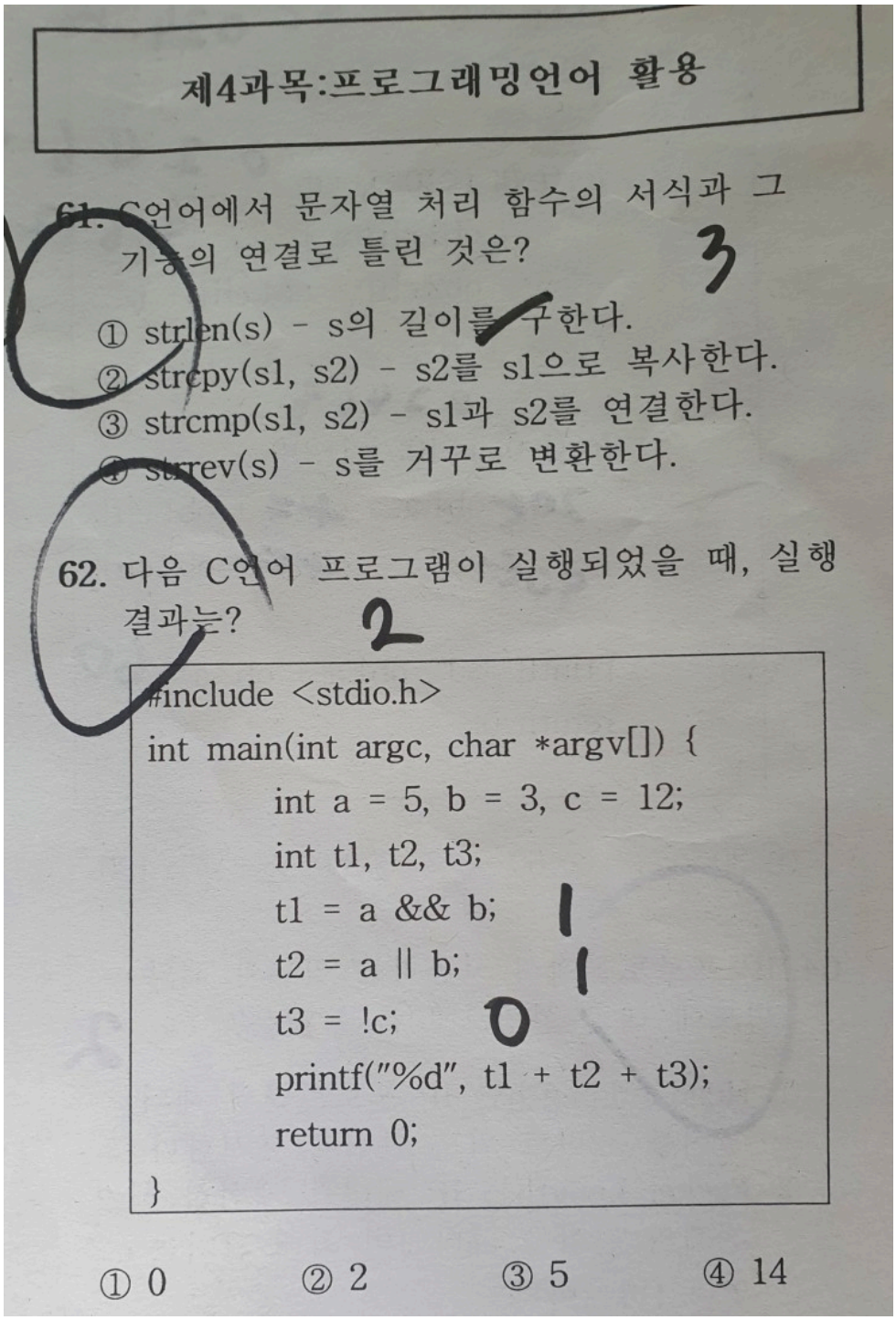
제4과목:프로그래밍언어 활용에 나왔던 프로그램 유형 문제 풀이를 진행해 보려고 한다.
사실 풀이라고까지 할 것도 없지만 해보고 싶었다.
정보처리기사에서 코드 관련 문제가 어떤 식으로 출제되는지 궁금한 사람에게 도움이 되지 않을까 싶다.

혹시 시험 끝나고 얼마 안 돼서 블로그에 문제 공유하면 문제가 있을까 해서 노파심에 문의 전화를 해봤다.
세 번 돌려진 끝에 사업 목적이 아니라 교육 목적이라면 상관 없다는 연락을 받았다!
살짝 애매한 답변이긴한데 시험 끝난지도 꽤 됐으니 괜찮겠지 생각하고 작성해본다.

C언어 4문제, JAVA 2문제, Python 1문제가 나왔다.
파이썬은 실행결과를 보고 소스 코드 빈 칸을 맞추는 문제, 나머지는 프로그램 실행 후 실행결과를 예측하는 문제이다.
언어별로 살펴보겠다.

C



제4과목:프로그래밍언어 활용

61. C언어에서 문자열 처리 함수의 서식과 그 기능의 연결로 틀린 것은?

① strlen(s) - s의 길이를 구한다.
② strcpy(s1, s2) - s2를 s1으로 복사한다.
③ strcmp(s1, s2) - s1과 s2를 연결한다.
④ strrev(s) - s를 거꾸로 변환한다.

62. 다음 C언어 프로그램이 실행되었을 때, 실행 결과는?

```
#include <stdio.h>
int main(int argc, char *argv[]) {
        int a = 5, b = 3, c = 12;
        int t1, t2, t3;
        t1 = a && b;
        t2 = a || b;
        t3 = !c;
        printf("%d", t1 + t2 + t3);
        return 0;
}
```

① 0 ② 2 ③ 5 ④ 14

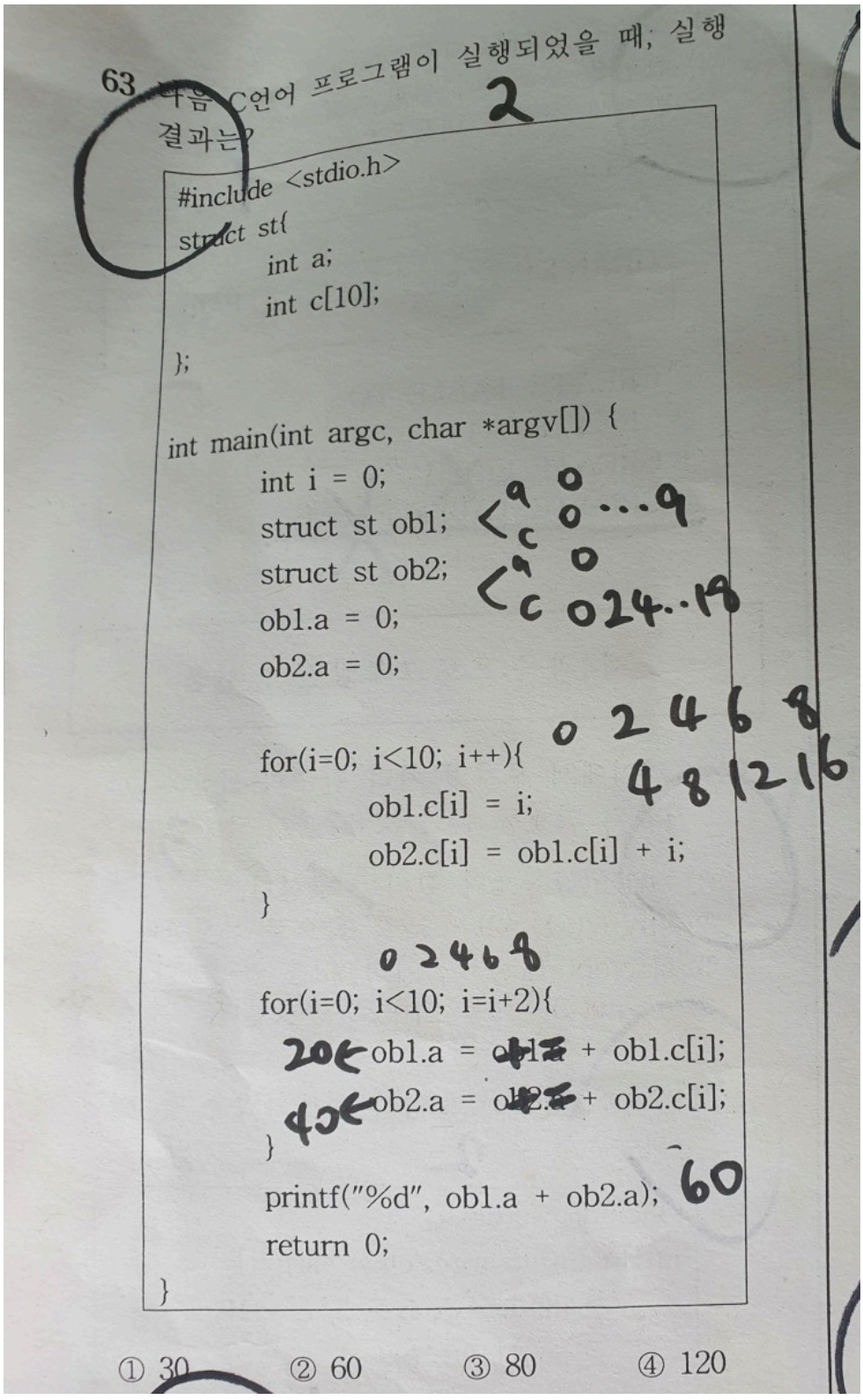
61번은 프로그램 유형의 문제는 아니지만 42서울하는 입장에서 굉장히 반가웠던 문제.
보기에 나온 함수들 기능 전부 다 구현해봤다.
정답은 3번. strcmp(s1, s2)는 s1과 s2를 연결하는 것이 아니라 비교한다.

62번이 프로그램 유형 문제다.
문제를 풀기 전 알아야 할 C언어의 논리연산자에 대해 아래 표로 정리해보았다.
C언어의 논리연산에서 0은 false, 0을 제외한 나머지 숫자는 true로 취급한다.
논리연산의 결과는 항상 0 또는 1이다.

구분	연산식	결과	설명
AND (논리곱)	false && false	0(false)	양쪽 다 true일 때만 1(true), 나머지는 0(false)
	false && true	0(false)	
	true && false	0(false)	

	true && true	1(true)	
OR (논리합)	false || false	0(false)	양쪽 다 false일 때만 0(false), 나머지는 1(true)
	false || true	1(true)	
	true || false	1(true)	
	true || true	1(true)	
NOT (논리부정)	!true	0(false)	피연산자의 논리값을 반대로 바꿈
	!false	1(true)	

a, b, c는 각각 5, 3, 12로 모두 0이 아니므로 true이다.
t1에는 a와 b의 AND(&&) 연산 값, t2에는 a와 b의 OR(||) 연산 값, t3에는 c의 NOT(!) 연산 값이 들어있다.
a와 b 모두 true이므로 t1과 t2 모두 1이다. c는 true이므로 !c인 t3는 그 반대인 0이다.
실행 결과는 t1, t2, t3의 합과 같으므로 1 + 1 + 0 = 2
정답은 2번이다.



63번은 구조체를 알아야 하는 문제.

구조체란 쉽게 말해 하나 이상의 변수를 묶어서 만든 새로운 자료형이다.

여기서 ob1.a는 구조체 ob1 안의 변수 a를 뜻하고, ob2.c[i]는 구조체 ob2 안의 배열 c의 i 인덱스에 있는 요소를 뜻한다.

첫 번째 for 문에서 ob1.c[i]에는 i를 대입하고, ob2.c[i]에는 ob1.c[i] + i를 대입하고 있는데 ob1.c[i]는 i이므로 사실상 2 * i인 셈이다.

따라서 첫 번째 for 문이 끝난 후 ob1.c, ob2.c에는 아래 표와 같이 값이 담겨있을 것이다.

i	0	1	2	3	4	5	6	7	8	9
ob1.c[i]	0	1	2	3	4	5	6	7	8	9

ob2.c[i]	0	2	4	6	8	10	12	14	16	18

두 번째 for 문에서 주의할 점은 i가 2씩 증가하고 있다는 점이다.

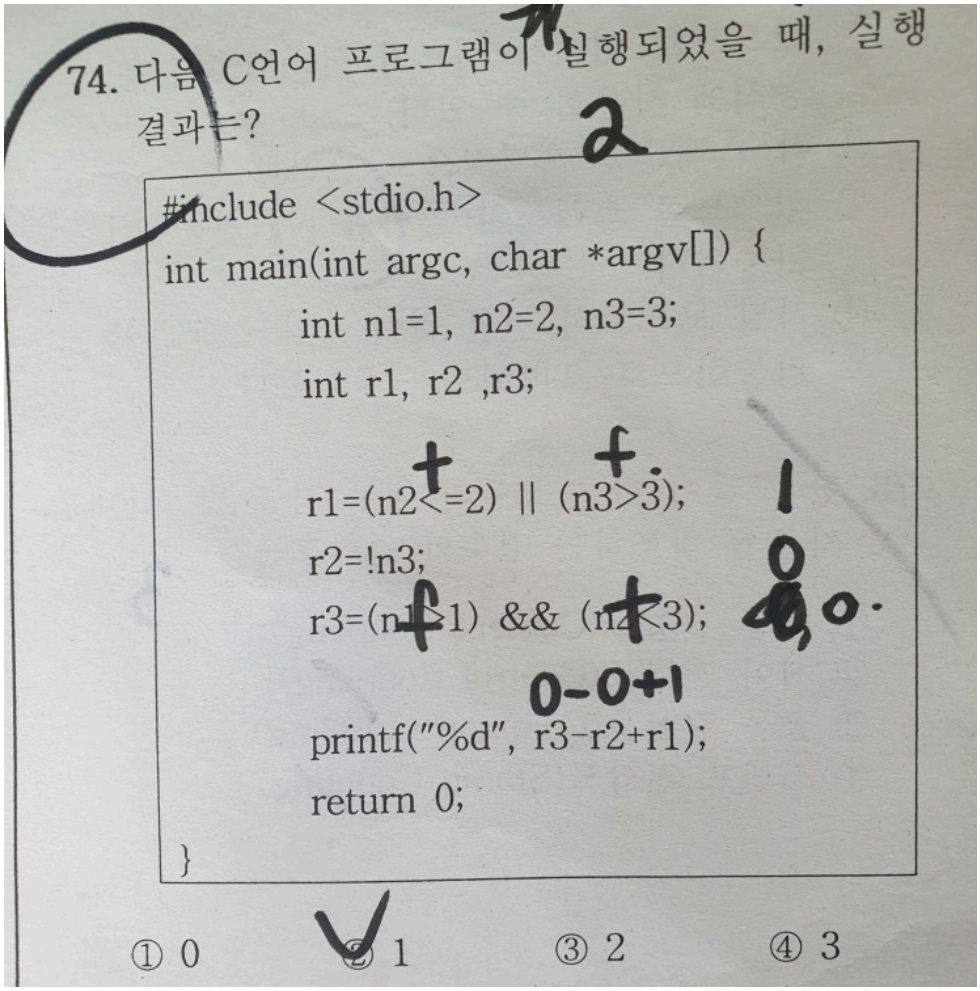
ob1.a, ob2.a가 0으로 초기화 되어 있는 상태에서 각각 ob1.c, ob2.c의 0, 2, 4, 6, 8 인덱스 요소를 더해주고 있으므로

결국 ob1.a와 ob2.a는 각각 ob1.c, ob2.c의 0, 2, 4, 6, 8 인덱스 요소의 합이라는 것을 알 수 있다.

따라서 ob1.a는 0 + 2+ 4 + 6 + 8 = 20, ob2.b는 0 + 4 + 8 + 12 + 16 = 40이다.

실행 결과는 ob1.a와 ob2.a의 합과 같으므로 20 + 40 = 60

정답은 2번이다.



74번은 62번 문제와 비슷한데, 관계연산자가 추가되어 조금 더 어려운 문제다.

관계연산자와 의미, 용례는 아래 표를 참고하면 된다.

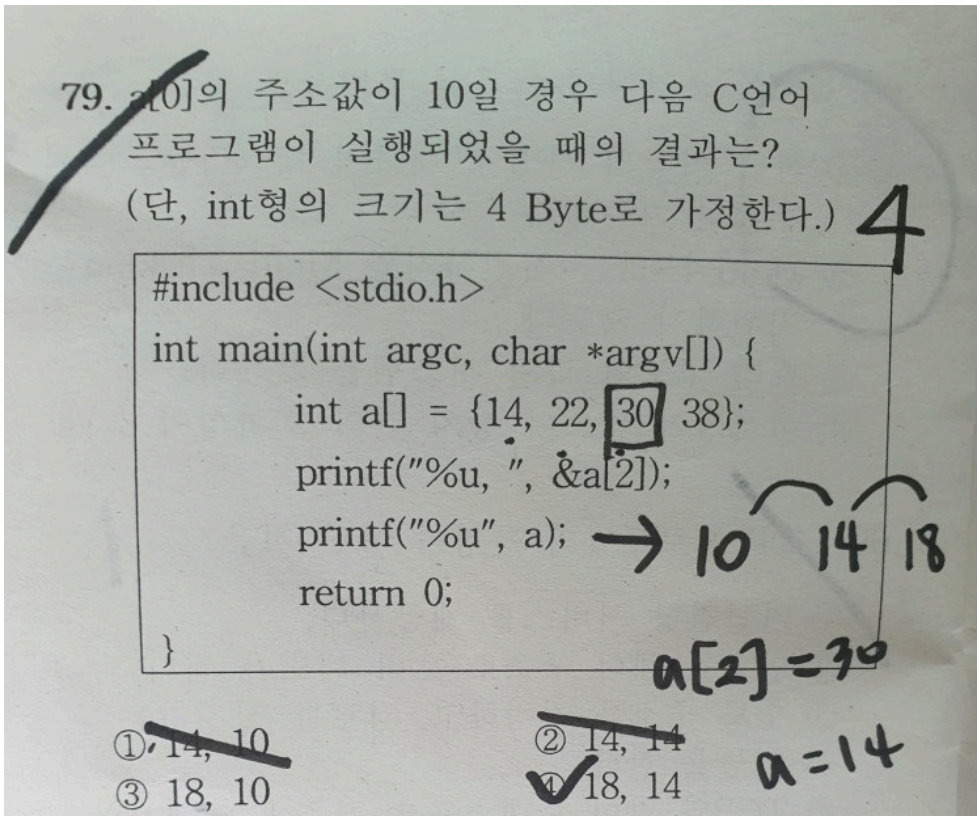
논리연산과 마찬가지로 연산 결과는 의미가 참이냐 거짓이냐에 따라 0(거짓) 또는 1(참)이다.

참은 true, 거짓은 false라 생각해도 무방하다.

연산자 기호	의미	사용예
==	x와 y가 같은가?	x == y
!=	x와 y가 다른가?	x != y
>	x가 y보다 큰가?	x > y
<	x가 y보다 작은가?	x < y
>=	x가 y보다 크거나 같은가?	x >= y
<=	x가 y보다 작거나 같은가?	x <= y

출처

r1에서 n2는 2이므로 n2<=2는 1(참)이고, n3은 3이므로 n3>3은 0(거짓)이다. 1과 0을 OR 연산하면 1이므로 r1은 1이다.
사실 OR 연산이라는 것을 먼저 확인했다면 n1<=2가 1이라는 것을 알았을 때 바로 r1이 1이라는 것을 알 수 있다.
OR 연산에서는 한 쪽이 true이면 다른 쪽과 관계 없이 결과가 1이기 때문이다.
r2에서 n3는 3으로 true이므로 그 반대인 !n3는 0이다. 따라서 r2는 0이다.
r3는 빠르게 풀어보자.
r3에서 n1은 1이므로 n1 > 1는 0(거짓)이다. AND 연산은 한 쪽이 false이면 다른 쪽과 관계 없이 결과가 0이므로 r3는 0이다.
실행 결과는 r3-r2+r1과 같으므로 0-0+1=1
정답은 2번이다.



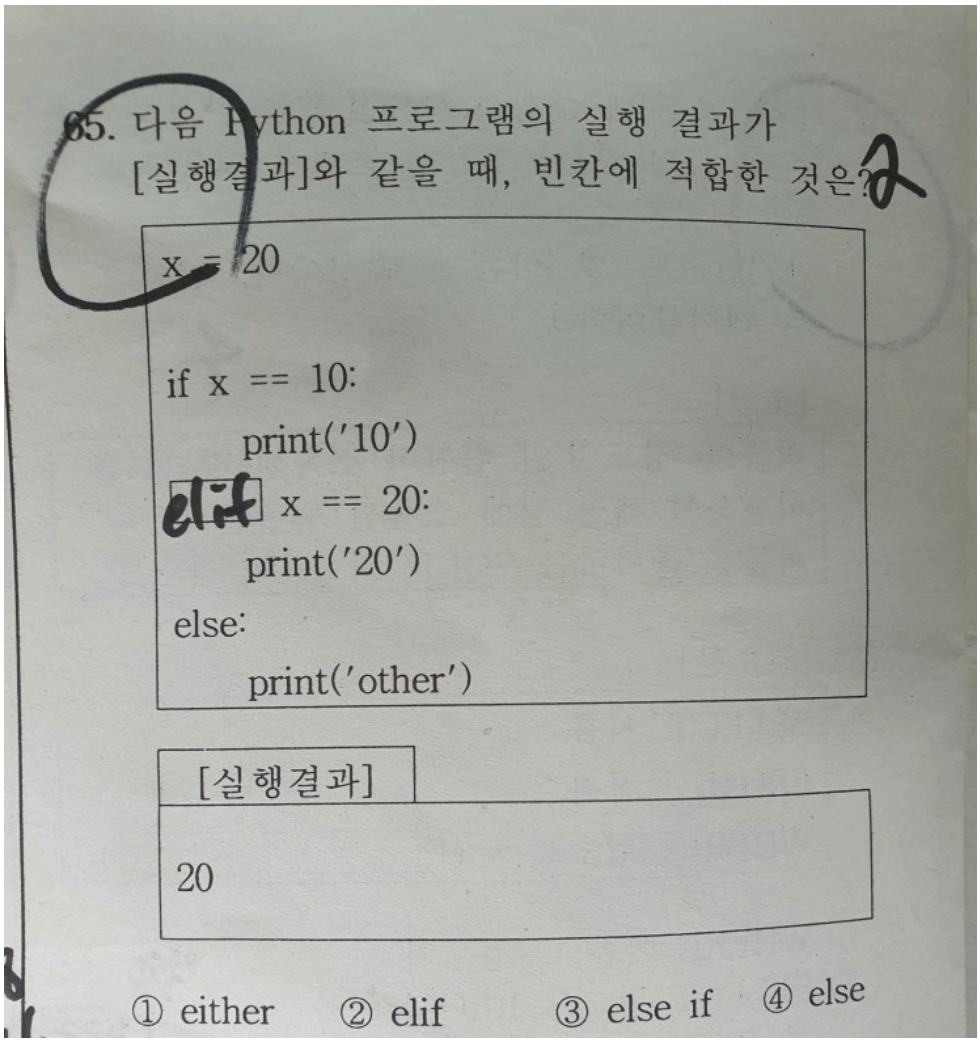
79번은 주소의 개념에 대해 알아야 하는 문제다.
C언어에서 변수 앞에 &를 붙이면 변수값이 아닌 그 변수의 주소값을 가리킨다.
배열의 주소값은 배열의 0 인덱스 주소값과 같으며, 인덱스가 1씩 증가할 때마다 주소값은 배열의 자료형 크기(Byte)만큼 증가한다.
공식으로 정리해보면 아래와 같다.

(a 배열의 자료형 크기가 b Byte일 때 a[c]의 주소값) = (a[0]의 주소값) + b * c

문제에서 a[0]의 주소값은 10이고 a는 자료형이 int형인 배열, int형의 크기는 4 Byte이다.
공식에 따라 a[2]의 주소값은 10 + 4 * 2 = 18이다.
따라서 실행 결과 앞에 "18, "이 출력되는 3번 또는 4번이 정답이다.

a를 출력했을 때 10이 나오느냐, 14가 나오느냐가 관건인데,
이는 a를 출력했을 때 a의 0 인덱스 값이 출력되느냐, 0 인덱스 주소값이 출력되느냐를 묻는 것과 같다.
부끄럽게도 나는 a의 0 인덱스 값이 출력될 것이라 생각하고 4번을 선택해 틀리고 말았다.
정답은 3번이다.
배열의 이름만 출력하면 그 배열의 주소값, 즉 배열의 0 인덱스 주소값이 출력된다.

Python


85. 다음 Python 프로그램의 실행 결과가 [실행결과]와 같을 때, 빈칸에 적합한 것은?

```
x = 20

if x == 10:
    print('10')
elif x == 20:
    print('20')
else:
    print('other')
```

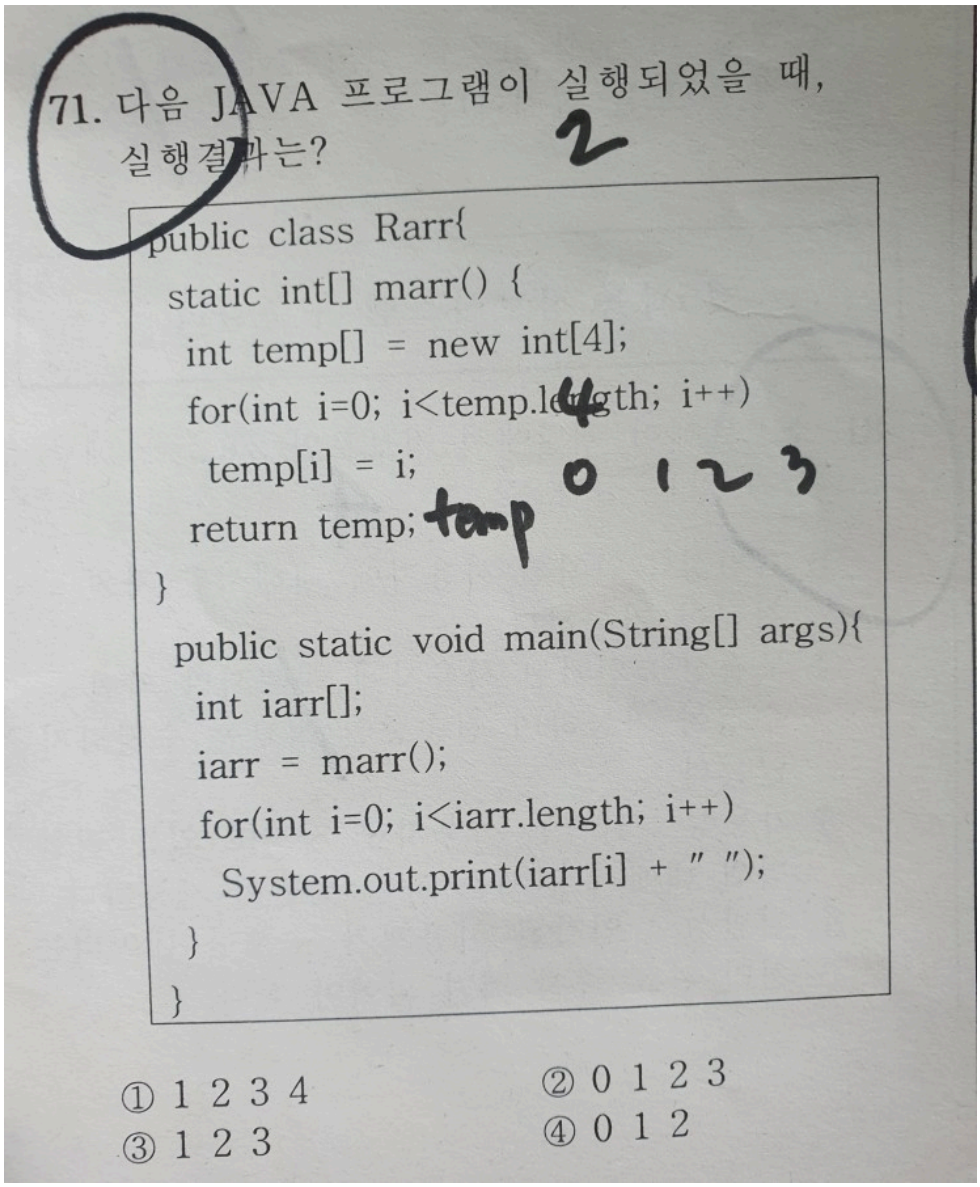
[실행결과]

```
20
```

① either ② elif ③ else if ④ else

85번 문제가 프로그램 유형 문제 중 유일한 빈 칸 채우기 문제다.
정답은 2번이다. 파이썬의 예약어인 elif를 알아야 풀 수 있다.
elif는 else인 상태에서 조건식을 지정할 때 사용하며 else if라는 뜻이다.
정답 이외의 것들을 넣으면 SyntaxError가 발생한다.

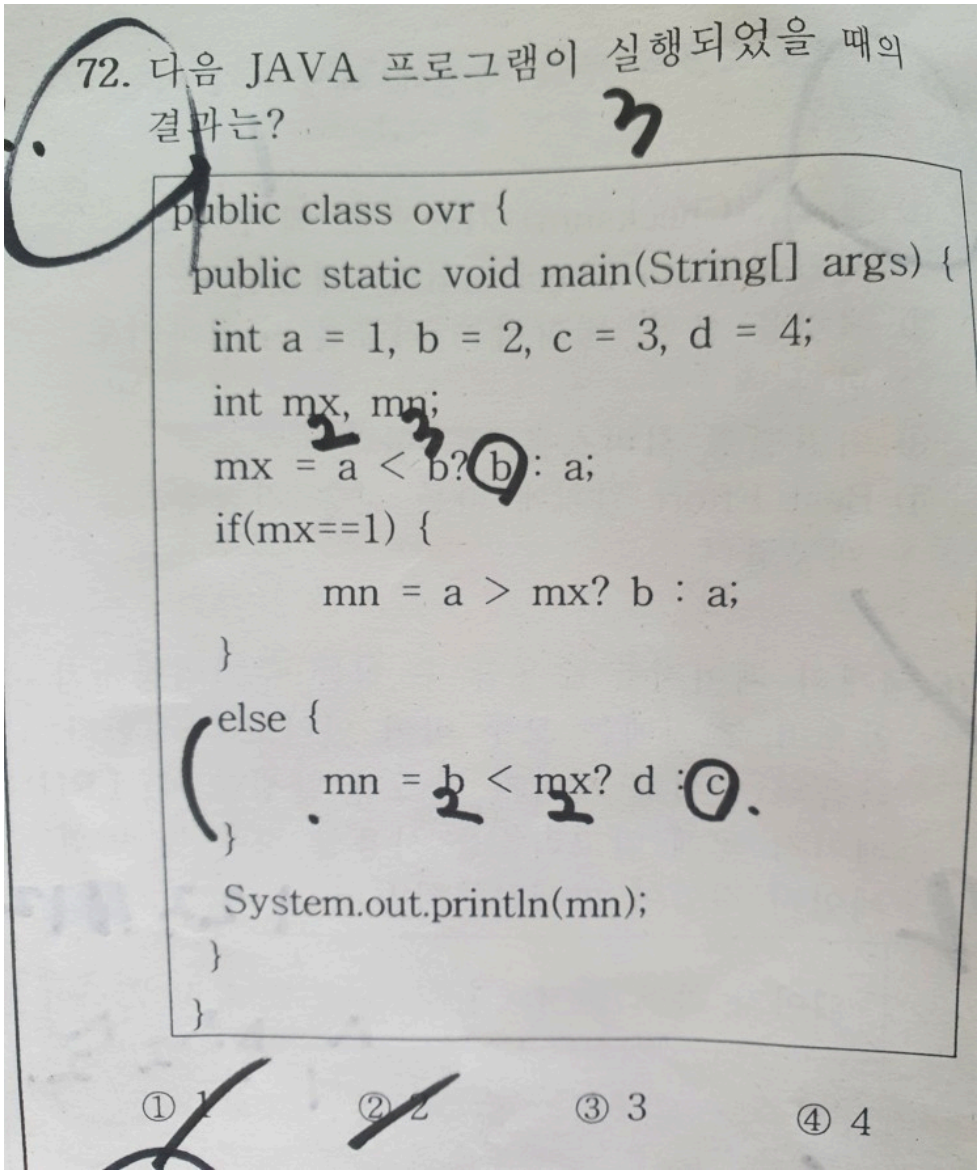
JAVA



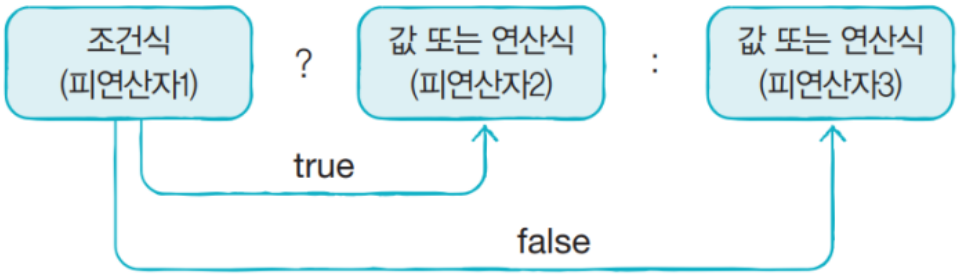
71번 문제는 메소드(클래스 함수)와 관련된 문제이다.
marr()의 반환값을 iarr에 대입하고 iarr의 길이만큼 0 인덱스 요소부터 순서대로 출력할 것이므로, marr()에 주목해야 한다.
marr()은 길이가 4인 int 배열을 0 인덱스부터 3 인덱스까지 인덱스 값으로 채워 반환하는 함수이다.
여기서 length는 배열의 속성으로, 배열의 길이를 나타낸다.
iarr에는 아래 표와 같이 값이 담겨있을 것이다.

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

인덱스 순서대로 요소 값을 출력하므로 정답은 2번이다.



72번 문제는 삼항연산자와 관련된 문제이다.
관계연산자도 보이는데, 앞에서 설명했으니 생략!
삼항연산자는 아래 그림과 같은 형태로 사용된다.



출처

mx = a < b ? b : a부터 살펴보자.
a는 1, b는 2이므로 a < b는 1(true)이다.
그림에 따라서 mx는 b, 즉 2로 초기화된다.

mx는 1이 아니므로 if 문은 넘어가고 else 문이 실행된다.
mn = b < mx ? d : c에서 b는 2, mx는 2이므로 b < mx는 0(false)이다.
그림에 따라서 mn은 c, 즉 3으로 초기화된다.

실행 결과는 mn과 같으므로 정답은 3번이다.

정답이 2번과 3번 뿐이고, 특히나 2번 정답이 많은 게 신기하다.
정리가 생각보다 오래 걸렸는데, 정리한 나에게나 이 글을 볼 다른 사람 모두에게 이 글이 도움이 되었으면 좋겠다.
문제에 나온 모든 프로그램 소스 코드는 문제 번호를 파일명으로 저장하고 압축해 첨부 파일로 업로드하였다.
자바는 클래스명을 파일명으로 저장하고, 문제 번호는 주석으로 적어놓았다.

정보처리기사2회필기220424.zip