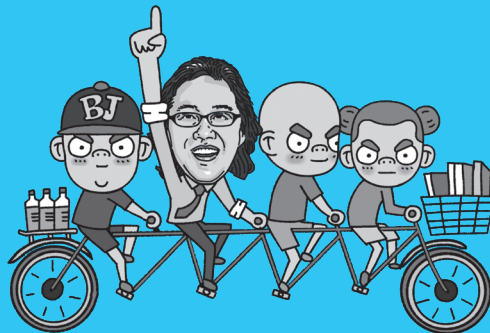


3부

정답 및 해설

- 알고리즘 해법 정답 및 해설
- 최신기출문제 정답 및 해설





알고리즘

① 2 ② HAP ③ C ④ A ⑤ B = C

A	B	HAP	N	K	C	출력
1	1	2	10	3	2	143
1	2	4		4	3	
2	3	7		5	5	
3	5	12		6	8	
5	8	20		7	13	
8	13	33		8	21	
13	21	54		9	34	
21	34	88		10	55	
34	55	143				



알고리즘

① 0 ② K ③ N/K ④ 1 ⑤ C

N	C	K	MOD(N,K)	배열 S	Y	출력
132	0	2	0	<div>2 2 3 11</div>	1	$2 \times 2 \times 3 \times 11$
66	1	2	0		2	
33	2	2	1		3	
11	3	3	0			
1	4	2	1			
		3	2			
		4	3			
		5	1			
		6	5			
		7	4			
		8	3			
		9	2			
		10	1			
		11	0			



알고리즘

① 0 ② J+1 ③ A[J] ④ A[J+1] ⑤ FLAG

N	i	FLAG	J	A[J]	A[J+1]	TM	배열 A
5	0 1	0 1 1 1	0 1 2 3 4	4 5 1 5 2 5 3 3	5 1 5 2 5 3 5	5 5 5	<div> <div>4 5 1 2 3</div> <div>1 5 5 5</div> <div>2 3</div> </div>
	2	0 1 1 1	0 1 2 3	4 1 4 2 4 3 3	1 4 2 4 3 4	4 4 4	<div> <div>4 1 2 3 5</div> <div>1 4 4 4</div> <div>2 3</div> </div>
	3	0	0 1 2	1 2	2 3		<div> <div>1 2 3 4 5</div> </div>



알고리즘

① YES ② $i + 1$ ③ $J + 1$ ④ 5 ⑤ $J > 5$

i	J	K	NMG	배열 A																									
1	3	1	1	<table><tr><td>17</td><td>24</td><td>1</td><td>8</td><td>15</td></tr><tr><td>23</td><td>5</td><td>7</td><td>14</td><td>16</td></tr><tr><td>4</td><td>6</td><td>13</td><td>20</td><td>22</td></tr><tr><td>10</td><td>12</td><td>19</td><td>21</td><td>3</td></tr><tr><td>11</td><td>18</td><td>25</td><td>2</td><td>9</td></tr></table>	17	24	1	8	15	23	5	7	14	16	4	6	13	20	22	10	12	19	21	3	11	18	25	2	9
17	24	1	8		15																								
23	5	7	14		16																								
4	6	13	20		22																								
10	12	19	21		3																								
11	18	25	2		9																								
0	4	2	2																										
5	5	3	3																										
4	6	4	4																										
3	1	5	0																										
2	2	6	1																										
3	3	7	2																										
2	4	8	3																										
1	5	9	4																										
0	6	10	0																										
5	1	11	1																										
4	2	12	2																										
5	3	13	3																										
4	4	14	4																										
3	5	15	0																										
2	6	16	1																										
1	1	17	2																										
2	2	18	3																										
1	3	19	4																										
0	4	20	0																										
5	5	21	1																										
4	6	22	2																										
3	1	23	3																										
4	2	24	4																										
3	3	25	0																										
2																													
1																													
0																													
5																													
6																													



SECTION
011

알고리즘 유형5 - 실무 응용

알고리즘

① 50000 ② T / M ③ $T - (PM[K] * M)$ ④ $M / 2$ ⑤ $PM[K] + TM[K]$

※ 아래 디버깅 표는 배열(TM, PM)을 초기화시키는 과정을 생략한 것입니다.

N	P	T	M	SW	K	배열	출력
엄철식	539620	539620	50000	1	1		출장비 지급표 성명 출장비 오만원 만원 오천원 천원 오백원 백원 오십원 십원 오원 일원 엄철식 539620 10 3 1 4 1 1 0 2 0 0
		39620	10000	0	2		
		9620	5000	1	3		
		4620	1000	0	4	PM 10 3 1 4 1 1 0 2 0 0	
		620	500	1	5		
		120	100	0	6		
		20	50	1	7	TM 10 3 1 4 1 1 0 2 0 0	
		20	10	0	8		
		0	5	1	9		
		0	1	0	10		
양동수	538973	538973	50000	1	1		출장비 지급표 성명 출장비 오만원 만원 오천원 천원 오백원 백원 오십원 십원 오원 일원 엄철식 539620 10 3 1 4 1 1 0 2 0 0 양동수 538973 10 3 1 3 1 4 1 2 0 3
		38973	10000	0	2		
		8973	5000	1	3		
		3973	1000	0	4	PM 10 3 1 3 1 4 1 2 0 3	
		973	500	1	5		
		473	100	0	6		
		73	50	1	7	TM 20 6 2 7 2 5 1 4 0 3	
		23	10	0	8		
		3	5	1	9		
		3	1	0	10		
이동훈	173105	173105	50000	1	1		출장비 지급표 성명 출장비 오만원 만원 오천원 천원 오백원 백원 오십원 십원 오원 일원 엄철식 539620 10 3 1 4 1 1 0 2 0 0 양동수 538973 10 3 1 3 1 4 1 2 0 3 이동훈 173105 3 2 0 3 0 1 0 0 1 0
		23105	10000	0	2		
		3105	5000	1	3		
		3105	1000	0	4	PM 3 2 0 3 0 1 0 0 1 0	
		105	500	1	5		
		105	100	0	6		
		5	50	1	7	TM 23 8 2 10 2 6 1 4 1 3	
		5	10	0	8		
		5	5	1	9		
		0	1	0	10		
QUIT							출장비 지급표 성명 출장비 오만원 만원 오천원 천원 오백원 백원 오십원 십원 오원 일원 엄철식 539620 10 3 1 4 1 1 0 2 0 0 양동수 538973 10 3 1 3 1 4 1 2 0 3 이동훈 173105 3 2 0 3 0 1 0 0 1 0 전체 회계 매수 23 8 2 10 2 6 1 4 1 3



[문제 1]

- ① $N + 1$ ② $77 - GK(N)$ ③ $GK(N) - 77$ ④ COMP ⑤ $GK(N)$

디버깅

배열 GK에 다음과 같이 값이 저장되었다고 가정하고 디버깅한 결과입니다.

GK	12	45	78	94	65	32	15	8	7	51
----	----	----	----	----	----	----	----	---	---	----

N	GK(N)	$GK(N) > 77$	CHA	COMP > CHA	COMP	FGK	출력
0	12	No	65	Yes	300	12	78
1	45	No	32	Yes	65	45	
2	78	Yes	1	Yes	32	78	
3	94	Yes	17	No	1		
4	65	No	12	No			
5	32	No	45	No			
6	15	No	62	No			
7	8	No	69	No			
8	7	No	70	No			
9	51	No	26	No			
10							

[문제 2]

- ① add ② result ③ pf ④ pf(i, j)

※ 답안 작성 시 주의 사항 : C언어 또는 Java에서 사용하는 변수명은 대소문자를 구분하기 때문에 변수명을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

[해설]

모든 C 프로그램은 반드시 main() 함수부터 시작한다.

```
main() {
  ① int i, j, result;
  ② int (*pf)(int, int);
  ③ scanf("%d, %d", &i, &j);
  ④ pf = add;
  ⑤ result = pf(i, j);
  printf("%d, ", result);
}
```

- ① 정수형 변수 i, j, result를 선언한다.
 ② 정수형 매개변수 두 개가 있는 정수형 함수 포인터 pf를 선언한다.

int (*pf)(int, int);

- ① ② ③

• ① : 반환값 자료형

• ② : 함수 포인터 이름, 함수 포인터 이름 앞에는 *를 붙인 뒤 ()로 묶어준다.

• ③ : 매개변수 자료형

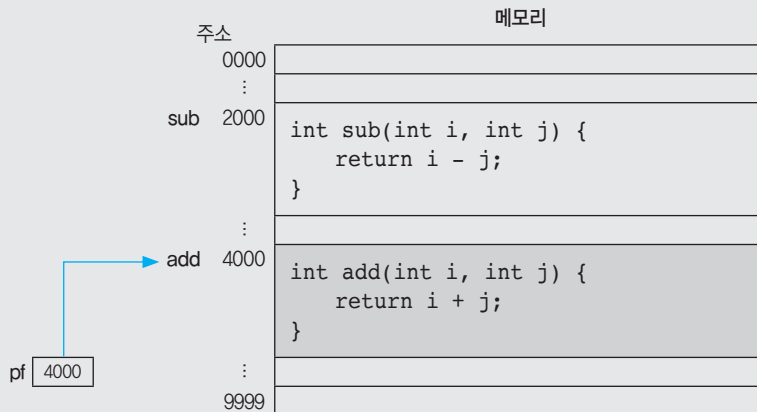
※ 함수 포인터 : 함수의 시작 주소를 저장할 수 있는 포인터로, 함수 포인터를 주고받거나 함수를 호출할 수 있음

③ 키보드로 숫자를 입력받아 값을 i와 j에 저장한다. 문제에서 12와 6을 입력받았다고 가정했으므로 i에는 12가 j에는 6이 저장된다.

④ add 함수의 시작 주소를 함수 포인터 pf에 저장한다. add 함수의 주소가 함수 포인터 pf에 저장된다는 것은 pf가 add 함수의 시작 주소를 가리키고 있다는 것을 의미한다.

⑤ i와 j, 즉 12와 6을 인수로 하여 pf 함수를 호출한 다음 돌려받은 값을 result에 저장한다.

pf에는 add 함수의 시작 주소가 저장되었으므로 pf 함수를 호출한다는 것은 add 함수를 호출하는 것과 같은 의미이다. 여기서 함수들이 할당된 공간의 주소는 임의로 정한 것이며, 이해를 돕기 위해 10진수로 표현했다.



```
⑥ int add(int i, int j) {
⑦     return i + j;
}
```

⑥ ⑤번에서 'pf(i, j)'라고 했으므로 정수형 변수 i는 i의 값 12를 받고 정수형 변수 j는 j의 값 6을 받는다.

⑦ i + j, 즉 12 + 6의 결과인 18을 가지고 add 함수를 호출했던 main() 함수로 제어를 옮긴다.

```
main() {
    int i, j, result;
    int (*pf)(int, int);
    scanf("%d, %d", &i, &j);
    pf = add;
    result = pf(i, j);
⑧ printf("%d, ", result);
⑨ pf = sub;
⑩ result = pf(i, j);
    printf("%d", result);
}
```

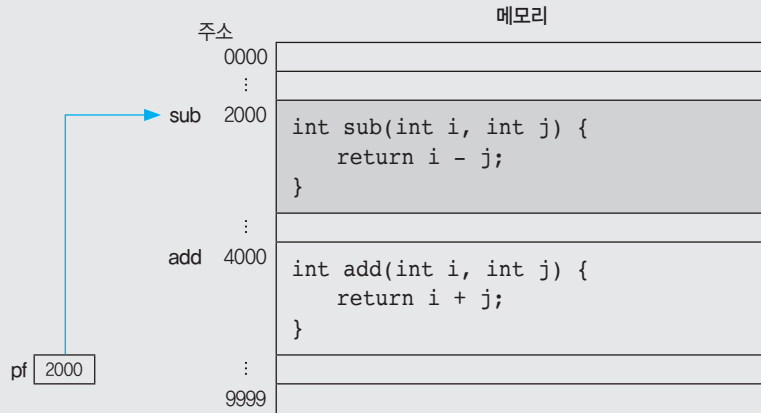
⑧ result에 저장된 값인 18을 정수형으로 출력하고 이어서 심표와 공백 한 칸을 출력한다.

결과 18,

⑨ sub 함수의 메모리 주소를 함수 포인터 pf에 저장한다.

⑩ i와 j, 즉 12와 6을 인수로 하여 pf 함수를 호출한 다음 돌려받은 값을 result에 저장한다.

pf에는 sub 함수의 메모리 주소가 저장되었으므로 pf 함수를 호출한다는 것은 sub 함수를 호출한다는 의미이다.



```
11 int sub(int i, int j){
12     return i - j;
}
```

- ⑪ ⑩번에서 'pf(i, j)'라고 했으므로 정수형 변수 i는 i의 값 12를 받고 정수형 변수 j는 j의 값 6을 받는다.
- ⑫ i - j, 즉 12 - 6의 결과인 6을 가지고 sub 함수를 호출했던 main() 함수로 제어를 옮긴다.

```
main() {
    int i, j, result;
    int (*pf)(int, int);
    scanf("%d, %d", &i, &j);
    pf = add;
    result = pf(i, j);
    printf("%d, ", result);
    pf = sub;
    result = pf(i, j);
13 printf("%d", result);
}
```

- ⑬ result에 저장된 값인 6을 정수형으로 출력한 후 프로그램을 종료한다.

결과 18, 6

[문제 3]

false

true

4

0

1

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값들을 공백이나 심표로 구분 하여 false true 4 0 1 또는 false, true, 4, 0, 1 같이 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[해설]

모든 Java 프로그램은 반드시 main() 메소드부터 시작한다.

```
public class Test {  
    public static void main(String [ ]args) {  
        ❶ int x = 1;  
        ❷ System.out.println(!(x > 0));  
        ❸ System.out.println(x > 0 || x < 4);  
        ❹ System.out.println(x << 2);  
        ❺ System.out.println(x & 2);  
        ❻ System.out.println(x % 3);  
    }  
}
```

❶ 정수형 변수 x를 선언하고 1로 초기화한다.

❷ !(논리 not)는 결과의 부정, 즉 결과가 true이면 false를, 결과가 false이면, true를 반환하는 논리 연산자이다.

• x의 값 1은 0보다 크므로 true이지만, !(논리 not) 연산이 수행되어 false가 출력된다. 출력 후 커서를 다음 줄의 처음으로 옮긴다.

결과 false

❸ ||(논리 or)는 두 값 중 하나라도 true면 true를 반환하는 논리 연산자이다.

$x > 0 || x < 4$

① ②

③

① : x의 값 1은 0보다 크므로 true이다.

② : x의 값 1은 4보다 작으므로 true이다.

③ : ①(true) || ②(true)이므로 true가 출력된다. 출력 후 커서를 다음 줄의 처음으로 옮긴다.

결과
false
true

❹ <<는 왼쪽 시프트 연산자이므로 x에 저장된 값 1을 왼쪽으로 2비트 이동시킨 다음 그 값을 출력한다. int는 4Byte이므로 4Byte 2진수로 변환하여 계산하면 된다.

• 4바이트에 1을 2진수로 표현하면 다음과 같다.

	32	31	30	...	20	...	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
1	0	0	0	...	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
부호 비트																2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
																128	64	32	16	8	4	2	1

부호 비트

• 부호를 제외한 전체 비트를 왼쪽으로 2비트 이동시킨다. 양수이므로 빈 자리(패딩 비트)에는 0이 들어오면 된다.

	32	31	30	...	20	...	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
4	0	0	0	...	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
부호 비트																2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
																128	64	32	16	8	4	2	1

부호 비트

패딩 비트

• 이것을 10진수로 변환하면 4이다. 4를 출력하고 커서를 다음 줄의 처음으로 옮긴다.

결과
false
true
4

❺ &(비트 and)는 두 비트가 모두 1일 때만 1이 되는 비트 연산자이다.

• x의 값 1과 2를 & 연산하면 다음과 같다.

```
x(1)  = 0001  
2     = 0010  
-----  
&     = 0000 = 0
```

• 0을 출력하고 커서를 다음 줄의 처음으로 옮긴다.

결과 false
true
4
0

⑥ x의 값 1을 3으로 나눈 나머진 1을 출력하고 커서를 다음 줄의 처음으로 옮긴다.

결과 false
true
4
0
1

[문제 4]

n

[해설]

모든 Java 프로그램은 반드시 main() 메소드부터 시작한다.

```
public class Test {  
    public static void main(String[] args) {  
        ① int p = 2, n = 3;  
        ② while (true) {  
            ③ double t = Math.sqrt(n);  
            ④ int m = (int)t;  
            ⑤ for (int i = 2; i <= m; i++) {  
                ⑥ int r = n % i;  
                ⑦ if (r == 0) break;  
                ⑧ if (i == m) p = n;  
            }  
            ⑨ n++;  
            ⑩ if (n > 100) break;  
        }  
        ⑪ System.out.println(p);  
    }  
}
```

- ① 정수형 변수 p와 n을 선언하고, 각각 2와 3으로 초기화한다.
- ② break문을 만날 때까지 ③~⑩번 문장을 반복 수행한다.
- ③ n의 제곱근을 구한 후 실수형 변수 t에 저장한다.
※ Math.sqrt() : Java 기본 라이브러리(java.lang)의 Math 클래스에 포함된 제곱근을 구하는 수학 함수
- ④ 실수형 변수 t의 값을 정수형으로 변환한 후 변수 m에 저장한다.
- ⑤ 반복 변수 i가 2부터 1씩 증가하면서 m보다 작거나 같은 동안 ⑥~⑧번을 반복 수행한다.
- ⑥ 정수형 변수 r을 선언하고, n을 i로 나눈 나머지 값으로 초기화한다.
- ⑦ r이 0이면 ⑤번 for 반복문을 탈출하여 ⑨번으로 이동한다.
- ⑧ i가 m의 값과 같으면 p에 n의 값을 저장한다.
- ⑨ 'n = n + 1'과 동일하다. n의 값을 1씩 누적시킨다.
- ⑩ n이 100보다 크면 ②번 while 반복문을 탈출하여 ⑪번으로 이동한다.
- ⑪ p의 값을 출력하고, 커서를 다음 줄의 처음으로 옮긴다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

p	n	t	m	i	r	출력
2	3	1.73205...	1	2	0	97
5	4	2.00000...	2	2	1	
7	5	2.23606...	2	2	0	
11	6	2.44949...	2	3	1	
13	7	2.64575...	2	2	0	
⋮	8	2.82842...	2	2	1	
97	9	3.00000...	3	3	0	
	10	3.16227...	3	2	0	
	11	3.31662...	3	2	1	
	12	3.46410...	3	3	2	
	13	3.60555...	3	2	0	
	⋮	⋮	⋮	2	1	
	97	9.84885...	9	3	1	
	98	9.89949...	9	4	⋮	
	99	9.94987...	9	2	1	
	100	10.00000...	10	2	1	
	101			3	1	
				⋮	2	
				2	1	
				3	6	
				4	1	
				5	7	
				6	0	
				7	1	
				8	0	
				9	0	
				2		
				2		
				3		
				2		

[문제 5]

- ① SELECT 학생이름, 생년월일 FROM student WHERE 전공 = 'computer' AND 학년 = 1;
- ② SELECT DISTINCT 전공 FROM student WHERE 학년 IN(2, 4);
- ③ SELECT 학생이름 FROM student WHERE 학년 >= 2 ORDER BY 학년 DESC;

[풀이]

①

• SQL문

SELECT 학생이름, 생년월일	'학생이름', '생년월일'을 표시한다.
FROM student	〈student〉 테이블을 대상으로 검색한다.
WHERE 전공 = 'computer'	'전공'이 "computer"이고,
AND 학년 = 1;	'학년'이 1인 튜플만을 대상으로 한다.

〈결과〉

학생이름	생년월일
oliver moore	020403

②

• SQL문

SELECT DISTINCT 전공	'전공'을 표시하되, 같은 '전공'은 한 번만 표시한다.
FROM student	〈student〉 테이블을 대상으로 검색한다.
WHERE 학년 IN(2, 4);	'학년'의 값이 2 또는 4인 튜플만을 대상으로 한다.

〈결과〉

전공
computer
law

③

• SQL문

SELECT 학생이름	'학생이름'을 표시한다.
FROM student	〈student〉 테이블을 대상으로 검색한다.
WHERE 학년 >= 2	'학년'이 2 이상인 튜플만을 대상으로 한다.
ORDER BY 학년 DESC;	'학년'을 기준으로 내림차순 정렬한다.

〈결과〉

학생이름
reggie brown
ethan miller
isaac davis
jaxon williams

[문제 6]

※ 다음 중 하나를 쓰면 됩니다.

E-R 모델, E-R 다이어그램, ERD, E-R Model, E-R Diagram

[문제 7]

① Address ② Hobby ③ Age ④ StudNo

[문제 8]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 에지, EDGE, Enhanced Data Rates For Global Evolution
- ② 비트코인, bitcoin
- ③ 고화질 멀티미디어 인터페이스, HDMI, High-Definition Multimedia Interface
- ④ 인터넷 전화, VoIP, Voice over Internet Protocol
- ⑤ 미라캐스트, Miracast

[문제 9]

※ 다음 중 하나를 쓰면 됩니다.

베타 테스트, Beta Test

[문제 10]

※ 다음 중 하나를 쓰면 됩니다.

스마트 그리드, Smart Grid

[문제 11]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① KMS, 지식 관리 시스템, Knowledge Management System
- ② CKO, 지식 경영 최고 책임자 Chief Knowledge Officer
- ③ EDMS, 전자 문서 관리 시스템, Electronic Document Management System
- ④ COP, 지식 공동체, Communities of Practice
- ⑤ SWOT 분석, SWOT Analysis

[문제 12]

- ① SQL ② DDL ③ SELECT ④ FROM ⑤ WHERE

※ 답안 작성 시 주의 사항 : 문제에서 영문 약어로 작성하라고 하였으므로, 영문 풀네임(Full-name), 한글 등으로 작성할 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[원문해석]

(SQL)이라는 이름은 구조화된 질의 언어로 불린다.

원래 (SQL)은 SEQUEL(구조화된 영문 질의 언어)이라고 불렸으며, IBM 연구소에서 실험적인 관계형 데이터베이스 시스템인 시스템 R을 위한 인터페이스로 설계 및 구현되었다.

(DDL)은 데이터베이스의 여러 구조를 정의하는 명령의 표준이다. (DDL) 문은 테이블, 색인 그리고 사용자와 같은 데이터베이스 객체들을 생성, 수정, 그리고 삭제한다. 일반적인 (DDL) 문은 CREATE, ALTER, 그리고 DROP이다.

(SQL)은 데이터베이스에서 정보를 검색하기 위한 기본적인 구문이 하나 있는데, (SELECT) 문이 그것이다. 이것은 관계대수의 (SELECT) 연산과는 다르다.

(SELECT) 문의 기본 형식은 (SELECT), (FROM), 그리고 (WHERE)의 세 가지 절로 구성되며, 다음과 같은 형식을 취한다.

- (SELECT) <속성 목록>
- (FROM) <테이블 목록>
- (WHERE) <조건>

<속성 목록>은 질의에 의해 검색될 값을 갖는 속성들의 목록이다.

<테이블 목록>은 질의를 처리하기 위해 요구되는 릴레이션 이름의 목록이다.

<조건>은 질의에 의해 검색될 튜플들을 식별할 조건식이다.



[문제 1]

① $S = J$ ② J ③ $F = J$ ④ $K + 1$ ⑤ $A(Q, J) = K$

디버깅

N	K	SW	S	F	J	Q	출력																																																	
7	0	4																																																						
	1 2 3 4 5 6 7		1	7	1	1 2 3 4 5 6 7 8	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2							3							4							5							6							7						
1																																																								
2																																																								
3																																																								
4																																																								
5																																																								
6																																																								
7																																																								
	8 9 10 11 12		2	6	2	2 3 4 5 6 7	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9						4	10						5	11						6	12						7						
1																																																								
2	8																																																							
3	9																																																							
4	10																																																							
5	11																																																							
6	12																																																							
7																																																								
	13 14 15		3	5	3	3 4 5 6	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13					4	10	14					5	11	15					6	12						7						
1																																																								
2	8																																																							
3	9	13																																																						
4	10	14																																																						
5	11	15																																																						
6	12																																																							
7																																																								
	16		4	4	4	4 5	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13					4	10	14	16				5	11	15					6	12						7						
1																																																								
2	8																																																							
3	9	13																																																						
4	10	14	16																																																					
5	11	15																																																						
6	12																																																							
7																																																								
	17 18 19		3	5	5	3 4 5 6	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13		17			4	10	14	16	18			5	11	15		19			6	12						7						
1																																																								
2	8																																																							
3	9	13		17																																																				
4	10	14	16	18																																																				
5	11	15		19																																																				
6	12																																																							
7																																																								

	20 21 22 23 24		2	6	6	2 3 4 5 6 7	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td>20</td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td>21</td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td>22</td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td>23</td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td>24</td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8				20		3	9	13		17	21		4	10	14	16	18	22		5	11	15		19	23		6	12				24		7						
1																																																								
2	8				20																																																			
3	9	13		17	21																																																			
4	10	14	16	18	22																																																			
5	11	15		19	23																																																			
6	12				24																																																			
7																																																								
	25 26 27 28 29 30 31		1	7	7	1 2 3 4 5 6 7 8	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td>25</td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td>20</td><td>26</td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td>21</td><td>27</td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td>22</td><td>28</td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td>23</td><td>29</td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td>24</td><td>30</td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td>31</td></tr></table>	1						25	2	8				20	26	3	9	13		17	21	27	4	10	14	16	18	22	28	5	11	15		19	23	29	6	12				24	30	7						31
1						25																																																		
2	8				20	26																																																		
3	9	13		17	21	27																																																		
4	10	14	16	18	22	28																																																		
5	11	15		19	23	29																																																		
6	12				24	30																																																		
7						31																																																		
				8																																																				

[문제 2]

- ① 1 ② n-2

※ 답안 작성 시 주의 사항 : C언어에서 사용하는 변수명은 대소문자를 구분하기 때문에 변수명을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

[피보나치 수의 정의]

피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 됩니다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2574, ...

Fibonacci(n)

- if n=0, 0 : n이 0이면 0을 반환
- if n=1, 1 : n이 1이면 1을 반환
- others, Fibonacci(n-2) + Fibonacci(n-1) : 그 외에는 바로 앞의 두 수의 합을 반환

```
int main(void) {
    ❶ int i=0;
    ❷ for(i=0; i<10; i++)
    ❸     printf("%d ", Fibonacci(i));
    return 0;
}
```

모든 C 프로그램은 반드시 main() 함수부터 시작한다.

- ❶ 정수형 변수 i를 선언하고, i의 초기값으로 0을 할당한다.
- ❷ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 10보다 작은 동안 ❸번 문장을 반복합니다. 즉 ❸번 문장을 10회 반복한다.
- ❸ i의 값을 인수로 하여 Fibonacci() 함수를 호출한 다음 돌려받은 값을 정수형으로 출력합니다. 처음에는 i가 0이므로 Fibonacci(0) 함수를 호출한다.

```
int Fibonacci(int n) {
    ❹ if ( n == 0 )
    ❺     return 0;
    else if ( n == 1 )
        return 1;
    else
        return Fibonacci(n-2) + Fibonacci(n-1);
}
```

③번에서 0이 전달되었으므로 n은 0이다. ④번의 조건을 만족하므로 ⑤번을 수행한다. 반환값 0을 가지고 Fibonacci(0) 함수를 호출했던 main() 함수로 제어를 옮긴다.

```
int main(void) {
    int i=0;
    ⑦ for(i=0; i<10; i++)
    ⑥⑧    printf("%d ", Fibonacci(i));
    return 0;
}
```

⑥ Fibonacci() 함수에서 돌려받은 값 0을 출력한다.

결과 0

⑦ i의 값을 1증가시킨 후 최종값과 비교한다. i값 1은 10보다 작으므로 ⑧번 문장을 수행한다.

⑧ i의 값을 인수로 하여 Fibonacci() 함수를 호출한 다음 돌려받은 값을 정수형으로 출력한다. i가 1이므로 Fibonacci(1) 함수를 호출한다.

```
int Fibonacci(int n){
    if ( n == 0 )
        return 0;
    ⑨ else if ( n == 1 )
    ⑩     return 1;
    else
        return Fibonacci(n-2) + Fibonacci(n-1);
}
```

⑧번에서 1이 전달되었으므로 n은 1이다. ⑨번의 조건을 만족하므로 ⑩번을 수행한다. 반환값 1을 가지고 Fibonacci(1) 함수를 호출했던 main() 함수로 제어를 옮긴다.

```
int main(void) {
    int i=0;
    ⑫ for(i=0; i<10; i++)
    ⑪⑬    printf("%d ", Fibonacci(i));
    return 0;
}
```

⑪ Fibonacci() 함수에서 돌려받은 값 1을 출력한다.

결과 0 1

⑫ i의 값을 1증가시킨 후 최종값과 비교한다. i값 2는 10보다 작으므로 ⑬번 문장을 수행한다.

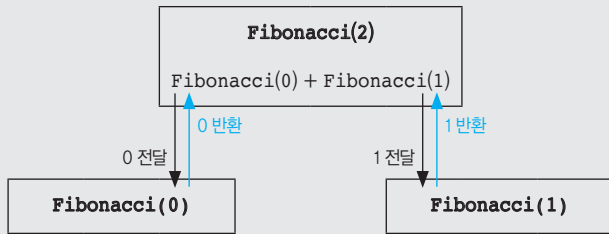
⑬ i 값을 인수로 하여 Fibonacci() 함수를 호출한 다음 돌려받은 값을 정수형으로 출력한다. i가 2이므로 Fibonacci(2) 함수를 호출한다.

i가 0일 때는 Fibonacci(0) 함수를 호출하여 리턴값이 0이고, i가 1일 때는 Fibonacci(1) 함수를 호출하여 리턴값이 1이라는 것을 이해했습니다. 나머지 과정은 자신이 자신을 호출하는 재귀 함수를 이용하는 데, 앞에서 이해한 Fibonacci(0), Fibonacci(1)의 결과를 사용하면 됩니다. 코드는 앞에서 이해한 내용이 반복되므로 이제부터는 두 수의 합으로 피보나치 수가 만들어지는 과정을 개괄적인 그림과 디버깅 표로 설명하겠습니다.

i=2일 때, Fibonacci(2) 함수를 호출하며 n은 2이다.

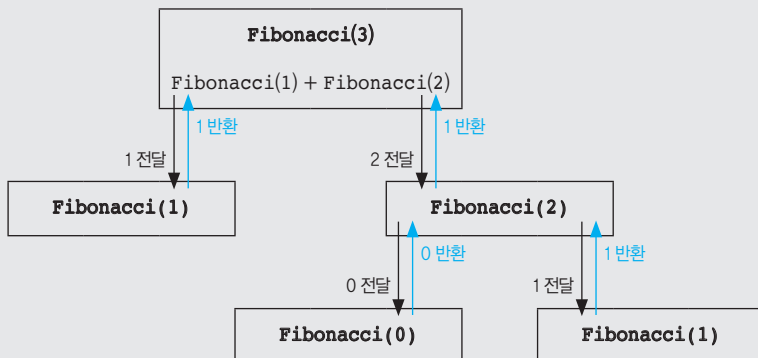
```
int Fibonacci(int n){
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    ⑭ else
    ⑮     return Fibonacci(n-2) + Fibonacci(n-1);
}
```

⑭의 조건에 해당되므로 ⑮번 문장, 즉 'Fibonacci(0) + Fibonacci(1)'을 수행한 후 결과를 반환한다.



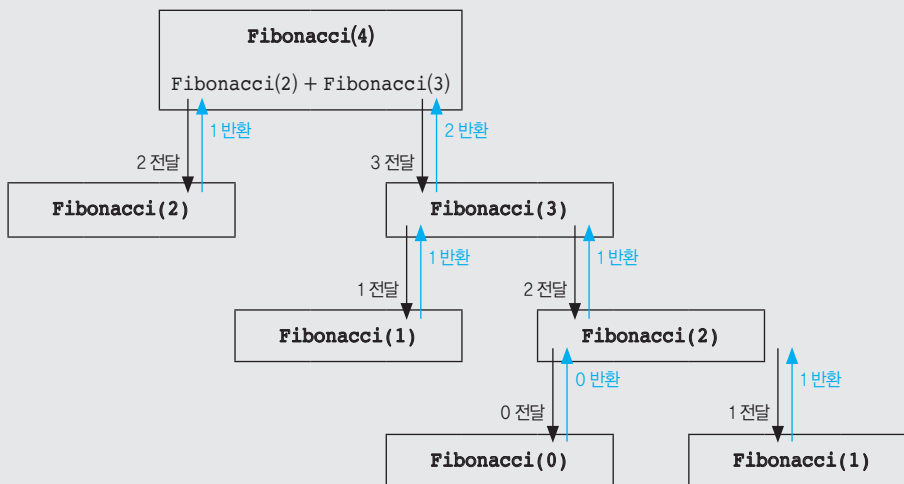
- $\text{Fibonacci}(n-2) + \text{Fibonacci}(n-1)$ 은 $0 + 1$ 이므로 1을 반환한다.
 - main 함수에서는 $\text{Fibonacci}(2)$ 함수를 호출하여 1을 돌려받으므로 최종적인 출력 결과는 다음과 같다.
- 결과 0 1 1

i=3일 때, $\text{Fibonacci}(3)$ 함수를 호출하며 n은 3이다.
 n이 3이므로 $\text{Fibonacci}(1) + \text{Fibonacci}(2)$ 를 수행한 후 결과를 반환한다.



- $\text{Fibonacci}(n-2) + \text{Fibonacci}(n-1)$ 은 $1 + 1$ 이므로 2를 반환한다.
 - main 함수에서는 $\text{Fibonacci}(3)$ 함수를 호출하여 2를 돌려받으므로 최종적인 출력 결과는 다음과 같다.
- 결과 0 1 1 2

i=4일 때, $\text{Fibonacci}(4)$ 함수를 호출하며 n은 4이다.
 n이 4이므로 $\text{Fibonacci}(2) + \text{Fibonacci}(3)$ 을 수행한 후 결과를 반환한다.



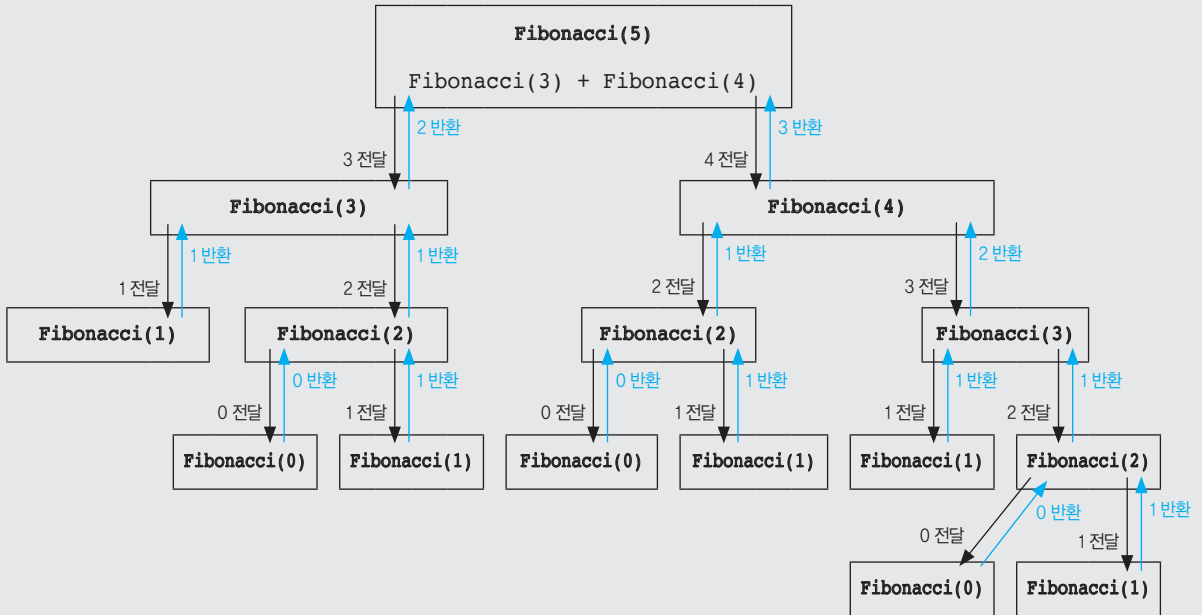
- $\text{Fibonacci}(n-2) + \text{Fibonacci}(n-1)$ 은 $1 + 2$ 이므로 3을 반환한다.

• main 함수에서는 Fibonacci(4) 함수를 호출하여 3을 돌려받으므로 최종적인 출력 결과는 다음과 같다.

결과 0 1 1 2 3

i=5일 때, Fibonacci(5) 함수를 호출하며 n은 5이다.

n이 5이므로 Fibonacci(3) + Fibonacci(4)를 수행한 후 결과를 반환한다.



• Fibonacci(n-2) + Fibonacci(n-1)은 2 + 3이므로 5를 반환한다.

• main 함수에서는 Fibonacci(5) 함수를 호출하여 5를 돌려받으므로 최종적인 출력 결과는 다음과 같다.

결과 0 1 1 2 3 5

이후 과정도 앞에서 살펴본 과정의 반복입니다. main() 함수의 i 값이 10이 될 때까지 모두 수행해 보세요.

위의 과정을 정리하면 디버깅 결과는 다음과 같습니다.

디버깅

i	함수 호출	n	Fibonacci() 함수 리턴값	출력
0	Fibonacci(0)	0	0	0
1	Fibonacci(1)	1	1	1
2	Fibonacci(2)	2	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(0) + Fibonacci(1) = 0 + 1	1
3	Fibonacci(3)	3	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(1) + Fibonacci(2) = 1 + 1	2
4	Fibonacci(4)	4	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(2) + Fibonacci(3) = 1 + 2	3
5	Fibonacci(5)	5	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(3) + Fibonacci(4) = 2 + 3	5
6	Fibonacci(6)	6	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(4) + Fibonacci(5) = 3 + 5	8
7	Fibonacci(7)	7	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(5) + Fibonacci(6) = 5 + 8	13
8	Fibonacci(8)	8	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(6) + Fibonacci(7) = 8 + 13	21
9	Fibonacci(9)	9	Fibonacci(n-2) + Fibonacci(n-1) = Fibonacci(7) + Fibonacci(8) = 13 + 21	34
10				

[문제 3]

2

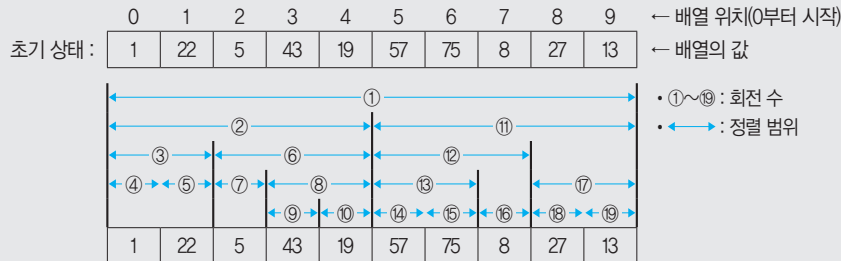
[해설]

- 문제의 코드는 퀵 정렬 알고리즘을 이용하여 배열을 정렬하는 프로그램입니다. 퀵 정렬은 정렬할 대상 중에서 임의의 값을 키로 정한 후 그 키를 기준으로 키 보다 작은 값은 키의 왼쪽으로, 큰 값은 키의 오른쪽으로 옮긴 후 왼쪽과 오른쪽을 서브(Sub) 단위로 하여 각각에 대해 같은 정렬을 반복 수행합니다.
- 문제에서는 분할의 기준이 되는 키는 정렬 범위, 즉 배열의 중간에 위치한 값을 기준 키로 사용했습니다. 배열의 중간 값을 키로 사용했다는 것만 파악할 수 있어도 코드 중 빈 괄호에 들어갈 답이 2라는 것을 유추할 수 있는 문제였습니다.

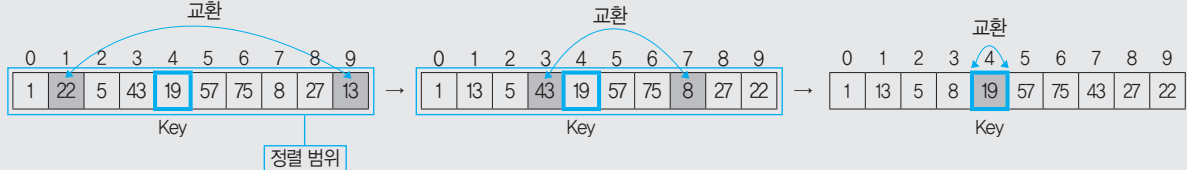
[알고리즘의 이해] 퀵 정렬(Quick Sort)

퀵 정렬은 정렬할 대상 중 임의의 값을 키로 정한 후, 그 키를 기준으로 왼쪽에 있는 요소 중에서는 키 보다 큰 값을 찾고 키의 오른쪽에 있는 요소 중에서는 키 보다 작은 값을 찾아서 서로 교환하여 두 개의 서브 그룹으로 나눕니다. 그러니까 이 작업이 완료되면 왼쪽 그룹에는 키 보다 작은 값만, 그리고 오른쪽 그룹에는 키 보다 큰 값만 있는 두 개의 서브 그룹이 만들어집니다. 이어서 이 두 개의 서브 그룹 각각에 대해서도 키를 정하고 키를 기준으로 작은 값과 큰 값을 서로 교환하면서 각각을 두 개의 서브 그룹으로 나눕니다. 배열 내의 모든 요소에 대해 정렬이 완성될 때까지 이러한 작업을 반복 수행합니다.

문제에 제시된 배열이 퀵 정렬을 통해 정렬되는 순서는 다음과 같습니다.

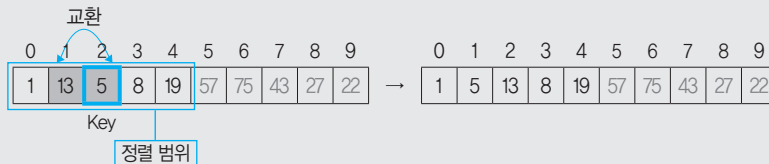


① 1회전 : 배열의 중간에 위치한 값을 키로 정하고, 키를 기준으로 왼쪽에 있는 요소의 첫 번째 요소부터 확인하여 키 보다 큰 값을 찾고, 오른쪽에 있는 요소의 마지막 요소부터 확인하여 키 보다 작은 값을 찾아 서로 교환합니다. 키 위치도 자료 교환에 포함되며, 이러한 교환 작업이 모두 끝나면 키를 기준으로 작은 값과 큰 값이 모여 있는 두 개의 그룹이 형성된다.



※ 마지막 교환이 발생했을 때 그때의 왼쪽 요소의 위치가 다음 정렬 범위의 오른쪽 끝 자리가 됩니다. 그러니까 여기서는 배열의 다섯 번째 요소가 오른쪽 끝자리가 되어 0~4 요소가 다음 회전의 정렬 범위가 됩니다.

② 2회전



※ 두 번째 요소에서 마지막 교환이 발생했으므로 다음 회전의 정렬 범위는 0~1 요소입니다.

③ 3회전

교환

0	1	2	3	4	5	6	7	8	9
1	5	13	8	19	57	75	43	27	22

Key

정렬 범위

※ 첫 번째 요소에서 마지막 교환이 발생했으므로 다음 회전의 정렬 범위는 0~0 요소입니다.

④ 4회전: 정렬 범위의 요소가 1개인 경우는 정렬이 완료된 것이므로 정렬을 수행하지 않고 다음 회전으로 넘어간다.

0	1	2	3	4	5	6	7	8	9
1	5	13	8	19	57	75	43	27	22

정렬 범위

⑤ 5회전: 정렬 범위의 요소가 1개이므로 다음 회전으로 넘어간다.

0	1	2	3	4	5	6	7	8	9
1	5	13	8	19	57	75	43	27	22

정렬 범위

⑥ 6회전: 2회전에서 마지막 교환이 발생했을 때 그때의 오른쪽 요소의 위치가 다음 정렬할 범위의 왼쪽 첫 번째 자리가 된다. 그러니까 여기서는 배열의 세 번째 요소가 왼쪽 첫 번째 자리가 되어 2~4 요소가 이번 회전의 정렬 범위가 된다.

0	1	2	3	4	5	6	7	8	9
1	5	13	8	19	57	75	43	27	22

→

0	1	2	3	4	5	6	7	8	9
1	5	8	13	19	57	75	43	27	22

Key

정렬 범위

※ 세 번째 요소에서 마지막 교환이 발생했으므로 다음 회전의 정렬 범위는 2~2 요소입니다.

⑦ 7회전: 정렬 범위의 요소가 1개이므로 다음 회전으로 넘어갑니다.

0	1	2	3	4	5	6	7	8	9
1	5	8	13	19	57	75	43	27	22

정렬 범위

⑧ 8회전: 6회전에서 마지막 교환이 발생한 네 번째 요소가 다음 정렬할 범위의 왼쪽 첫 번째 자리가 되므로, 3~4 요소가 이번 회전의 정렬 범위가 된다.

0	1	2	3	4	5	6	7	8	9
1	5	8	13	19	57	75	43	27	22

Key

정렬 범위

⑨ 9회전: 정렬 범위의 요소가 1개이므로 다음 회전으로 넘어간다.

0	1	2	3	4	5	6	7	8	9
1	5	8	13	19	57	75	43	27	22

정렬 범위

⑩ 10회전: 정렬 범위의 요소가 1개이므로 다음 회전으로 넘어간다.

0	1	2	3	4	5	6	7	8	9
1	5	8	13	19	57	75	43	27	22

정렬 범위

※ 위와 같은 방식으로 오른쪽 그룹의 정렬 과정인 11~19회전을 수행합니다.

[코드 설명]

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

```
public static void main(String[] args) {  
  ❶ int arr[] = { 1, 22, 5, 43, 19, 57, 75, 8, 27, 13 };  
  ❷ sort(arr, 0, arr.length - 1);  
}
```

❶ 10개의 요소를 갖는 정수형 배열 arr을 선언하고 초기화한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
arr	1	22	5	43	19	57	75	8	27	13

❷ 배열 arr의 주소, 0, arr의 길이(10)에서 1을 뺀 9를 인수로 하여 sort() 메소드를 호출한다.

※ length() : 배열의 길이를 반환하는 메소드

※ 배열의 값이 아닌 주소가 인수로 전달되었으므로 다른 메소드에서의 배열 값의 변화는 main() 메소드의 배열에도 영향을 준다.

```
  ❸ private static void sort(int[] arr, int left, int right) {  
  ❹ int pivot;  
  ❺ if (left < right) {  
  ❻ pivot = partition(arr, left, right);  
    sort(arr, left, pivot - 1);  
    sort(arr, pivot, right);  
  }  
}
```

❸ sort() 메소드의 시작점이다. ❷번에서 전달받은 값을 각각 배열 arr, 정수형 변수 left와 right가 받는다.

❹ 정수형 변수 pivot을 선언한다.

❺ left가 right보다 작으면 중괄호{ } 안의 문장을 수행하고, 아니면 sort() 메소드를 종료하고 호출한 곳으로 제어를 옮긴다.

❻ 배열 arr의 주소, left의 값 0, right의 값 9를 인수로 하여 partition() 메소드를 호출하고 돌려받은 값을 pivot에 저장한다.

```
  ❼ private static int partition(int[] arr, int left, int right) {  
  ❽ int pivot = arr[(left + right) / 2];  
  ❾ int tmp;  
  ❿ while (left <= right) {  
    11 while (arr[left] < pivot)  
    12 left++;  
    13 while (arr[right] > pivot)  
    14 right--;  
    15 if (left <= right) {  
    16 tmp = arr[left];  
    17 arr[left] = arr[right];  
    18 arr[right] = tmp;  
    19 left++;  
    20 right--;  
  }  
  21 return left;  
}
```

❼ partition() 메소드의 시작점이다. ❸번에서 전달받은 값을 각각 배열 arr, 정수형 변수 left와 right가 받는다.

❽ 정수형 변수 pivot을 선언하고 arr[(left + right) / 2]의 값으로 초기화한다. left가 0, right가 9이므로 pivot에는 arr[4]의 값 19가 저장된다.

※ 9 / 2의 결과는 4.5지만 left와 right가 정수형 변수이므로 연산 결과 또한 소수점 이하가 잘려진 정수 4가 된다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	pivot	left	right	tmp
arr	1	22	5	43	19	57	75	8	27	13	19	0	9	

⑨ 정수형 변수 tmp를 선언한다.

⑩ left가 right보다 작거나 같은 동안 ⑪~⑫번을 반복 수행한다. left의 값 0은 right의 값 9보다 작으므로 ⑪번으로 이동한다.

⑪ arr[left]가 pivot보다 작은 동안 ⑫번을 반복 수행한다. arr[left]의 값 1은 pivot의 값 19보다 작으므로 ⑫번으로 이동한다.

⑫ 'left = left + 1;'과 동일하다. left의 값을 1씩 증가시킨다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

left	arr[left]	pivot
0	1	19
1	22	

⑬ arr[right]가 pivot보다 큰 동안 ⑭번을 반복 수행한다. arr[right]의 값 13은 pivot의 값 19보다 작으므로 ⑭번을 수행하지 않고 ⑮번으로 이동한다.

⑭ 'right = right - 1;'과 동일하다. right의 값을 1씩 감소시킨다.

⑮ left가 right보다 작거나 같으면 ⑮~⑯번을 수행한다. left의 값 1은 right의 값 9보다 작으므로 ⑮번으로 이동한다.

⑮~⑯ tmp를 사용하여 arr[left]와 arr[right]의 값을 교환한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	pivot	left	right	tmp
arr	1	22	5	43	19	57	75	8	27	13	19	1	9	22
		13								22				

⑰ 'left = left + 1;'과 동일하다. left의 값을 1 증가시킨다.

⑱ 'right = right - 1;'과 동일하다. right의 값을 1 감소시킨다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	pivot	left	right	tmp
arr	1	13	5	43	19	57	75	8	27	22	19	2	8	22

⑩번 while문의 조건(left < right)가 거짓이 될 때까지 ⑪~⑯번을 반복하면 값의 변화는 다음과 같다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	pivot	left	right	tmp
arr	1	13	5	43	19	57	75	8	27	22	19	2	8	22
	1	13	5	8	19	57	75	43	27	22	19	3	7	43
	1	13	5	8	19	57	75	43	27	22	19	4	6	43
	1	13	5	8	19	57	75	43	27	22	19	4	5	43
	1	13	5	8	19	57	75	43	27	22	19	4	4	19
	1	13	5	8	19	57	75	43	27	22	19	5	3	19

㉑ 반환값으로 left의 값 5를 가지고 partition() 메소드를 호출했던 곳으로 제어를 옮긴다.

```

3 private static void sort(int[] arr, int left, int right){
4     int pivot;
5     if (left < right) {
6         pivot = partition(arr, left, right);
7         sort(arr, left, pivot - 1);
8         sort(arr, pivot, right);
9     }
10 }

```

㉓ ㉒번에서 반환받은 5를 pivot에 저장한다.

※ 메소드 내에서 선언된 변수들은 메소드가 종료하면 모두 소멸한다. sort() 메소드의 변수들과 partition() 메소드의 변수들은 이름은 같아도 서로 다른 변수이다. 하지만 배열의 값을 전달한 것이 아닌 주소를 전달하여 변경하였으므로 partition() 메소드에서의 배열 값의 변화는 sort() 메소드에서의 배열에도 적용된다. 현재 sort의 배열과 변수들의 값은 다음과 같다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]		pivot	left	right
arr	1	13	5	8	19	57	75	43	27	22		5	0	9

㉔ 배열 arr의 주소, left의 값 0, pivot에서 1을 뺀 값 4를 인수로 하여 sort() 메소드를 호출한다.

```

㉔ private static void sort(int[] arr, int left, int right) {
    int pivot;
㉕ if (left < right) {
㉖     pivot = partition(arr, left, right);
        sort(arr, left, pivot - 1);
        sort(arr, pivot, right);
    }
}

```

㉔ ㉔번에서 전달받은 값을 각각 배열 arr, 정수형 변수 left와 right가 받는다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]		left	right
arr	1	13	5	8	19	57	75	43	27	22		0	4

㉕ left가 right보다 작으므로 ㉖번으로 이동한다.

㉖ 배열 arr의 주소, left의 값 0, right의 값 4를 인수로 하여 partition() 메소드를 호출하고 돌려받은 값을 pivot에 저장한다.

※ 여기까지가 위의 퀵 정렬 과정 그림 중 1회전에 해당합니다. 19회전을 마칠 때까지 이러한 과정을 계속 반복해서 수행합니다.

[문제 4]

데이터 결합도, 스템프 결합도, 제어 결합도, 외부 결합도, 공통 결합도, 내용 결합도

[문제 5]

① COMMIT ② ROLLBACK

[문제 6]

※ 다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

기본 릴레이선의 기본키를 구성하는 어떤 속성도 NULL 일 수 없다는 규정

[문제 7]

※ 각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① SELECT
- ② PROJECT
- ③ INTERSECTION, 교집합
- ④ DIFFERENCE, 차집합

[문제 8]

- ① CHECK ② IN

[해설]

- ① CREATE TABLE 사원
② (직원코드 NUMBER NOT NULL,
③ 성명 CHAR(10) UNIQUE,
④ 직책 CHAR(10) CHECK(직책 IN('사원', '대리', '과장', '팀장')),
⑤ 연봉 NUMBER);

- ① <사원> 테이블을 생성한다.
② '직원코드' 속성은 숫자형으로, NULL을 갖지 않는다.
③ '성명' 속성은 최대 문자 10자로, 중복된 값을 가질 수 없다.
④ '직책' 속성은 최대 문자 10자로, "사원", "대리", "과장", "팀장" 중에 하나의 값을 가져야 한다.
⑤ '연봉' 속성은 숫자형이다.

[문제 9]

사설망의 IP 주소와 공중망의 IP 주소를 상호 변환하는 기능

[문제 10]

※ 각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① DES, Data Encryption Standard, 데이터 암호 표준 ② RPO, Recovery Point Objective, 목표 복구 시점
③ OTP, One-Time Password, 일회용 패스워드 ④ DDoS, Distributed Denial of Service, 분산 서비스 거부 공격
⑤ 백도어, Back Door

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 11]

정의, 측정, 분석, 향상, 관리

[문제 12]

- ① 수주 ② 출하지시 ③ 입고 ④ 입하 ⑤ 물류센터

[문제 13]

Linux

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

(Linux)는 Linus B. Torvalds가 개발한 컴퓨터 운영체제이며, 다양한 하드웨어 플랫폼에서 실행되는 매우 인기있는 Unix 운영체제의 한 버전이다. (Linux)는 서버 OS로 널리 사용되고 있으며, 데스크톱 시장에서 우위를 점하고 있다. 또한 그것은 누구나 무료로 사용할 수 있는 오픈 소스 소프트웨어이다.

[문제 14]

※ 제시된 답안 중 한 가지만 쓰면 됩니다.

RAID, Redundant Array of Inexpensive Disks, Redundant Array of Independent Disks

[원문해석]

(RAID)는 데이터 공간이나 성능 향상을 위해 여러 개의 물리적 디스크 드라이브 구성 요소를 하나의 논리 장치로 결합하는 데이터 저장소 가상화 기술이다.



정답 및 해설

2020년 7월 정보처리산업기사 실기

[문제 1]

- ① p+1 ② 101 ③ q×p ④ 100 ⑤ s

※ 답안 작성 시 주의 사항 : 문제의 처리조건으로 연산자에 대한 별도의 언급이 없으면, 곱하기는 x, *, 나누기는 /, ÷ 중 어떤 것을 사용해도 관계 없습니다.

디버깅

p	q	c	s	출력
0	0	0	0	171700
1	100	(100×1)	100	
2	99	(99×2)	298	
3	98	(98×3)	592	
4	97	(97×4)	980	
5	96	(96×5)	1460	
⋮	⋮	⋮	⋮	
96	5	(5×96)	170720	
97	4	(4×97)	171108	
98	3	(3×98)	171402	
99	2	(2×99)	171600	
100	1	(1×100)	171700	

[문제 2]

76543210

5467

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에逗를 넣거나 줄 나눔을 하지 않았을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[해설]

이 프로그램은 자기가 자기를 호출하는 순환 프로그램의 순환 과정입니다. 순환 프로그램은 순환하는 만큼 반복하여 실행하면서 변수에 저장된 값을 추적하면 결과를 이해하기 쉽습니다.

```
main(){
  ① int n;
  ② n = recur(7);
  printf("\n");
  printf("%d", n);
}
```

① 정수형 변수 n을 선언한다.

② recur(7) 함수를 호출하고 반환받은 값을 n에 저장한다.

```
recur(i) {
  ③ printf("%d", i);
  ④ if (i < 1)
    return 2;
  else
  ⑤   return 3 * recur(i - 1) + 1;
}
```

①회

recur() 함수가 호출될 때 70이 전달되었으므로 i는 70이다. ③번을 통해 i의 값 7을 화면에 출력하고, ④번의 조건을 만족하지 않으므로 ⑤번을 수행한다. ⑤번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(6)인 상태로 호출된다.

결과 7

```
recur(i) {
  ⑥ printf("%d", i);
  ⑦ if (i < 1)
    return 2;
  else
  ⑧   return 3 * recur(i - 1) + 1;
}
```

②회

recur() 함수가 호출될 때 60이 전달되었으므로 i는 60이다. ⑥번을 통해 i의 값 6을 화면에 출력하고, ⑦번의 조건을 만족하지 않으므로 ⑧번을 수행한다. ⑧번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(5)인 상태로 호출된다.

결과 76

```
recur(i) {
  ⑨ printf("%d", i);
  ⑩ if (i < 1)
    return 2;
  else
  ⑪   return 3 * recur(i - 1) + 1;
}
```

③회

recur() 함수가 호출될 때 5가 전달되었으므로 i는 5이다. ⑨번을 통해 i의 값 5를 화면에 출력하고, ⑩번의 조건을 만족하지 않으므로 ⑪번을 수행한다. ⑪번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(4)인 상태로 호출된다.

결과 765

```
recur(i) {
  ⑫ printf("%d", i);
  ⑬ if (i < 1)
    return 2;
  else
  ⑭   return 3 * recur(i - 1) + 1;
}
```

④회

recur() 함수가 호출될 때 4가 전달되었으므로 i는 4이다. ⑫번을 통해 i의 값 4를 화면에 출력하고, ⑬번의 조건을 만족하지 않으므로 ⑭번을 수행한다. ⑭번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(3)인 상태로 호출된다.

결과 7654

```
recur(i) {
  ⑮ printf("%d", i);
  ⑯ if (i < 1)
    return 2;
  else
  ⑰   return 3 * recur(i - 1) + 1;
}
```

⑤회

recur() 함수가 호출될 때 3이 전달되었으므로 i는 3이다. ⑮번을 통해 i의 값 3을 화면에 출력하고, ⑮번의 조건을 만족하지 않으므로 ⑰번을 수행한다. ⑰번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(2)인 상태로 호출된다.

결과 76543

⑥회

```
recur(i){
  ⑮ printf("%d", i);
  ⑰ if (i < 1)
      return 2;
  else
    ⑲ return 3 * recur(i - 1) + 1;
}
```

recur() 함수가 호출될 때 2가 전달되었으므로 i는 2이다. ⑮번을 통해 i의 값 2를 화면에 출력하고, ⑮번의 조건을 만족하지 않으므로 ⑲번을 수행한다. ⑲번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(1)인 상태로 호출된다.

결과 765432

⑦회

```
recur(i){
  ⑲ printf("%d", i);
  ⑲ if (i < 1)
      return 2;
  else
    ㉑ return 3 * recur(i - 1) + 1;
}
```

recur() 함수가 호출될 때 1이 전달되었으므로 i는 1이다. ㉑번을 통해 i의 값 1을 화면에 출력하고, ㉑번의 조건을 만족하지 않으므로 ㉑번을 수행한다. ㉑번을 수행하기 위해 recur() 함수를 호출하는데, 호출할 때 전달되는 값은 i-1이므로 recur(0)인 상태로 호출된다.

결과 7654321

⑧회

```
recur(i){
  ㉑ printf("%d", i);
  ㉑ if (i < 1)
      return 2;
  else
    ㉓ return 3 * recur(i - 1) + 1;
}
```

recur() 함수가 호출될 때 0이 전달되었으므로 i는 0이다. ㉑번을 통해 i의 값 0을 화면에 출력하고, ㉑번의 조건을 만족하므로 ㉑번을 수행한다. 'return 2'이므로 함수의 실행을 종료하고 2를 반환하면서 제어를 ⑦회 recur(0) 함수를 호출했던 곳으로 옮긴다.

결과 76543210

⑦회

```
recur(i){
  printf("%d", i);
  if (i < 1)
    return 2;
  else
    ㉓ return 3 * recur(i - 1) + 1;
}
```

앞선 과정에서 2가 반환되었으므로

㉓ return (3 * recur(i-1) + 1)

㉓

㉓

- ㉓ : 2 (㉑번에서 2가 반환되었으므로)

- ㉓ : 7

결과 ㉓번은 'return 3*2+1', 즉 'return 7;'이므로 함수의 실행을 종료하고 7을 반환하면서 제어를 ⑥회 recur(1) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {
    printf("%d", i);
    if (i < 1)
        return 2;
    else
        ㉔ return 3 * recur(i - 1) + 1;
}
```

⑥회

앞선 과정에서 7이 반환되었으므로

㉔ return (3 * recur(i-1) + 1)

③

④

- ③ : 7 (㉔번에서 7이 반환되었으므로)

- ④ : 22

결국 ㉔번은 'return 3*7+1', 즉 'return 22'이므로 함수의 실행을 종료하고 22를 반환하면서 제어를 ⑤회 recur(2) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {
    printf("%d", i);
    if (i < 1)
        return 2;
    else
        ㉔ return 3 * recur(i - 1) + 1;
}
```

⑤회

앞선 과정에서 22가 반환되었으므로

㉔ return (3 * recur(i-1) + 1)

③

④

- ③ : 22 (㉔번에서 22가 반환되었으므로)

- ④ : 67

결국 ㉔번은 'return 3*22+1', 즉 'return 67'이므로 함수의 실행을 종료하고 67을 반환하면서 제어를 ④회 recur(3) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {
    printf("%d", i);
    if (i < 1)
        return 2;
    else
        ㉔ return 3 * recur(i - 1) + 1;
}
```

④회

앞선 과정에서 67이 반환되었으므로

㉔ return (3 * recur(i-1) + 1)

③

④

- ③ : 67 (㉔번에서 67이 반환되었으므로)

- ④ : 202

결국 ㉔번은 'return 3*67+1', 즉 'return 202'이므로 함수의 실행을 종료하고 202를 반환하면서 제어를 ③회 recur(4) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {
    printf("%d", i);
    if (i < 1)
        return 2;
    else
        ㉔ return 3 * recur(i - 1) + 1;
}
```

③회

앞선 과정에서 202가 반환되었으므로

㉓ return (3 * recur(i-1) + 1)

㉑

㉒

- ㉑ : 202 (㉓번에서 202가 반환되었으므로)

- ㉒ : 607

결국 ㉓번은 'return 3*7+1;', 즉 'return 607;'이므로 함수의 실행을 종료하고 607을 반환하면서 제어를 ㉒회 recur(5) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {  
    printf("%d", i);  
    if (i < 1)  
        return 2;  
    else  
        return 3 * recur(i - 1) + 1;  
}
```

㉒회

앞선 과정에서 607이 반환되었으므로

㉔ return (3 * recur(i-1) + 1)

㉑

㉒

- ㉑ : 607 (㉓번에서 607이 반환되었으므로)

- ㉒ : 1822

결국 ㉔번은 'return 3*607+1;', 즉 'return 1822;'이므로 함수의 실행을 종료하고 1822를 반환하면서 제어를 ㉑회 recur(6) 함수를 호출했던 곳으로 옮긴다.

```
recur(i) {  
    printf("%d", i);  
    if (i < 1)  
        return 2;  
    else  
        return 3 * recur(i - 1) + 1;  
}
```

㉑회

앞선 과정에서 1822가 반환되었으므로

㉕ return (3 * recur(i-1) + 1)

㉑

㉒

- ㉑ : 1822 (㉓번에서 1822가 반환되었으므로)

- ㉒ : 5467

결국 ㉕번은 'return 3*7+1;', 즉 'return 5467;'이므로 함수의 실행을 종료하고 5467을 반환하면서 제어를 main()의 recur(7) 함수를 호출했던 곳으로 옮긴다.

```
main() {  
    int n;  
    ㉔ n = recur(7);  
    ㉕ printf("\n");  
    ㉖ printf("%d", n);  
}
```

㉔ ㉓번에서 반환받은 값 5467을 n에 저장한다.

㉕ 커서를 다음 줄로 이동시킨다.

결과 76543210

㉖ 화면에 n의 값을 출력한다.

결과 76543210
5467

재귀 함수 실행에 따른 변수들의 값의 변화는 다음과 같다.

i(recur)	리턴값(recur)
7	$3 * \text{recur}(6) + 1 = 3 * 1822 + 1 = 5467$
6	$3 * \text{recur}(5) + 1 = 3 * 607 + 1 = 1822$
5	$3 * \text{recur}(4) + 1 = 3 * 202 + 1 = 607$
4	$3 * \text{recur}(3) + 1 = 3 * 67 + 1 = 202$
3	$3 * \text{recur}(2) + 1 = 3 * 22 + 1 = 67$
2	$3 * \text{recur}(1) + 1 = 3 * 7 + 1 = 22$
1	$3 * \text{recur}(0) + 1 = 3 * 2 + 1 = 7$
0	2

[문제 3]

3

[해설]

```
public class Test {
    public static void main(String[] args) {
        ❶ int exint[] = { 2,4,2,67,6,4,7,2,5,4,6 };
        ❷ int value = 0;
        ❸ for (int i = 0; i < exint.length; i++)
        ❹     if (exint[i] == 2)
        ❺         value++;
        ❻ System.out.println(value);
    }
}
```

- ❶ 11개의 요소를 갖는 정수형 배열 exint를 선언하고 초기화한다.
- ❷ 정수형 변수 value를 선언하고 0으로 초기화한다.
- ❸ 반복 변수 i가 1씩 증가하면서 배열 exint의 길이(11)보다 작은 동안 ❹~❺번 문장을 반복 수행한다.
- length : 배열의 길이를 반환한다. 배열 exint는 11개의 요소를 가지므로 11을 반환한다.
- ❹ exint[i]가 2이면 ❺번을 수행한다.
- ❺ 'value = value + 1;'과 동일하다. value에 1을 더한다.
- ❻ 화면에 value의 값을 출력하고 커서를 다음 줄로 이동한다.

디버깅

	exint[0]	exint[1]	exint[2]	exint[3]	exint[4]	exint[5]	exint[6]	exint[7]	exint[8]	exint[9]	exint[10]
exint[]	2	4	2	67	6	4	7	2	5	4	6

i	exint[i]	exint[i] == 2	value	출력
0	2	YES	0	3
1	4	NO	1	
2	2	YES	2	
3	67	NO	3	
4	6	NO		
5	4	NO		
6	7	NO		
7	2	YES		
8	5	NO		
9	4	NO		
10	6	NO		
11				

[문제 4]

- ① SET ② WHERE

[풀이]

UPDATE 사원

SET 연봉 = 연봉 + 100000

WHERE 직급 = '차장';

〈사원〉 테이블을 갱신하라.

'연봉' 속성에 100000을 더하라.

직급이 '차장'인 튜플만을 대상으로 하라.

[문제 5]

- (1) 5 (2) 3

[문제 6]

※ 제시된 답안 중 한 가지만 쓰면 됩니다.

관계대수, Relational Algebra

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 7]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① WEP, Wired Equivalent Privacy
② 스니핑, Sniffing

[문제 8]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 기가파이, Giga Fi
- ② VoIP 보안 위협, VoIP Security Threat
- ③ 다중경로 페이딩, Multipath Fading
- ④ STTD, Space Time Transmit Diversity
- ⑤ VHT, Very High Throughput

[문제 9]

※ 제시된 답안 중 한 가지만 쓰면 됩니다.

ERP, Enterprise Resource Planning

[문제 10]

※ 제시된 답안 중 한 가지만 쓰면 됩니다.

기계 학습, 머신 러닝, Machine Learning

[문제11]

※ 각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① SCM, Supply Chain Management, 공급망 관리, 공급 사슬 관리
- ② RFID, Radio Frequency Identification, 전파 식별

문제의 지문이 긴 관계로 일부만 복원이 되었습니다. 실제 시험에서는 총 4문항이 출제되었으며 세 번째와 네 번째 문항은 지문을 읽고 제시된 보기에서 하나를 골라 작성하는 문제였는데, 두 문항 모두 제시된 지문에서 쉽게 답을 찾을 수 있었습니다.

[문제12]

- ① ISP ② Domain Name ③ E-Mail ④ Modem ⑤ Telnet

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

(ISP)는 개인이나 회사들이 인터넷에 접속하여 가상 호스팅, 웹 사이트 구축 등을 할 수 있도록 관련 서비스를 제공하는 회사이다. 일반적으로 (ISP)가 제공하는 인터넷 서비스에는 인터넷 접속, 인터넷 중계, 도메인 이름 등록, 웹 호스팅 등이 있다.

(Domain Name)은 숫자로 된 IP 주소를 사람이 이해하기 쉬운 문자 형태로 표현한 것이다. (Domain Name)은 컴퓨터의 이름과 호스트 서버가 위치한 조직 또는 국가의 타입으로 구성된다. (Domain Name)은 일반적으로 오른쪽에서 왼쪽으로 읽으며, 가장 오른쪽 부분이 최상위 도메인에 해당한다.

(E-Mail)은 텍스트 메시지와 임의의 파일을 네트워크를 통해 전송하는 것을 말한다. (E-Mail)은 네트워크 (E-Mail) 서버나 호스트 컴퓨터의 가상 우편함에 보내진다. 컴퓨터 내의 (E-Mail) 프로그램은 (E-Mail) 서버를 자주 조회하여 새로운 (E-Mail)이 도착하면 사용자에게 알려준다.

(Modem)은 부호화된 디지털 정보를 전송하기 위해 아날로그 신호로 변조하고 수신된 정보를 원래의 신호로 복조하는 장치이다. 목적은 전송이 쉽고 원래의 디지털 데이터로 복원할 수 있는 신호를 만드는 것이다.

(TELNET)은 인터넷과 같은 TCP/IP 네트워크를 위한 단말기 모방(Terminal Emulation) 프로그램이다. (TELNET) 프로그램은 PC에서 실행되어 PC를 네트워크 상의 서버와 연결시켜준다. (TELNET)을 통해 명령을 입력할 수 있으며, 명령을 입력하면 서버 콘솔에서 직접 입력한 것과 같이 실행된다.

(TELNET) 세션을 시작하려면 반드시 유효한 사용자 이름과 암호를 입력하여 서버에 로그인해야 한다. (TELNET)은 웹 서버를 원격으로 제어하기 위한 일반적인 방식이다.



정답 및 해설

2020년 5월 정보처리산업기사 실기

[문제 1]

- ① 0 ② K ③ N/K ④ 1 ⑤ C

디버깅

다음은 N에 30이 입력되었다고 가정한 디버깅 결과입니다.

N	C	K	MOD(N, K)	S[C]	Y	출력
30	0	2	0	2	1	2×
15	1	2	1	3	2	3×
5	2	3	0	5	3	5
1	3	2	1			
		3	2			
		4	1			
		5	0			

[문제 2]

- ① $n < 1$ 또는 $n == 0$ 또는 $n == 1$ ② $n * \text{factorial}(n - 1)$

※ 답안 작성 시 주의 사항 : C언어 코드 형식으로 쓰라는 조건이 있으므로 C언어 코드 형식에 맞게 답안을 작성해야 합니다. 예를 들어, C언어의 변수명은 대소문자를 구분하므로 n 을 N 과 같이 대문자로 작성하거나 '같다'는 의미의 관계 연산자를 ' $==$ '이 아닌 ' $=$ '으로 작성하면 오답으로 처리됩니다.

[해설]

이 프로그램은 자기가 자기를 호출하는 순환 프로그램의 순환 과정입니다. 순환 프로그램은 순환하는 만큼 반복하여 실행하면서 변수에 저장된 값을 추가하면 결과를 이해하기 쉽습니다.

```
main() {
  ❶ factorial(1);
  factorial(2);
  factorial(3);
  factorial(4);
  factorial(7);
}
```

❶ main() 함수에서 factorial(1) 함수를 호출한다.

```
factorial(int n) {
  ❷ if (n <= 1)
  ❸   return 1;
  else
    return (n * factorial(n - 1));
  ①회 }
```

factorial() 함수가 호출될 때 1이 전달되었으므로 n 은 1이다. ❷번의 조건을 만족하므로 ❸번을 수행한다. 'return 1;'이므로 함수의 실행을 종료하고 1을 반환하면서 제어를 처음 factorial(1) 함수를 호출했던 ❶번으로 옮긴 후 main 함수의 이후 문장을 수행한다.

```
main(){
    factorial(1);
    ④ factorial(2);
    factorial(3);
    factorial(4);
    factorial(7);
}
```

④ main() 함수에서 factorial(2) 함수를 호출한다.

```
factorial(int n){
    ⑤ if ( n <= 1 )
        return 1;
    else
    ⑥ return ( n * factorial(n - 1) );
}
```

①회

factorial() 함수가 호출될 때 2가 전달되었으므로 n은 2이다. ⑤번의 조건을 만족하지 않으므로 ⑥번을 수행한다. ⑥번을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 n-1이므로 factorial(1)인 상태로 호출된다.

```
factorial(int n){
    ⑦ if ( n <= 1 )
    ⑧ return 1;
    else
        return ( n * factorial(n - 1) );
}
```

②회

factorial() 함수가 호출될 때 1이 전달되었으므로 n은 1이다. ⑦번의 조건을 만족하므로 ⑧번을 수행한다. 'return 1;'이므로 함수의 실행을 종료하고 1을 반환하면서 제어를 ①회 factorial(1) 함수를 호출했던 ⑥번으로 옮긴다.

```
factorial(int n){
    if ( n <= 1 )
        return 1;
    else
    ⑨ return ( n * factorial(n - 1) );
}
```

③회

앞선 과정에서 1이 반환되었으므로

⑨ return n * factorial(n-1)

③ ④

- ③ : 2 (⑥번을 수행하기 위해 factorial(1) 함수를 호출할 때 n은 2였으므로)

- ④ : 1 (⑧번에서 1이 반환되었으므로)

결국 ⑨번은 'return (2*1);', 즉 'return 2;'이므로 함수의 실행을 종료하고 2를 반환하면서 제어를 처음 factorial(2) 함수를 호출했던 ④번으로 옮긴 후 이후 문장을 수행한다.

```
main(){
    factorial(1);
    factorial(2);
    ⑩ factorial(3);
    factorial(4);
    factorial(7);
}
```

⑩ main() 함수에서 factorial(3) 함수를 호출한다.

```
factorial(int n){
  ⑪ if ( n <= 1 )
      return 1;
  else
  ⑫   return ( n * factorial(n - 1) );
}
```

①회

factorial() 함수가 호출될 때 3이 전달되었으므로 n은 3이다. ⑪번의 조건을 만족하지 않으므로 ⑫번을 수행한다. ⑫번을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 n-1이므로 factorial(2)인 상태로 호출된다.

```
factorial(int n){
  ⑬ if ( n <= 1 )
      return 1;
  else
  ⑭   return ( n * factorial(n - 1) );
}
```

②회

factorial() 함수가 호출될 때 2가 전달되었으므로 n은 2이다. ⑬번의 조건을 만족하지 않으므로 ⑭번을 수행한다. ⑭번을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 n-1이므로 factorial(1)인 상태로 호출된다.

```
factorial(int n){
  ⑮ if ( n <= 1 )
  ⑯   return 1;
  else
      return ( n * factorial(n - 1) );
}
```

③회

factorial() 함수가 호출될 때 1이 전달되었으므로 n은 1이다. ⑮번의 조건을 만족하므로 ⑯번을 수행한다. 'return 1;'이므로 함수의 실행을 종료하고 1을 반환하면서 제어를 ②회 factorial(1) 함수를 호출했던 곳으로 옮긴다.

```
factorial(int n){
    if ( n <= 1 )
        return 1;
    else
  ⑰   return ( n * factorial(n - 1) );
}
```

②회

앞선 과정에서 1이 반환되었으므로

⑰ return n * factorial(n-1)

② ①

- ② : 2 (⑭번을 수행하기 위해 factorial(1) 함수를 호출할 때 n은 2였으므로)

- ① : 1 (⑯번에서 1이 반환되었으므로)

결국 ⑰번은 'return (2*1);', 즉 'return 2;'이므로 함수의 실행을 종료하고 2를 반환하면서 제어를 ①회 factorial(2) 함수를 호출했던 곳으로 옮긴다.

```
factorial(int n){
    if ( n <= 1 )
        return 1;
    else
  ⑱   return ( n * factorial(n - 1) );
}
```

①회

앞선 과정에서 2가 반환되었으므로

⑱ return n * factorial(n-1)

② ①

- ② : 3 (⑫번을 수행하기 위해 factorial(2) 함수를 호출할 때 n은 3이었으므로)

- ① : 2 (⑰번에서 2가 반환되었으므로)

결국 ⑩번은 'return (3*2);', 즉 'return 6;'이므로 함수의 실행을 종료하고 6을 반환하면서 제어를 처음 factorial(3) 함수를 호출했던 ⑩번으로 옮긴 후 이후 문장을 수행한다.

이후의 factorial(4)와 factorial(7)도 자기가 자기를 호출하는 순환 프로그램의 순환 과정입니다. 앞선 과정처럼 순환하는 만큼 반복하여 실행하면서 변수에 저장된 값을 추적하면서 과정을 이해해 보세요.

n (factorial)	리턴값 (factorial)
1	1
2	$2 * \text{factorial}(1) = 2 * 1 = 2$
3	$3 * \text{factorial}(2) = 3 * 2(2*1) = 6$
4	$4 * \text{factorial}(3) = 4 * 6(3*2*1) = 24$
7	$7 * \text{factorial}(6) = 7 * 720(6*5*4*3*2*1) = 5040$

[문제 3]

2

디버깅

n	출력
17	2
18	
16	
48	
12	
2	

[문제 4]

- ① 기본키 ② 외래키 ③ 접수종류 ④ 무결성 ⑤ PROJECT

[문제 5]

- ① 논리적 ② 물리적

[문제 6]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 128 ② QoS, Quality of Service, 서비스 품질

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 7]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 에지, EDGE, Enhanced Data Rates For Global Evolution ② 비트코인, bitcoin
 ③ 고화질 멀티미디어 인터페이스, HDMI, High-Definition Multimedia Interface ④ 인터넷 전화, VoIP, Voice over Internet Protocol
 ⑤ 미라캐스트, Miracast

[문제 8]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 저작권, copyright ② DRM, Digital Rights Management, 디지털 저작권 관리 ③ copyleft

[문제 9]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① B2B ② C2C

문제의 지문이 긴 관계로 일부만 복원이 되었습니다. 실제 시험에서는 총 3문항이 출제되었으며, 세 번째 문항은 제시된 지문에서 답을 찾는 문제였습니다.

[문제 10]

- ① Copyright ② C ③ ROM ④ Firewall ⑤ Compiler

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

(저작권)이란 원저작물의 창작자가 저작물의 사용과 배포에 있어 일반적으로 제한된 시간 동안 독점적인 권리를 인정하는 법적 권리이다. 배타적 권리는 (저작권)법에서 정한 제한 및 예외 사항에 의해 제한되므로 절대적인 권리는 아니다.

전산에 있어 (C)는 1969년에서 1973년 사이 AT&T 벨연구소에서 Dennis Ritchie가 최초 개발한 범용 프로그래밍 언어이다. ALGOL 계보의 다른 대부분의 명령어와 마찬가지로 (C)는 구조적 프로그래밍을 위한 기능을 가지고 있으며, 정적 시스템이 여러 의도치 않은 작업을 예방하는 한편 어휘형 변수와 재귀를 허용한다.

(ROM)은 컴퓨터 및 다른 전자 장치에서 사용하는 저장 매체의 한 부류이다. (ROM)에 저장된 데이터는 수정이 느리거나 어렵거나 전혀 할 수 없기 때문에 주로 펌웨어를 배포하기 위해 사용된다.

(방화벽(Firewall))은 침입자로부터 컴퓨터 안전을 지키는 주요한 수단이다. (방화벽)은 사실 네트워크나 사용자 컴퓨터로부터 들어오거나 나가는 트래픽을 허용하거나 차단한다.

(컴파일러)는 고급 프로그래밍 언어(C/C++, COBOL 등)로 작성된 프로그램을 기계어로 바꾸어주는 소프트웨어다. (컴파일러)는 보통 어셈블리 언어를 생성하고 그 다음 어셈블리 언어를 기계어로 변환한다.



[문제 1]

- ① $C = C + SW$ ② $ROT = ROT - 1$ ③ $R = R + SW$ ④ $HEY(R, C)$ ⑤ $SW \times (-1)$

디버깅

SW	ROT	R	C	P	N	배열 HEY																									
1	5	1	0 1 2 3 4 5	1 2 3 4 5 6	0 1 2 3 4 5	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>16</td><td>17</td><td>18</td><td>19</td><td>6</td></tr><tr><td>15</td><td>24</td><td>25</td><td>20</td><td>7</td></tr><tr><td>14</td><td>23</td><td>22</td><td>21</td><td>8</td></tr><tr><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr></table>	1	2	3	4	5	16	17	18	19	6	15	24	25	20	7	14	23	22	21	8	13	12	11	10	9
	1	2	3	4	5																										
16	17	18	19	6																											
15	24	25	20	7																											
14	23	22	21	8																											
13	12	11	10	9																											
4	2 3 4 5		1 2 3 4 5	6 7 8 9																											
-1			4 3 2 1	1 2 3 4 5	10 11 12 13																										
	3	4 3 2		1 2 3 4	14 15 16																										
1			2 3 4	1 2 3 4	17 18 19																										
	2	3 4		1 2 3	20 21																										
-1			3 2	1 2 3	22 23																										
	1	3		1 2	24																										
1			3	1 2	25																										
	0																														

[문제 2]

```
1  2  3  4  5
   6  7  8  9
     10 11 12
       13 14
        15
```

[해설]

```
public class Test {
    public static void main(String[] args) {
        ❶ int a[][] = new int[5][5];
        ❷ initArray(a);
        ❸ assignVal(a);
        ❹ prntArray(a);
    }

    ❺ static void initArray(int[][] a) {
        ❻ for(int i = 0; i < a.length; i++) {
            ❼ for(int j = 0; j < a[i].length; j++)
                ❽ a[i][j] = 0;
        }
    }

    ❾ static void assignVal(int[][] a) {
        ❿ int n = 1;
        ⓫ for(int i = 0; i < a.length; i++) {
            ⓬ for(int j = i; j < a[i].length; j++)
                ⓭ a[i][j] = n++;
        }
    }

    ⓮ static void prntArray(int[][] a) {
        ⓯ for(int i = 0; i < a.length; i++) {
            ⓰ for(int j = 0; j < a[i].length; j++)
                ⓱ if(a[i][j] == 0)
                    ⓲ System.out.print(" ");
                else
                    ⓳ System.out.printf("%3d", a[i][j]);
            ⓴ System.out.println();
        }
    }
}
```

- ❶ 5행 5열의 2차원 배열 `a`를 선언합니다.
- ❷ 배열 `a`를 인수로 하여 `initArray()` 메소드를 호출합니다.
- ❸ `initArray()` 메소드의 시작점이다. 인수로 받은 배열 `a`를 동일한 이름의 배열 `a`로 선언합니다.
- ❹ 정수형 변수 `i`를 반복 변수로 선언하면서 초기값으로 0을 갖고, 1씩 증가시키면서 배열 `a`의 행의 개수보다 작은 동안 ❺번을 반복하여 수행합니다.
 - `length` : 배열의 길이가 저장된 속성
 - `a.length` : 2차원 배열의 길이는 행의 개수를 의미한다. 배열 `a`는 5행으로 이루어져 있으므로 5가 됩니다.
 - `a[i].length` : 1차원 배열의 길이는 요소의 개수를 의미한다. `a[i]`는 배열 `a`의 한 행으로, 하나의 행은 1차원 배열과 같으므로 `a[i]`에 속하는 `a[i][0]~a[i][4]`의 개수인 5가 `a[i].length`의 값이 됩니다.

- ⑤ 정수형 변수 `j`를 반복 변수로 선언하면서 초기값으로 0을 갖고, 1씩 증가시키면서 배열 `a[i]` 행의 요소의 개수보다 작은 동안 ⑥번을 반복하여 수행합니다.
 ⑥ `a[i][j]`에 0을 저장합니다. ④, ⑤번의 반복이 종료되면 제어가 ⑦번으로 이동합니다.
 ⑦ 배열 `a`를 인수로 하여 `assignVal()` 메소드를 호출합니다.
 ⑧ `assignVal()` 메소드의 시작점입니다. 인수로 받은 배열 `a`를 동일한 이름의 배열 `a`로 선언합니다.
 ⑨ 정수형 변수 `n`을 선언하고 1로 초기화합니다.
 ⑩ 정수형 변수 `i`를 반복 변수로 선언하면서 초기값으로 0을 갖고, 1씩 증가시키면서 배열 `a`의 행의 개수보다 작은 동안 ⑪번을 반복하여 수행합니다.
 ⑪ 정수형 변수 `j`를 반복 변수로 선언하면서 초기값으로 `i` 값을 갖고, 1씩 증가시키면서 배열 `a[i]` 행의 요소의 개수 보다 작은 동안 ⑫번을 반복하여 수행합니다.
 ⑫ `a[i][j]`에 `n`의 값을 저장하고, `n`의 값을 1 증가시킵니다. ⑩, ⑪번의 반복이 종료되면 제어가 ⑬번으로 이동합니다.

• 디버깅

i	j	n	배열 a																													
0	0	1	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>0</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>0</td><td>0</td><td>10</td><td>11</td><td>12</td></tr><tr><td>0</td><td>0</td><td>0</td><td>13</td><td>14</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>15</td></tr></table>					1	2	3	4	5	0	6	7	8	9	0	0	10	11	12	0	0	0	13	14	0	0	0	0	15
	1	2						3	4	5																						
	0	6						7	8	9																						
	0	0						10	11	12																						
	0	0						0	13	14																						
	0	0						0	0	15																						
1	2																															
2	3																															
3	4																															
4	5																															
1	1	6																														
	2	7																														
	3	8																														
	4	9																														
	5																															
2	2	10																														
	3	11																														
	4	12																														
	5																															
	3	3						13																								
4		14																														
5																																
4	4	15																														
	5	16																														
5																																

- ⑬ 배열 `a`를 인수로 하여 `prntArray()` 메소드를 호출합니다.
 ⑭ `prntArray()` 메소드의 시작점입니다. 인수로 받은 배열 `a`를 동일한 이름의 배열 `a`로 선언합니다.
 ⑮ 정수형 변수 `i`를 반복 변수로 선언하면서 초기값으로 0을 갖고, 1씩 증가시키면서 배열 `a`의 행의 개수보다 작은 동안 ⑯번을 반복하여 수행합니다.
 ⑯ 정수형 변수 `j`를 반복 변수로 선언하면서 초기값으로 0을 갖고, 1씩 증가시키면서 배열 `a[i]` 행의 요소의 개수 보다 작은 동안 ⑰번을 반복하여 수행합니다.
 ⑰ `a[i][j]`가 0이면 ⑱번 문장을 실행하고, 아니면 ⑲번 문장을 실행합니다.
 ⑱ 공백 3칸을 출력합니다.
 • `System.out.print()`: 화면 출력을 위한 메소드로, 줄 바꿈 없이 커서가 있는 위치부터 출력됩니다. `println()`과 동일하게 제어문자('\t', '\n' 등)는 사용할 수 있지만, 서식 지정자('%d', '%f' 등)는 사용할 수 없습니다.
 ⑲ `a[i][j]`의 값을 3자리를 확보한 후 오른쪽부터 출력합니다.
 ⑳ 다음 행을 출력하기 위해 커서를 다음 줄의 처음으로 옮깁니다.
 ㉑ ⑮, ⑯번의 반복이 종료되면 `prntArray(a)` 메소드를 호출했던 `main()` 메소드로 제어를 옮긴 후 `main()` 메소드의 종료 브레이스 `}`를 만나 프로그램을 종료합니다.

[문제 3]

recursive(n - 1)

디버깅

main() 함수		recursive() 함수	출력
i	n	i	
5	5		
	4		
	3		
	2		
	1		
	0		
		5 = 2 * 2 + 1	5
		11 = 2 * 5 + 1	11
		23 = 2 * 11 + 1	23
		47 = 2 * 23 + 1	47
		95 = 2 * 47 + 1	95

[해설]

모든 C 프로그램은 반드시 main() 함수부터 시작해야 합니다.

```
main(void) {
  ❶ int i;
  ❷ printf("숫자를 입력하시오: ");
  ❸ scanf("%d", &i);
  ❹ recursive(i);
}
```

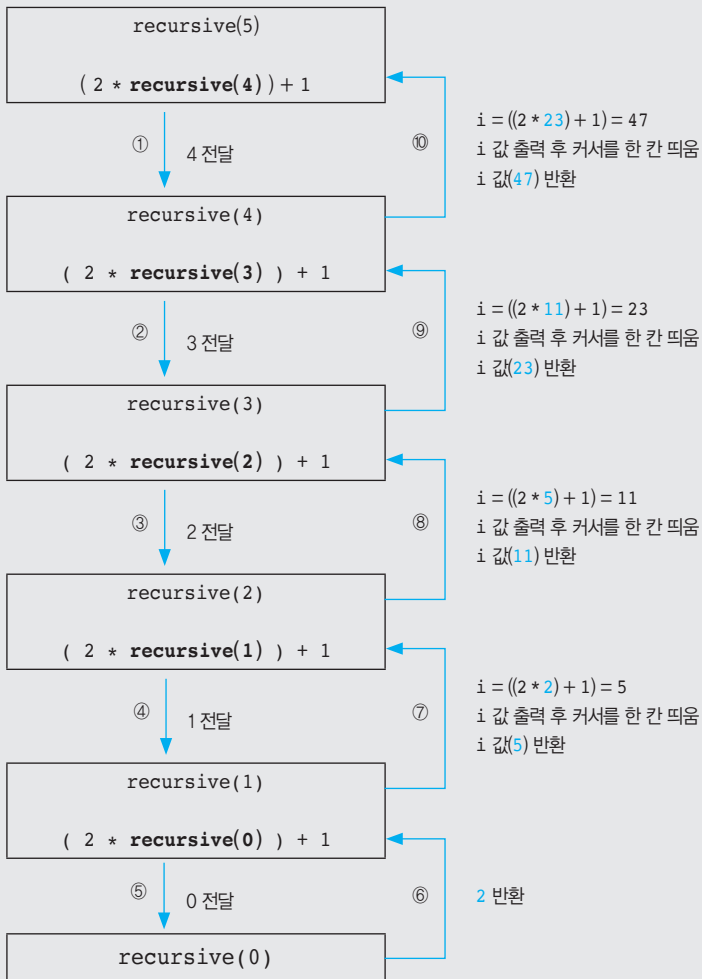
- ❶ 정수형 변수 i를 선언합니다.
- ❷ "숫자를 입력하시오: "를 출력합니다.
- ❸ 키보드로 숫자를 입력받아 그 값을 i에 저장합니다. 5가 입력되었다고 가정하였으므로 i에 5가 저장됩니다.
- ❹ i의 값을 인수로 하여 recursive() 함수를 호출합니다. i가 5이므로 recursive(5)로 호출합니다.

```
int recursive(int n) {
  ❶ int i;
  ❷ if (n < 1)
    return 2;
  else {
    ❸ i = (2 * recursive(n - 1)) + 1;
    ❹ printf("%d ", i);
    ❺ return i;
  }
}
```

recursive() 함수가 호출될 때 5를 전달받았으므로 n은 5입니다.

- ❶ 정수형 변수 i를 선언합니다.
- ❷의 조건을 만족하지 않으므로 ❸을 수행합니다.

③ $(2 * \text{recursive}(n-1) + 1)$ 을 수행한 후 결과를 i 에 저장합니다. 지금부터는 자신이 자신을 호출하는 재귀 함수를 이용합니다. n 이 1보다 크거나 같을 때까지 자신을 호출하는 과정이 수행되다 n 이 0이 될 때 2가 반환되면서 호출했던 과정을 복귀할 때 반환된 값을 이용해 i 값을 계산하여 출력한 후 i 값을 반환한다는 것을 염두에 두고 과정을 개괄적인 그림을 통해 살펴봅니다.



③ $(2 * \text{recursive}(n-1) + 1) = (2 * 47) + 1$ 의 결과인 95를 i 에 저장합니다.

④ i 값 출력 후 커서를 한 칸 띄웁니다.

⑤ 반환값 95를 가지고 $\text{recursive}(5)$ 함수를 호출했던 $\text{main}()$ 함수로 제어를 옮긴 후 $\text{main}()$ 함수의 종료 브레이스 $\{ \}$ 를 만나 프로그램을 종료합니다.

[문제 4]

- DDL : Create, Alter, Drop
- DML : Select, Update, Insert, Delete

[문제 5]

※ 다음에 제시된 답안 중 한 가지만 쓰면 됩니다.

데이터베이스 관리자, DBA, Database Administrator

[문제 6]

- ① 릴레이션(Relation) ② 도메인(Domain) ③ AND ④ HAVING ⑤ SET

[풀이]

(1)

• SQL문

SELECT *	모든 속성을 표시한다.
FROM 사원	〈사원〉 테이블을 대상으로 검색한다.
WHERE 지사 = '서울' AND 직위 = '사원';	'지사'가 "서울"이고 '직위'가 "사원"인 튜플만을 대상으로 한다.

• SQL 실행 결과

이름	부서	직위	기본급	지사
이명진	개발	사원	230	서울
박한상	디자인	사원	250	서울

(2)

• SQL문

SELECT 부서, AVG(기본급)	'부서', '기본급'의 평균을 표시한다.
FROM 사원	〈사원〉 테이블을 대상으로 검색한다.
GROUP BY 부서	'부서'를 기준으로 그룹을 지정한다.
HAVING COUNT(*) >= 2;	그룹 중 튜플의 개수가 2개 이상인 부서만을 표시한다.

• SQL 실행 결과

부서	AVG(기본급)
인사	340
개발	365

(3)

• SQL문

UPDATE 사원	〈사원〉 테이블을 갱신한다.
SET 기본급 = 300	'기본급'을 300으로 갱신한다.
WHERE 직위 = '대리' AND 기본급 <= 300;	'직위'가 "대리"이고 '기본급'이 300이하인 튜플만을 대상으로 한다.

• SQL 실행 결과

이름	부서	직위	기본급	지사
권은경	인사	대리	300	서울

[문제 7]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① EAI, 전사적 응용 통합, Enterprise Application Integration
 ② CRM, 고객 관계 관리, Customer Relationship Management
 ③ CPM, 기업 성과 관리, Corporate Performance Management

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 8]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① PET, Privacy Enhancing Technology ② 허니팟, Honeypot ③ 딥페이크, Deepfake

[문제 9]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① GPU, Graphics Processor Unit, Graphics Processing Unit ② 디브스, Devops ③ 네트워크 슬라이싱, Network Slicing

[문제 10]

- ① Kernel ② Shell ③ Multi-Tasking ④ C ⑤ Linux

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

- ① 운영체제에서 가장 핵심적인 역할을 한다. 메모리나 프로세서와 같은 자원을 관리하거나, 시스템이 원활히 운영되도록 제어하는 역할을 수행한다. 일반적으로 종료된 입·출력 연산과 같이 서비스에 대한 경쟁적인 모든 요청들을 처리하는 인터럽트 처리기, 프로그램들의 처리 시간을 어떤 순서대로 나눌지 결정하는 스케줄러, 그리고 각각의 프로세스에게 권한을 부여하는 관리자를 포함한다.
- ② 시스템과 사용자 간의 인터페이스를 제공하는 명령줄 인터프리터이며, 커널과 함께 Unix의 주요 구성 요소 중 하나이다. DOS 운영체제에서 command.com 파일과 동일한 역할을 수행한다. 메모리에 적재되지 않고 외부 메모리에 파일 형태로 존재한다. 또한 명령 파이프라이닝을 지원하며, 입·출력 방향을 변경하는 것이 가능하다.
- ③ 하나 이상의 프로그램을 동시에 실행할 수 있는 기능이다. 예를 들어, MP3 음악 파일을 들으면서 인터넷에서 파일을 다운로드할 수 있다. 컴퓨터 하드웨어를 보다 효율적으로 사용할 수 있도록 하고, 개별 업무의 우선순위를 변경하여 중요한 작업이 덜 중요하게 생각되는 것보다 더 많은 프로세서 자원을 할당하도록 하는 조치를 포함하고 있다.
- ④ 전산에 있어 이것은 1969년에서 1973년 사이 AT&T 벨연구소에서 Dennis Ritchie가 최초 개발한 범용 프로그래밍 언어이다. ALGOL 계보의 다른 대부분의 명령형 언어와 마찬가지로 구조적 프로그래밍을 위한 기능을 가지고 있으며, 어휘의 가변적 범위와 재귀를 허용한다.
- ⑤ Linus B. Torvalds가 개발한 컴퓨터 운영체제이며, 다양한 하드웨어 플랫폼에서 실행되는 매우 인기있는 Unix 운영체제의 한 버전이다. 서버 OS로 널리 사용되며, 데스크톱 시장에서 우위를 점하고 있다. 또한 그것은 누구나 무료로 사용할 수 있는 오픈 소스 소프트웨어이다.





정답 및 해설

2019년 6월 정보처리산업기사 실기

[문제 1]

① N ② 5 ③ M ④ G ⑤ S

디버깅

배열 A에 대해 다음과 같이 값이 저장되었다고 가정하고 디버깅한 결과입니다.

• 배열 A

79	34	1	55	22	9	2	7	15	10
----	----	---	----	----	---	---	---	----	----

S	A(S)	A(S)>5	M	N>M	N	G	출력
0	79	Yes	74	Yes	999	79	7
1	34	Yes	29	Yes	74	34	
2	1	No	4	Yes	29	1	
3	55	Yes	50	No	4	2	
4	22	Yes	17	No	3	7	
5	9	Yes	4	No	2		
6	2	No	3	Yes			
7	7	Yes	2	Yes			
8	15	Yes	10	No			
9	10	Yes	5	No			
10							

[문제 2]

00001011

※ 답안 작성 시 주의 사항 : Java에서는 배열 선언 시 모든 요소가 0으로 초기화됩니다. 그러므로 a 배열 출력 시 값이 저장되지 않은 요소에는 0이 출력됩니다.

디버깅

d	n	d%2	a[n]	배열 a	출력																
11	0	1	1	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td></tr></table>	0	0	0	0	0	0	0	0	1	1	0	1					
0	0	0	0	0	0	0	0														
1	1	0	1																		
5	1	1	1																		
2	2	0	0																		
1	3	1	1																		
0	4																				

	1	9 3	3 9	9	0 1 2 3 4 5 6	<div> <div>4</div> <div>9</div> <div>3</div> <div>6</div> <div>8</div> <div>1</div> </div> <div>3 9</div>	943681 493681 439681
	2	9 6	6 9	9	0 1 2 3 4 5 6	<div> <div>4</div> <div>3</div> <div>9</div> <div>6</div> <div>8</div> <div>1</div> </div> <div>6 9</div>	943681 493681 439681 436981
	3	9 8	8 9	9	0 1 2 3 4 5 6	<div> <div>4</div> <div>3</div> <div>6</div> <div>9</div> <div>8</div> <div>1</div> </div> <div>8 9</div>	943681 493681 439681 436981 436891
	4	9 1	1 9	9	0 1 2 3 4 5 6	<div> <div>4</div> <div>3</div> <div>6</div> <div>8</div> <div>9</div> <div>1</div> </div> <div>1 9</div>	943681 493681 439681 436981 436891 436819
2	0	4 3	3 4	4	0 1 2 ⋮	<div> <div>4</div> <div>3</div> <div>6</div> <div>8</div> <div>1</div> <div>9</div> </div> <div>3 4</div>	943681 493681 439681 436981 436891 436819 346819
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

	5	0	3 1	1 3	3	0 1 2 :	<div> <div>314689</div> <div>13</div> </div>	943681 493681 439681 436981 436891 436819 346819 346819 346189 346189 346189 341689 341689 314689 134689
		1						
	6							

[문제 4]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 접수종류 ② 조인, JOIN ③ 자연, Natural ④ SELECT ⑤ π 인타넷(인터넷접수)

[문제 5]

- (1) AND (2) ASC (3) NOT IN

[풀이]

(1)

• SQL문

SELECT *	모든 속성을 표시한다.
FROM 학생	<학생> 테이블을 대상으로 검색한다.
WHERE 학번 LIKE '19%' AND 나이 = 21;	'학번'이 '19'로 시작하고 '나이'가 21인 튜플만을 대상으로 한다.

• SQL 실행 결과

학번	이름	나이
190098E	우길산	21

(2)

• SQL 문

SELECT 학번, 평균	'학번', '평균'을 표시한다.
FROM 성적	〈성적〉 테이블을 대상으로 검색한다.
ORDER BY 평균 ASC;	'평균'을 기준으로 오름차순 정렬한다.

• SQL 실행 결과

학번	평균
190098E	73
180098E	78
187620E	88
197720E	93

(3)

• SQL문

SELECT *	모든 속성을 검색한다.
FROM 학생	〈학생〉 테이블을 대상으로 검색한다.
WHERE 학번 NOT IN (SELECT 학번 FROM 성적);	〈학생〉 테이블에 '학번'이 NOT IN 다음에 쓰인 하위 질의의 결과에 없는 자료만을 대상으로 한다. '학번'을 검색한다. 〈성적〉 테이블에서 검색한다. 즉 〈성적〉 테이블의 '학번'을 검색한다.

• SQL 실행 결과

학번	이름	나이
160892B	성희연	23
177720B	김종수	24

[문제 6]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 무결성, Integrity ② SQL 주입 공격, SQL injection attack ③ CCL, 저작물 이용 약관, Creative Commons License

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 7]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① RSS, 맞춤형 정보 배달, Really Simple Syndication ② 비트코인, Bitcoin ③ 베이퍼웨어, 증발품, Vaporware
④ 브로드 데이터, broad data

[문제 8]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① SCM, Supply Chain Management, 공급망 관리, 공급 사슬 관리 ② RFID, Radio Frequency Identification, 전파 식별

※ 문제의 지문이 긴 관계로 일부만 복원이 되었습니다. 실제 시험에서는 총 4문항이 출제되었으며 세 번째와 네 번째 문항은 지문을 읽고 제시된 보기에서 하나를 골라 작성하는 문제였는데, 두 문항 모두 제시된 지문에서 쉽게 답을 찾을 수 있었습니다.

[문제 9]

① Deadlock ② Exclusion ③ Hold ④ Preemption ⑤ Circular waiting

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

(교착상태)는 두 개의 프로세스가 수행되기 전 서로가 끝나기를 기다리고 있을 때 발생하는 상황이다. 결국 두 프로세스 모두 동작이 멈춘다. 병행 환경 하에서 요청하는 프로세스들에게 시스템 자원을 무작정 허용함으로써 (교착상태)는 발생할 수 있다. (교착상태)가 발생하기 위해서는 다음의 조건이 충족되어야 한다.

상호 (배제) – 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 한다. 이를 달성하기 위해 적어도 한 개 이상의 자원이 비공유 상태로 유지되어야 한다.

(점유) 및 대기 – 각 프로세스들은 다른 자원들을 얻기 위해 대기하는 동안 이미 할당된 자원들을 계속 점유한다.

비(선점) – 프로세스에 할당된 자원은 오직 프로세스의 자발적인 행동에 의해서만 할당이 해제되며, 시스템이 강제로 해제할 수 없어야 한다.

(환형 대기) – 각 프로세스들은 체인의 다음 프로세스가 요청하는 하나 이상의 자원을 점유한다. 예를 들어, 체인이 P1, P2, P3, P4로 구성되어 있다면, P1은 P2를 기다리고 있고 P2는 P3을 기다리고 있고 P3는 P4를 기다리고 있고 P4는 P1을 기다리고 있다.





정답 및 해설

2019년 4월 정보처리산업기사 실기

[문제 1]

- ① SUM(T) ② T = T + 1 ③ K ④ SUM(H) ⑤ RANK = RANK + 1

디버깅

순서도에 대해 3건의 데이터를 각각 다음과 같은 순서대로 입력받았다고 가정하고 디버깅한 결과입니다.

NAME	TOE	GUN	LIC
White	89	45	55
Brown	67	80	39
Lilly	77	65	100

SUM	T	K	SUM(K)	H	SUM(H)	RANK	출력
189	1						<div> 1 White 189 2 2 Brown 186 3 3 Lilly 242 1 </div>
186	2						
242	3						
	4						
		1	189	1	189	1	
				2	186	2	
				3	242		
		2	186	1	189	1	
				2	186	2	
				3	242	3	
		3	242	1	189	1	
				2	186		
				3	242		

[문제 2]

15

디버깅

n	i	sum	출력
5	1	0	15
	2	1	
	3	3	
	4	6	
	5	10	
	6	15	

[문제 3]

$n * i$ 또는 $i * n$

디버깅

n	i	$n * i$	출력
6	1	6	$6 \times 1 = 6$
	2	12	$6 \times 2 = 12$
	3	18	$6 \times 3 = 18$
	4	24	$6 \times 4 = 24$
	5	30	$6 \times 5 = 30$
	6	36	$6 \times 6 = 36$
	7	42	$6 \times 7 = 42$
	8	48	$6 \times 8 = 48$
	9	54	$6 \times 9 = 54$
	10		

[문제 4]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 외부 스키마, 서브 스키마, External Schema, Sub Schema
- ② 개념 스키마, 논리 스키마, Conceptual Schema, Logical Schema
- ③ 내부 스키마, Internal Schema

[문제 5]

- ① 제2정규형(2NF) ② SET ③ 학과번호 ④ LIKE

[풀이]

• SQL문

UPDATE 학부생
SET 학과번호 = 999
WHERE 담당관 LIKE '이%';

'학부생' 테이블을 갱신한다.
'학과번호'를 999로 갱신한다.
'이름'이 "이"로 시작하는 튜플만을 대상으로 한다.

• SQL 실행 결과

학부	학과번호	입학생수	담당관
정경대학	110	300	김해울
공과대학	999	250	이성관
인문대학	120	400	김해울
정경대학	120	300	김성수
인문대학	999	180	이올해

[문제 6]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 무결성, Integrity ② XSS, 크로스 사이트 스크립팅, Cross Site Scripting ③ DDoS

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 7]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① EPG ② 펌토셀, 펌토셀 기지국, Femtocell, Femtocell Base Station ③ MVNO ④ IPTV, IP 텔레비전, Internet Protocol TeleVision
⑤ URC

[문제 8]

- ① EAI ② CRM ③ CPM ④ WBS ⑤ COCOMO

[문제 9]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① USB, Universal Serial Bus ② ISP, Internet Service Provider

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

- ① 개인용 컴퓨터와 주변기기들의 연결을 위해 프로토콜 규격을 정하는 산업표준들 중 하나다. 하나의 포트만으로 최대 127개의 장치를 연결하는 것이 가능하다. 초기에는 개인용 컴퓨터를 위해 개발되었으나, 오늘날에는 다양한 전자 제품에도 사용된다.
- ② 인터넷 접속 및 중계, 도메인 네임 등록, 웹 호스팅, 유즈넷 서비스, 콜로케이션 등 다양한 인터넷 서비스를 제공하는 사업자를 가리키는 용어다. 최초의 상업적인 서비스 제공자는 1989년 호주와 미국에서 설립된 'The World'다. 한국의 대표적인 제공업자로는 KT, SK 브로드밴드, LG 유플러스 등이 있다.





[문제 1]

1
3
3
5

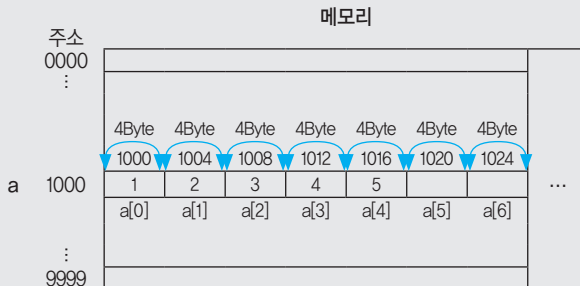
※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 결과를 한 줄로 1 3 3 5 혹은 1, 3, 3, 5로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 처리됩니다.

[해설]

모든 C 프로그램은 반드시 main() 함수부터 시작해야 합니다.

```
main() {
  ① int a[7] = { 1,2,3,4,5 };
  ② func(a);
    func(a + 2);
}
```

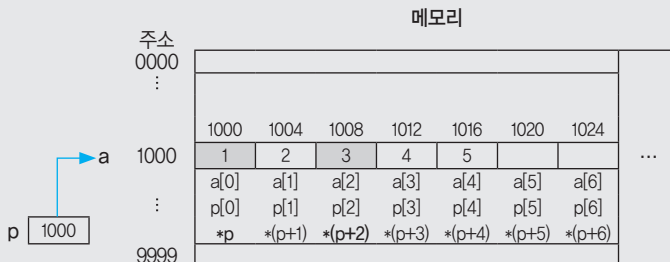
① 7개의 요소를 갖는 정수형 배열 a를 선언하고 주어진 값으로 초기화합니다. 정수형이므로 각 요소별로 4Byte씩 총 28Byte 공간이 메모리의 어딘가에 할당됩니다. 여기서 a 배열이 할당된 공간의 주소는 임의로 정한 것이며, 이해를 돕기 위해 주소를 10진수로 표현했습니다.



② a를 인수로 하여 함수 func()를 호출합니다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달됩니다. 그러니까 func(a)는 func(&a[0])와 같은 의미죠.

```
③ func(int *p) {
  ④ printf("%d\\n", *p);
  ⑤ printf("%d\\n", p[2]);
  ⑥ }
```

③ ②에서 'func(a)'라고 했으므로 정수형 포인터 변수 p는 a 배열의 시작 주소를 받습니다. 즉 정수형 포인터 변수 p와 a 배열의 주소가 동일하므로 a 배열의 요소를 p 배열의 요소로 표현할 수 있습니다.



④ 결과 1

실행문에서 *p는 p가 가리키는 곳에 저장된 값을 의미합니다. 즉 배열의 시작 주소인 p[0]에 저장된 값 1을 정수형으로 출력한 후 커서를 다음 줄 처음으로 옮깁니다.

⑤ 결과 1 3

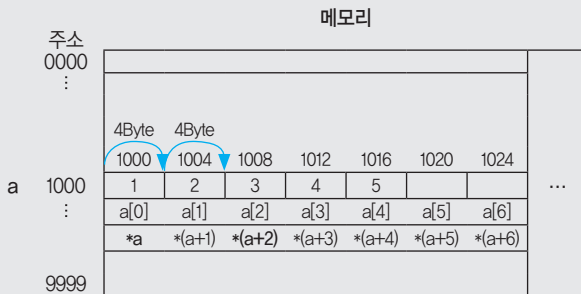
p[2]의 값인 3을 정수형으로 출력한 후 커서를 다음 줄 처음으로 옮깁니다.

p[2]는 *(p+2)와 같은 의미입니다. p+2는 p가 가지고 있는 주소 1000에서 2번지 증가는 1008이고 *(p+2)는 1008 번지가 가리키고 있는 값을 의미하니 결과 같은 거죠.

⑥ 함수를 마치고 func(a) 함수를 호출했던 main() 함수로 제어를 옮깁니다.

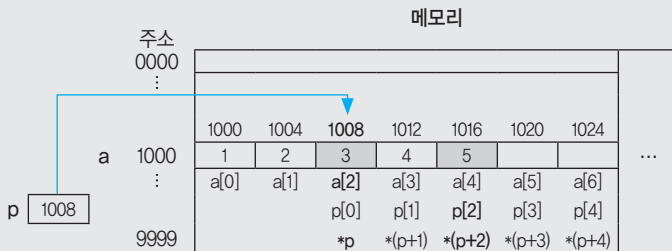
```
main() {
    int a[7] = { 1,2,3,4,5 };
    func(a);
    ⑦ func(a + 2);
}
```

⑦ a+2를 인수로 하여 함수 func()를 호출합니다. a는 배열의 이름으로 배열의 시작 주소를 의미합니다. a+2는 a에서 정수형으로 두 번지 증가한, 즉 8Byte 증가한 번지를 말하는 것이므로 &a[2]와 같습니다.



```
⑧ func(int *p) {
    ⑨ printf("%d\n", *p);
    ⑩ printf("%d\n", p[2]);
    ⑪ }
```

⑧ ⑦에서 'func(a+2)'라고 했으므로 정수형 포인터 변수 p는 a 배열에서 정수형으로 두 번지 증가한 a[2]의 주소(&a[2])를 받습니다.



⑨ 결과 1 3 3

p는 a[2]의 주소를 가지고 있고, *p는 p가 가리키는 곳의 값을 출력하는 것이므로 a[2]의 3을 정수형으로 출력한 후 커서를 다음 줄 처음으로 옮깁니다.

⑩ 결과 1 3 3 5

p[2]의 값인 5를 정수형으로 출력한 후 커서를 다음 줄 처음으로 옮깁니다.

- ① 함수를 마치고 func(a+2) 함수를 호출했던 main() 함수로 제어를 옮긴 후 프로그램을 종료합니다.

[문제 2]

- ① add ② result ③ pf ④ pf(i, j)

[해설]

모든 C 프로그램은 반드시 main() 함수부터 시작해야 합니다.

```
main() {  
  ① int i, j, result;  
  ② int (*pf)(int, int);  
  ③ scanf("%d, %d", &i, &j);  
  ④ pf = add;  
  ⑤ result = pf(i, j);  
  printf("%d, ", result);  
}
```

- ① 정수형 변수 i, j, result를 선언합니다.

- ② 정수형 매개변수 두 개가 있는 정수형 함수 포인터 pf를 선언합니다.

int (*pf)(int, int);

- ① ② ③

• ①: 반환값 자료형

• ②: 함수 포인터 이름, 함수 포인터 이름 앞에는 *를 붙인 뒤 ()로 묶어줍니다.

• ③: 매개변수 자료형

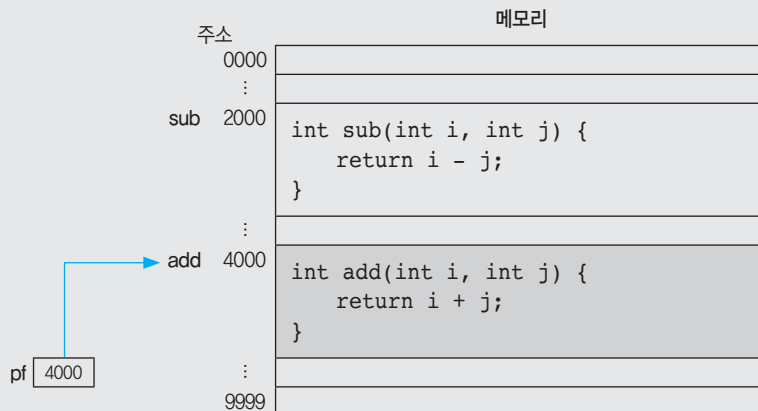
※ 함수 포인터: 함수의 시작 주소를 저장할 수 있는 포인터로, 함수 포인터를 주고받거나 함수를 호출할 수 있습니다.

- ③ 키보드로 숫자를 입력받아 값을 i와 j에 저장합니다. 문제에서 12와 6을 입력받았다고 가정했으므로 i에는 12가 j에는 6이 저장됩니다.

- ④ add 함수의 시작 주소를 함수 포인터 pf에 저장합니다. add 함수의 주소가 함수 포인터 pf에 저장된다는 것은 pf가 add 함수의 시작 주소를 가리키고 있다는 것을 의미합니다.

- ⑤ i와 j, 즉 12와 6을 인수로 하여 pf 함수를 호출한 다음 돌려받은 값을 result에 저장합니다.

pf에는 add 함수의 시작 주소가 저장되었으므로 pf 함수를 호출한다는 것은 add 함수를 호출하는 것과 같은 의미입니다. 여기서 함수들이 할당된 공간의 주소는 임의로 정한 것이며, 이해를 돕기 위해 10진수로 표현했습니다.



```
⑥ int add(int i, int j) {  
  ⑦ return i + j;  
}
```

⑥ ⑤번에서 'p(i, j)'라고 했으므로 정수형 변수 i는 i의 값 12를 받고 정수형 변수 j는 j의 값 6을 받습니다.

⑦ i + j, 즉 12 + 6의 결과인 18을 가지고 add 함수를 호출했던 main() 함수로 제어를 옮깁니다.

```
main() {
    int i, j, result;
    int (*pf)(int, int);
    scanf("%d, %d", &i, &j);
    pf = add;
    result = pf(i, j);
    ⑧ printf("%d, ", result);
    ⑨ pf = sub;
    ⑩ result = pf(i, j);
    printf("%d", result);
}
```

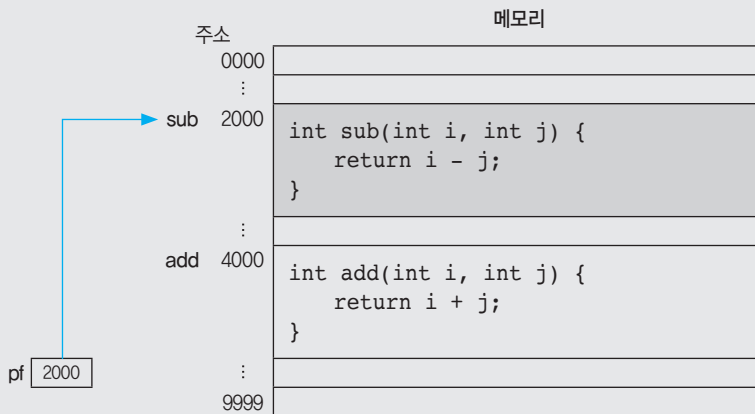
⑧ 결과 18,

result에 저장된 값인 18을 정수형으로 출력하고 이어서 심표와 공백 한 칸(" ")을 출력합니다.

⑨ sub 함수의 메모리 주소를 함수 포인터 pf에 저장합니다.

⑩ i와 j, 즉 12와 6을 인수로 하여 pf 함수를 호출한 다음 돌려받은 값을 result에 저장합니다.

pf에는 sub 함수의 메모리 주소가 저장되었으므로 pf 함수를 호출한다는 것은 sub 함수를 호출한다는 의미입니다.



```
11 int sub(int i, int j) {
12     return i - j;
}
```

⑪ ⑩번에서 'p(i, j)'라고 했으므로 정수형 변수 i는 i의 값 12를 받고 정수형 변수 j는 j의 값 6을 받습니다.

⑫ i - j, 즉 12 - 6의 결과인 6을 가지고 add 함수를 호출했던 main() 함수로 제어를 옮깁니다.

```
main() {
    int i, j, result;
    int (*pf)(int, int);
    scanf("%d, %d", &i, &j);
    pf = add;
    result = pf(i, j);
    printf("%d, ", result);
    pf = sub;
    result = pf(i, j);
    13 printf("%d", result);
}
```

㉓ 결과 18, 6

result에 저장된 값인 6을 정수형으로 출력한 후 프로그램을 종료합니다.

[문제 3]

- ① $N + 1$ ② $77 - GK(N)$ ③ $GK(N) - 77$ ④ COMP ⑤ $GK(N)$

디버깅

배열 GK에 다음과 같이 값이 저장되었다고 가정하고 디버깅한 결과입니다.

GK(10)	12	45	78	94	65	32	15	8	7	51
--------	----	----	----	----	----	----	----	---	---	----

N	GK(N)	$GK(N) > 77$	CHA	$COMP > CHA$	COMP	FGK	출력
0	12	No	65	Yes	300	12	78
1	45	No	32	Yes	65	45	
2	78	Yes	1	Yes	32	78	
3	94	Yes	17	No	1		
4	65	No	12	No			
5	32	No	45	No			
6	15	No	62	No			
7	8	No	69	No			
8	7	No	70	No			
9	51	No	26	No			
10							

[문제 4]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① 조작 ② 데이터베이스 관리자, DBA, Database Administrator ③ 후보키, Candidate Key ④ 참조

[문제 5]

- (1) 이름 (2) WHERE (3) AND (4) DISTINCT (5) ORDER BY

※ 답안 작성 시 주의 사항 : 대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

[풀이]

(1)

- SQL 실행 결과

이름
장길산
성준향
홍길동
일지매
김선달

• SQL 문

```
SELECT 이름      이름을 표시한다.
FROM 직원;      <직원> 테이블을 대상으로 검색한다.
```

(2)

• SQL 실행 결과

직원코드	이름	부서
161353	장길산	인사
151453	홍길동	인사

• SQL 문

```
SELECT *      모든 속성을 표시한다.
FROM 직원      <직원> 테이블을 대상으로 검색한다.
WHERE 부서 = '인사'; '부서' 속성의 값이 '인사'인 자료만을 대상으로 한다.
```

(3)

• SQL 실행 결과

신청코드	이름	분야	활동점수
D-1012	장길산	등산	3000

• SQL 문

```
SELECT *      모든 속성을 표시한다.
FROM 동호회      <동호회> 테이블을 대상으로 검색한다.
WHERE 분야 = '등산' AND 활동점수 >= 3000; '분야' 속성의 값이 '등산'이고 '활동점수' 속성의 값이 3000 이상인 자료만을 대상으로 한다.
```

(4)

• SQL 실행 결과

분야
골프
등산
바이크
배드민턴
수영

• SQL 문

```
SELECT DISTINCT 분야      '분야' 속성을 표시하되, 같은 '분야'는 한 번만 표시한다.
FROM 동호회;              <동호회> 테이블을 대상으로 검색한다.
```

(5)

• SQL 실행 결과

이름	분야
성춘향	바이크
홍길동	등산
장길산	배드민턴
장길산	등산
김선달	골프
김선달	수영

• SQL 문

SELECT 이름, 분야	'이름', '분야' 속성을 표시한다.
FROM 동호회	<동호회> 테이블을 대상으로 검색한다.
ORDER BY 활동점수;	'활동점수' 속성의 값을 기준으로 오름차순 정렬한다.

[문제 6]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 지식 관리 시스템, KMS, Knowledge Management System ② 지식 경영 최고 책임자, CKO, Chief Knowledge Officer
 ③ 전자 문서 관리 시스템, EDMS, Electronic Document Management System ④ 지식 공동체, COP, Communities of Practice
 ⑤ SWOT

※ **답안 작성 시 주의 사항** : 한글 또는 영문으로 쓰라는 지시사항이 없을 경우 한글 정답이나 영문 정답 모두 맞는 것으로 처리됩니다. 그리고 한글이나 영문 두 가지를 쓸 경우 두 개 다 맞아야만 정답으로 인정되므로 영문이나 한글 한 가지만 쓰는 것이 좋습니다. 영문 스펠링이나 한글 맞춤법이 틀렸을 경우 완전히 틀린 것으로 간주된다는 것도 잊지 마세요.

[문제 7]

- ① Internet ② LAN ③ Topology ④ Protocol ⑤ Node

※ **답안 작성 시 주의 사항** : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

컴퓨터 및 통신 기술의 태동기에는 이 두 기술이 전혀 다른 주제로 취급되었다. 하지만 이러한 기술들이 발전함에 따라 두 가지가 병합되기 시작했다. (Internet)라고 하는 신기술의 혁신이 등장하고 인간의 삶 전체를 변화시켰다.

(Internet)는 컴퓨터와 네트워크가 결합된 데이터 통신 시스템이다. 물리적으로는, (Internet)는 통신 회선으로 연결된 여러 컴퓨터 또는 터미널로 구성된다. (Internet)은 사용자가 다양한 네트워크를 통해 정보에 대한 접근을 공유할 수 있도록 한다. 일부 네트워크는 전 세계에 걸쳐 있고 다른 네트워크는 비교적 작은 지역을 담당한다.

(Internet)에는 다음을 포함한 여러 종류가 있다.

- (LAN) : 컴퓨터들이 지리적으로 가까운 거리에 있다(예를 들어 같은 건물).
- WAN : 컴퓨터들이 멀리 떨어져 있으며 전화 회선이나 무선 전파로 연결된다.
- CAN(Compus Area Network) : 컴퓨터들이 캠퍼스 또는 군사 기지 등과 같은 제한된 지리적 영역 내에 있다.
- MAN(Metropolitan Area Network) : 소도시나 대도시를 위해 설계된 데이터 네트워크이다.
- HAN(Home Area Network) : 사용자의 디지털 기기들을 연결하는 사용자 가정 내의 네트워크이다.

이러한 종류 이외에도 여러 종류의 네트워크로 분류하는 데 다음과 같은 특성이 사용된다.

- (Topology) : 컴퓨터 시스템의 기하학적 배치
일반적인 (Topology)에는 버스, 스타 및 링이 포함된다.
- (Protocol) : 네트워크의 컴퓨터들이 통신을 하기 위해 사용하는 공통된 규칙 및 신호 집합을 정의한다. LAN에 가장 널리 사용되는 (Protocol) 중 하나를 이더넷이라고 한다. PC에서 널리 사용되는 또 다른 LAN (Protocol)은(는) IBM 토큰링 네트워크이다.
- 아키텍처 : 네트워크는 넓은 의미로 피어 투 피어 또는 클라이언트/서버 아키텍처로 분류할 수 있다.

네트워크상의 컴퓨터를 (Node)라고도 한다. 네트워크에서는 처리 위치이다. 일반적으로 (Node)는 컴퓨터일 수도 있고 프린터와 같은 다른 장치일 수 있다.

모든 (Node)에는 DLC(Data Link Control) 주소 또는 MAC(Media Access Control) 주소라고도 하는 고유한 네트워크 주소가 있다.

네트워크에 자원을 할당하는 컴퓨터 및 장치들을 서버라고 한다.

[문제 8]

- ① 다음 중 하나를 쓰면 됩니다.

디지털 포렌식, Digital forensic

- ② OTP ③ CCL

※ 답안 작성 시 주의 사항 : ②번과 ③번 문항의 경우 3글자 약어로 작성하라는 조건이 제시되었으므로, 한글 또는 영문 Full-name으로 답안을 작성할 경우 오답이 될 수 있습니다.

[문제 9]

- ① RSS ② HVC

- ③ 다음 중 하나를 쓰면 됩니다.

노모포비아, Nomophobia

- ④ 다음 중 하나를 쓰면 됩니다.

타조, Tajo

※ 답안 작성 시 주의 사항 : ①번과 ②번 문항의 경우 3글자 약어로 작성하라는 조건이 제시되었으므로, 한글 또는 영문 Full-name으로 답안을 작성할 경우 오답이 될 수 있습니다.





[문제 1]

① i ② j ③ i*j

※ 답안 작성 시 주의 사항 : C언어 코드 형식으로 쓰라는 조건이 있으므로 C언어 코드 형식에 맞게 답안을 작성해야 합니다. 예를 들어, C언어의 변수명은 대소문자를 구분하므로 i나 j를 I나 J와 같이 대문자로 작성하면 오답으로 처리됩니다.

디버깅

i	j	i*j	출력
2	1	2	2 * 1 = 2
	2	4	2 * 2 = 4
	3	6	2 * 3 = 6
	4	8	2 * 4 = 8
	5	10	2 * 5 = 10
	6	12	2 * 6 = 12
	7	14	2 * 7 = 14
	8	16	2 * 8 = 16
	9	18	2 * 9 = 18
	10		
⋮	⋮	⋮	⋮
9	1	9	9 * 1 = 9
	2	18	9 * 2 = 18
	3	27	9 * 3 = 27
	4	36	9 * 4 = 36
	5	45	9 * 5 = 45
	6	54	9 * 6 = 54
	7	63	9 * 7 = 63
	8	72	9 * 8 = 72
	9	81	9 * 9 = 81
	10		
10			

[문제 2]

n

디버깅						
p	n	t	m	i	r	출력
2	3	1.73205...	1	2	0	97
5	4	2.00000...	2	2	1	
7	5	2.23606...	2	2	0	
11	6	2.44949...	2	2	1	
13	7	2.64575...	2	2	0	
:	8	2.82842...	2	2	1	
97	9	3.00000...	3	2	0	
	10	3.16227...	3	3	0	
	11	3.31662...	3	2	1	
	12	3.46410...	3	2	2	
	13	3.60555...	3	3	0	
	:	:	:	2	1	
	97	9.84885...	9	2	1	
	98	9.89949...	9	3	:	
	99	9.94987...	9	:	1	
	100	10.00000...	10	2	1	
	101			3	1	
				4	2	
				5	1	
				6	6	
				7	1	
				8	7	
				9	0	
				2	1	
				2	0	
				3	0	
				2		

[문제 3]

- ① A(i) ② B(J) ③ B(M) ④ X ⑤ A(M)

디버깅

배열 A와 배열 B에 각각 다음과 같이 저장되었다고 가정하고 디버깅한 결과입니다.

- 배열 A

2

4

6

8

10
- 배열 B

1

3

5

7

9

[문제 4]

① CHECK ② IN

[틀이]

CREATE TABLE 사원

〈사원〉 테이블을 생성한다.

(직원코드 NUMBER NOT NULL,

직원코드 속성은 숫자형으로, NULL을 갖지 않는다.

성명 CHAR(10) UNIQUE,

성명 속성은 최대 문자 10자로, 중복된 값을 가질 수 없다.

직책 CHAR(10) CHECK (직책 IN('사원', '대리', '과장', '팀장')), 직책 속성은 최대 문자 10자로, '사원', '대리', '과장', '팀장' 중에 하나의 값을 가져야 한다.

연봉 NUMBER);

연봉 속성은 숫자형이다.

[문제 5]

각 문항별로 다음 중 하나를 쓰면 됩니다.

① SQL, Structured Query Language ② DDL, Data Description Language ③ Relation ④ DML, Data Manipulation Language

⑤ DCL, Data Control Language

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

원래 (SQL)은 SEQUEL(구조화된 영문 질의 언어)이라고 불렸으며, IBM 연구소에서 실험적인 관계형 데이터베이스 시스템인 SYSTEM R을 위한 인터페이스로 설계 및 구현되었다.

(DDL)은 데이터베이스의 여러 구조를 정의하는 명령의 표준이다. (DDL) 문은 테이블, 색인 그리고 사용자와 같은 데이터베이스 객체들을 생성, 수정 그리고 제거한다. 일반적인 (DDL) 문은 CREATE, ALTER, 그리고 DROP이다. (Relation)은 같은 구조를 갖는 튜플의 집합이다. (SQL)에서 (Relation)은 거의 테이블과 같은 의미로 사용된다. (Relation) 변수는 (DDL)에 의해 정의될 수 있다.

(DML)은 애플리케이션이나 쿼리를 통해 저장된 데이터를 처리하기 위해 데이터베이스 사용자가 사용하는 언어이다. 데이터베이스 사용자와 데이터베이스 관리 시스템의 사이에서 인터페이스를 제공한다. 또한 (DML)은 SELECT, INSERT, DELETE 그리고 UPDATE의 네 가지 종류가 있다.

(DCL)은 데이터의 보안, 무결성, 회복 그리고 병렬 수행 제어를 정의하기 위해 사용하는 언어이다. 데이터베이스 관리자는 데이터 관리 목적으로 (DCL)을 사용한다. 또한 이것은 COMMIT, ROLLBACK, GRANT, 그리고 REVOKE의 네 가지 종류가 있다.

[문제 6]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 무결성, integrity ② 데이터 암호화 표준, DES, Data Encryption Standard ③ 크로스 사이트 스크립팅, XSS, Cross Site Scripting
④ 해시, Hash ⑤ IPsec, Internet Protocol Security, IP 보안 프로토콜

※ 답안 작성 시 주의 사항 : 한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 7]

- ① 주소 ② 취미 ③ 경력 ④ 사원번호

[문제 8]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 에지, EDGE, Enhanced Data Rates For Global Evolution ② 비트코인, bitcoin
③ 고화질 멀티미디어 인터페이스, HDMI, High-Definition Multimedia Interface ④ 인터넷 전화, VoIP, Voice over Internet Protocol
⑤ 미라캐스트, Miracast

[문제 9]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 스키마, schema ② 인스턴스, instance ③ 차수, degree ④ 카디널리티, cardinality ⑤ 원자

[문제 10]

- ① 수주 ② 출하지시 ③ 입고 ④ 입하 ⑤ 물류센터





[문제 1]

① 7 ② n

※ 답안 작성 시 주의 사항 : C언어에서 사용하는 변수명은 대소문자를 구분하기 때문에 변수명을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

디버깅

n	i	j	출력
7	1	1 2	1
	2	1 2 3	1 1 2
	3	1 2 3 4	1 1 2 1 2 3
	4	1 2 3 4 5	1 1 2 1 2 3 1 2 3 4
	⋮	⋮	⋮
	7	1 2 3 4 5 6 7 8	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 5 6 1 2 3 4 5 6 7
	8		

[문제 2]

1 2 4 7

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 공백 없이 1247로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

main 함수		arr 함수						출력								
i	n[]	a[]	n	i	j	sw	temp									
	<table><tr><td>4</td><td>2</td><td>7</td><td>1</td></tr></table>	4	2	7	1	<table><tr><td>4</td><td>2</td><td>7</td><td>1</td></tr></table>	4	2	7	1	4					
	4	2	7	1												
	4	2	7	1												
		<table><tr><td>4</td><td>2</td><td>7</td><td>1</td></tr><tr><td>1</td><td></td><td></td><td>4</td></tr></table>	4	2	7	1	1			4		0	1 2 3 4	0 1 3	4	
	4	2	7	1												
1			4													
	<table><tr><td>1</td><td>2</td><td>7</td><td>4</td></tr></table>	1	2	7	4		1	2 3 4	1	2						
1	2	7	4													
	<table><tr><td>1</td><td>2</td><td>7</td><td>4</td></tr><tr><td></td><td></td><td>4</td><td>7</td></tr></table>	1	2	7	4			4	7		2	3 4	2 3	7		
1	2	7	4													
		4	7													
				3												
0 1 2 3 4	<table><tr><td>1</td><td>2</td><td>4</td><td>7</td></tr></table>	1	2	4	7							1 1 2 1 2 4 1 2 4 7				
1	2	4	7													

[문제 3]

- ① S = I ② I ③ F = I ④ K + 1 ⑤ A(J, I) = K

디버깅

N	K	SW	S	F	I	J	출력																																																	
7	0	4																																																						
	1 2 3 4 5 6 7		1	7	1	1 2 3 4 5 6 7 8	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2							3							4							5							6							7						
1																																																								
2																																																								
3																																																								
4																																																								
5																																																								
6																																																								
7																																																								
	8 9 10 11 12		2	6	2	2 3 4 5 6 7	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9						4	10						5	11						6	12						7						
1																																																								
2	8																																																							
3	9																																																							
4	10																																																							
5	11																																																							
6	12																																																							
7																																																								
	13 14 15		3	5	3	3 4 5 6	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td></td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13					4	10	14					5	11	15					6	12						7						
1																																																								
2	8																																																							
3	9	13																																																						
4	10	14																																																						
5	11	15																																																						
6	12																																																							
7																																																								
	16		4	4	4	4 5	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td></td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13					4	10	14	16				5	11	15					6	12						7						
1																																																								
2	8																																																							
3	9	13																																																						
4	10	14	16																																																					
5	11	15																																																						
6	12																																																							
7																																																								
	17 18 19		3	5	5	3 4 5 6	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td></td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td></td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td></td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8						3	9	13		17			4	10	14	16	18			5	11	15		19			6	12						7						
1																																																								
2	8																																																							
3	9	13		17																																																				
4	10	14	16	18																																																				
5	11	15		19																																																				
6	12																																																							
7																																																								
	20 21 22 23 24		2	6	6	2 3 4 5 6 7	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td>20</td><td></td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td>21</td><td></td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td>22</td><td></td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td>23</td><td></td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td>24</td><td></td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	1							2	8				20		3	9	13		17	21		4	10	14	16	18	22		5	11	15		19	23		6	12				24		7						
1																																																								
2	8				20																																																			
3	9	13		17	21																																																			
4	10	14	16	18	22																																																			
5	11	15		19	23																																																			
6	12				24																																																			
7																																																								
	25 26 27 28 29 30 31		1	7	7	1 2 3 4 5 6 7 8	<table><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td>25</td></tr><tr><td>2</td><td>8</td><td></td><td></td><td></td><td>20</td><td>26</td></tr><tr><td>3</td><td>9</td><td>13</td><td></td><td>17</td><td>21</td><td>27</td></tr><tr><td>4</td><td>10</td><td>14</td><td>16</td><td>18</td><td>22</td><td>28</td></tr><tr><td>5</td><td>11</td><td>15</td><td></td><td>19</td><td>23</td><td>29</td></tr><tr><td>6</td><td>12</td><td></td><td></td><td></td><td>24</td><td>30</td></tr><tr><td>7</td><td></td><td></td><td></td><td></td><td></td><td>31</td></tr></table>	1						25	2	8				20	26	3	9	13		17	21	27	4	10	14	16	18	22	28	5	11	15		19	23	29	6	12				24	30	7						31
1						25																																																		
2	8				20	26																																																		
3	9	13		17	21	27																																																		
4	10	14	16	18	22	28																																																		
5	11	15		19	23	29																																																		
6	12				24	30																																																		
7						31																																																		
					8																																																			

[문제 4]

- ① select 학생이름, 생년월일 from 학생 where 전공 = 'computer' and 학년=1;
- ② select distinct 전공 from 학생 where 학년 = 2 or 학년 = 4;
또는 select distinct 전공 from 학생 where 학년 in(2, 4);
- ③ select 학생이름 from 학생 where 학년 >= 2 order by 학년 desc;

[문제 5]

- ① Array ② Index ③ Tree ④ Stack ⑤ Queue

※ 답안 작성 시 주의 사항 : 영문으로 쓰라는 지시사항이 있을 경우 스펠링에 유의하세요. 스펠링을 한 개라도 잘못 쓰면 오답으로 처리되니까요.

[원문해석]

- ① (배열)은 인덱스(혹은 첨자)에 의해 구별되는 동일하게 입력된 데이터 항목의 집합이다. 변수와 다르게 같은 이름을 가지고 있어 같은 형식의 데이터를 저장하고 수집하기에 편리하다.
- ② (인덱스)는 데이터를 보거나 기록할 때 데이터 레코드에 빠르게 접근하기 위한 키와 포인터로 구성된 데이터 구조이다. 테이블의 한 열 또는 여러 열을 사용하기 위해 생성될 수 있다. 빠른 검색과 레코드 접근과 관련된 효율적인 시퀀싱 작업들을 위한 기반을 제공한다.
- ③ (트리)는 정보를 지정하는 노드가 가장 눈에 띄는 구조이다. 부모 노드, 자식 노드, 자손 노드로 구성되며, 분기와 동일한 구조를 가진다. 분기의 끝에 있는 노드, 즉 가장 어린 노드를 리프라고 한다.
- ④ (스택)은 항목이 추가된 순서와 반대로 제거되는 특수한 형태의 데이터 구조이다. 그래서 가장 최근에 추가된 항목이 첫 번째로 제거된 항목이다. 이것은 선입, 선출(LIFO)이라고 불린다.
- ⑤ (큐)는 수집에 있어 개체들이 순서대로 유지되고, 수집에 관한 주요(혹은 유일한) 작업은 단말의 전면 위치에서 개체들의 추가와 단말의 후면에서 개체들의 제거인 수집의 특별한 유형이다. 이는 선입선출(FIFO) 데이터 구조와 관련이 있다.

[문제 6]

각 문항별로 제시된 답안 중 한 가지만 쓰면 됩니다.

- ① DES, Data Encryption Standard, 데이터 암호 표준 ② RPO, Recovery Point Objective, 목표 복구 시점
- ③ OTP, One-Time Password, 일회용 패스워드 ④ DDoS, Distributed Denial of Service, 분산 서비스 거부 공격 ⑤ 백도어, Back Door

[문제 7]

- ① Cartesian Product, 카티션 프로덕트, 교차곱 ② select ③ theta, 세타, θ ④ ▶◀ ⑤ natural, 자연, 내추럴

[문제 8]

- ① NAS, Network Attached Storage, 망 부착 저장 장치 ② SON, Self Organizing Network, 자동 구성 네트워크 ③ Memristor, 멤리스터
- ④ VoIP, Voice over Internet Protocol, 음성 인터넷 프로토콜, 인터넷 전화 ⑤ AllJoyn, 올조인

[문제 9]

- ① MBO ② 입찰 ③ 낙찰 ④ 계약

※ ②~④번 문항은 모두 지문 안에서 답을 찾을 수 있는 문제였습니다.