2021학년도 중등학교교사 임용후보자 선정경쟁시험

정보 · 컴퓨터

수험 번호 : (　　)　　**성　명 : (　　)**

제1차 시험	2교시 전공A	12문항 40점	시험 시간 90분

○ 문제지 전체 면수가 맞는지 확인하시오.
○ 모든 문항에는 배점이 표시되어 있습니다.

1. 괄호 안의 ㉠에 해당하는 단어를 쓰시오. [2점]

> 잊힐 권리(the right to be forgotten)는 인터넷 사용자가 소셜 네트워크 서비스나 포털 게시판 등에 올린 게시물의 (㉠)을/를 요청할 수 있는 권리이다. 인터넷에 올라간 사적인 정보는 개인의 것이지만, 그 정보를 (㉠)할 수 있는 권한은 해당 기관에 있기 때문에 문제가 발생한다. 인터넷에 공개된 정보로 인해 개인의 권리가 침해되는 경우, 정보통신망 이용촉진 및 정보보호 등에 관한 법률(법률 제17358호, 2020.6.9., 일부개정) 제44조의 2에 근거하여 정보의 (㉠)을/를 해당 기관에 요청할 수 있다. 잊힐 권리와 관련하여 셰리 터클(Sherry Turkle)이 제시한 "인터넷은 절대 잊지 않는다."라는 금언을 마음에 담고 있어야 한다는 주장도 있다.

2. 다음은 어떤 학생이 IEEE 802.11 CSMA/CA 프로토콜의 송신측 동작을 순서대로 나열한 것인데, 밑줄 친 ㉠~㉣의 순서가 잘못되었다. <조건>을 고려하여 밑줄 친 ㉠~㉣을 프로토콜 동작에 맞게 순서대로 나열하시오. [2점]

> 채널이 사용되지 않음을 감지하였다.
> DIFS 시간 동안 기다린다.
> ㉠ RTS를 전송한다.
> ㉡ 데이터 프레임을 전송한다.
> ㉢ SIFS 시간 동안 기다린다.
> ㉣ CTS 수신을 기다린다.
> ACK 수신을 기다린다.

<조 건>

○ 송신측의 동작만 기술한 것이다.
○ 데이터 전송 오류나 충돌은 발생하지 않는다고 가정한다.
○ RTS : Request to Send
○ DIFS : Distributed InterFrame Space
○ SIFS : Short InterFrame Space
○ CTS : Clear to Send

3. 다음은 정보과 교육과정에 대한 교사들 간의 대화이다. 2015 개정 교육과정(교육부 고시 제2020-225호)에 근거하여 괄호 안의 ㉠, ㉡에 해당하는 내용을 순서대로 쓰시오. [2점]

> A 교사 : 선생님, 선생님들은 중 · 고등학교 시절에 정보 수업에 참여한 경험이 있나요?
> B 교사 : 저는 제7차 교육과정이 운영되던 시기에 중 · 고등학교를 다녔어요. 그때는 '컴퓨터' 과목이 중학교에 있었어요. 고등학교에서는 '정보사회와 컴퓨터'를 배웠고요. 2007 개정 교육과정에서부터 중학교와 고등학교 보통 교과의 컴퓨터 관련 과목 명칭이 '정보'로 통일되면서 과목 목표나 내용 영역이 동일하게 설정되었어요.
> C 교사 : 맞아요. 2015 개정 교육과정에서부터는 중학교 '정보' 과목이 (㉠) 교과(군)에 편성되었고, 중학교 3년간 34시간 이상 이수해야 하는 필수 과목이 되었어요. 고등학교 '정보' 과목은 보통 교과의 일반 선택 과목으로 변경되었고, 중 · 고등학교 '정보' 과목과 연계해서 교육할 수 있는 다양한 전문 교과 과목들도 있어요. 과학계열 고등학교에는 전문 교과 I 에 '정보과학' 과목이 편성되어 있고, 특성화 고등학교와 산업수요 맞춤형 고등학교에는 (㉡)의 17개 교과(군) 등에 다양한 관련 과목들이 기초 과목과 실무 과목으로 편성되어 있어요.

4. (가)는 어떤 운영체제 S가 제공하는 일부 시스템 호출의 기능 설명이고, (나)는 운영체제 S상에서 수행되는 프로세스 간에 발생 가능한 임계영역 문제의 해결책으로서 C 언어의 함수로 정의한 상호배제 프리미티브(primitive)이다. <조건>을 고려하여 (나)의 ㉠과 ㉡에 들어갈 코드를 순서대로 쓰시오. [2점]

(가)

`int create(char *pathname);`
파일경로명이 `pathname`인 파일을 생성하고 성공했을 경우 0을 리턴(return)한다. 여러 프로세스가 동시 혹은 미세한 시간 차를 두고 시스템 호출 `create()`를 호출하더라도 그중 하나의 프로세스만이 해당 파일 생성에 성공한다. 파일경로명이 `pathname`인 파일이 이미 존재할 경우 -1을 리턴한다.

`void remove(char *pathname);`
파일경로명이 `pathname`인 파일을 삭제한다. 해당 파일이 없을 경우 아무 일도 수행하지 않고 리턴한다.

(나)

```
void enterCS(char *pathname) {
    while (create(pathname) == [ ㉠ ])
        ;
}

void exitCS(char *pathname) {
    [ ㉡ ];
}
```

<조 건>

○ (나)의 `enterCS()`와 `exitCS()`는 (가)에 제시된 파일 생성/삭제 기능의 두 시스템 호출을 사용해 정의된다.

○ (나)의 `enterCS()`와 `exitCS()`는 다음 예와 같이 임계영역 진입/진출 위치에 동일 인자(파일경로명)를 설정해 호출하는 형태로 서로 짝 지어 사용될 경우, 임계영역 문제 해결에 필요한 한계대기(bounded waiting) 조건을 충족하지는 못하지만 상호배제(mutual exclusion)와 진행(progress) 조건은 충족한다.

(예)

```
/* 임계영역 이전 코드 */
enterCS("file1");
/* 임계영역 */
exitCS("file1");
/* 임계영역 이후 코드 */
```

5. 다음은 퀵 정렬을 하는 C 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

```
int partition(int a[], int n, int p, int q) {
    int v = a[p], i, j, t;
    i = p;
    j = q + 1;
    do {
        do
            i++;
        while (a[i] < v);
        do
            j--;
        while (a[j] > v);
        if (i < j) {
            t = a[i];
            a[i] = a[j];
            a[j] = t;
        }
    } while (i < j);
    a[p] = a[j];
    a[j] = v;
    return j;
}

void quicksort(int a[], int n, int p, int q) {
    int j;
    if (p < q) {
        j = partition(a, n, p, q);
      ㉠ print(a, n);
      ㉡ printf("%d ", j);
        quicksort(a, n, p, j - 1);
        quicksort(a, n, j + 1, q);
    }
}
```

<조 건>

○ `partition(a, n, p, q)` 함수는 `a[q+1]`의 값이 `a[p]`에서 `a[q]`까지의 값보다 크다는 것을 가정한다.

○ `print(a, n)`은 배열 `a[0]`부터 `a[n-1]`까지의 값을 순서대로 출력하는 함수이다.

<작성 방법>

○ 배열 a의 값이 {43,87,15,32,29,76,65,1000}인 상태에서 `quicksort(a, 7, 0, 6)`을 호출하였을 때, 밑줄 친 ㉠이 1번째 수행될 때와 2번째 수행될 때의 출력 결과를 순서대로 쓸 것.

○ 배열 a의 값이 {43,87,15,32,29,76,65,1000}인 상태에서 `quicksort(a, 7, 0, 6)`을 호출하였을 때, 밑줄 친 ㉡의 수행 출력 결과를 순서대로 쓸 것.

6. 다음 C 프로그램과 <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

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

int a[] = {2, 4, 5, 1, 6, 3, 0};

int *rCount(int *a, int low, int high, int d) {
    int i;
    int *p = malloc(d * sizeof(int));
    for (i = 0; i < d; i++)
        p[i] = 0;
    for (i = low; i <= high; i++)
        p[a[i] % d]++;
    return p;
}

int linkCount(int s, int t) {
    int nlinks = 0;
    while (s != t) {
        s = a[s];
        nlinks++;
    }
    return nlinks;
}

int f(int x) {
    static int id = 2;
    id = a[id];
    return id + x;
}

int g(int x) {
    static int val = 3;
    return val += x;
}

int main(void) {
    int b[] = {12, 19, 15, 26, 29, 21, 14, 32, 17, 22, 23};
    int *p = rCount(b, 2, 8, 3);

  ㉠ printf("%d ", p[2]);
  ㉡ printf("%d ", linkCount(5, 6));
    g(5);
  ㉢ printf("%d ", g(f(4)));
    return 0;
}
```

<조 건>

- ○ malloc(size)는 size 개의 바이트를 저장·활용할 수 있는 저장 공간을 할당하고 그 시작 주소를 리턴해 주는 라이브러리 함수이다.
- ○ malloc() 함수 호출은 항상 성공한다고 가정한다.

<작성 방법>

- ○ 밑줄 친 ㉠이 수행될 때 출력되는 값을 쓰고, 함수 rCount()가 리턴하는 주소가 가리키는 저장 공간에 어떤 의미의 값들이 저장되어 있는지 서술할 것.
- ○ 밑줄 친 ㉡이 수행될 때 출력되는 값을 쓸 것.
- ○ 밑줄 친 ㉢이 수행될 때 출력되는 값을 쓸 것.

7. (가)는 출판사, 도서, 서점 간의 도서 공급 업무에 대한 요구 사항을 분석한 것이다. (가)와 <조건>을 고려하여 E-R 다이어그램을 작성한 후, 릴레이션 스키마를 생성하려고 한다. <작성 방법>에 따라 서술하시오. [4점]

(가)

[요구 사항 분석]

1. 도서는 도서번호, 도서이름, 단가를 가지고, 도서번호는 유일한 성질을 갖는다.
2. 서점은 서점번호, 서점이름을 가지고, 서점번호는 유일한 성질을 갖는다.
3. 출판사는 출판사번호, 출판사이름, 주소를 가지고, 주소는 출판사의 위치 정보를 구하기 위해 시, 구, 동별로 세분하여 나타낸다. 출판사번호는 유일한 성질을 갖는다.
4. 한 출판사는 여러 종류의 도서를 출판할 수 있다.
5. 한 종류의 도서는 하나의 출판사에서 출판된다.
6. 한 종류의 도서는 여러 권씩 여러 곳의 서점에 공급될 수 있다.
7. 서점은 여러 종류의 도서를 공급받을 수 있다.

<조 건>

○ 릴레이션 스키마를 표현할 때 기본키는 밑줄로, 외래키는 위 첨자 *로 표시한다.
(예) 등록(학번*, 과목번호*, 성적, 중간성적, 기말성적)
○ 개체(Entity) 이름과 속성 이름은 (가)를 참조하고, '도서' 개체와 '서점' 개체 사이의 관계(Relationship) 이름은 '공급', '도서' 개체와 '출판사' 개체 사이의 관계 이름은 '출판'이라 한다.
○ '출판사' 개체의 주소 속성은 복합 속성이다.
○ '공급' 관계는 '수량'이라는 속성을 가진다.
○ 개체 사이의 관계는 2진 관계 타입만 고려한다.

<작성 방법>

○ '공급' 관계에 대한 릴레이션 스키마를 쓰되, 기본키와 외래키를 표시할 것.
○ 작성된 E-R 다이어그램을 최소 개수의 릴레이션 스키마를 갖도록 변환하였을 때, '출판사', '도서', '서점' 개체의 릴레이션 스키마를 쓰되, 기본키와 외래키를 표시할 것.

8. (가)는 가상의 16비트 컴퓨터 M에서 현재 메모리 상태이고, (나)는 메모리에 저장된 프로그램 코드의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

주소	데이터
… (상략) …	… (상략) …
1000	2000
1002	2002
1004	2004
1006	2006
1008	2008
… (하략) …	… (하략) …

(나)

```
   LOAD, 0, 1000
   INCR
㉠ STORE, 1, 1004
```

<조 건>

○ 컴퓨터 M에서 메모리 주소의 길이는 16비트이다.
○ 명령어는 Opcode(연산코드)와 AM(주소지정 방법), Addr(주소)로 되어 있다.
○ Opcode의 길이는 5비트, AM의 길이는 1비트, Addr의 길이는 10비트로 고정되어 있다.
○ 레지스터는 AC(Accumulator)만 사용한다.
○ AM=0이면 직접 주소지정(direct addressing), AM=1이면 간접 주소지정(indirect addressing) 방식에 의해 Addr을 유효 주소(effective address)로 계산한다.
○ 직접 주소지정의 경우 Addr이 곧 유효 주소이다.
○ 간접 주소지정의 경우 Addr이 가리키는 곳에 저장되어 있는 값(16비트)이 유효 주소이다.
○ Opcode의 종류에는 LOAD, STORE, INCR 등이 있고 각각에 대한 설명은 다음과 같다.
  - LOAD, AM, Addr : Addr의 유효 주소에 저장되어 있는 값(16비트)을 AC에 넣는다.
  - STORE, AM, Addr : AC의 값(16비트)을 Addr의 유효 주소에 저장한다.
  - INCR : AC의 값을 1 증가시킨다.
○ 문제에 제시된 주소나 데이터는 모두 10진수이다.

<작성 방법>

○ (가)의 메모리 상태에서 (나)의 코드를 실행할 때, 밑줄 친 ㉠ 명령어에서 Addr의 유효 주소와 저장 값(AC 값)을 순서대로 쓸 것.
○ 컴퓨터 M에서 직접 주소지정 방식에 비해 간접 주소지정 방식이 갖는 장점과 단점을 서술할 것.

9. (가)는 시점 t에 어떤 운영체제 X에 의해 수행되는 프로세스와 관리되는 자원의 상태이고, (나)는 시점 t 이후 해당 프로세스들이 운영체제 X에 자원 할당이나 반환을 요청한 순서이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

	R_1	R_2	R_3	R_4	R_5
P_1	2	3	3	1	2
P_2	4	1	1	2	1
P_3	2	1	1	4	1
P_4	1	2	1	1	0
P_5	1	0	1	5	3

요구 행렬 C

	R_1	R_2	R_3	R_4	R_5
P_1	0	2	1	1	1
P_2	3	1	1	1	0
P_3	1	1	0	1	0
P_4	1	0	1	0	0
P_5	0	0	1	1	1

할당 행렬 A

R_1	R_2	R_3	R_4	R_5
2	0	0	2	1

가용 벡터 V

(나)

① P_1에 의한 할당 요청 벡터 (1, 0, 0, 0, 1)의 자원 할당 요청
② P_3에 의한 할당 요청 벡터 (0, 0, 0, 2, 0)의 자원 할당 요청
③ P_5에 의한 반환 요청 벡터 (0, 0, 1, 1, 0)의 자원 반환 요청
④ P_4에 의한 할당 요청 벡터 (0, 1, 0, 1, 0)의 자원 할당 요청

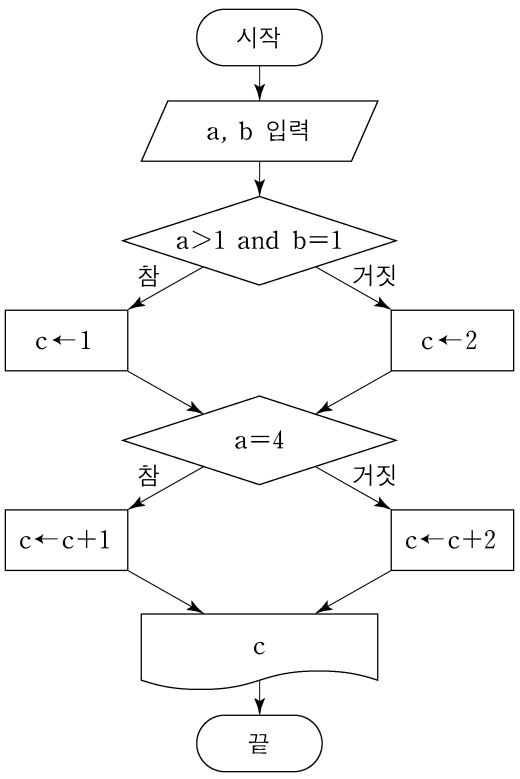
<조 건>

○ 운영체제 X는 교착상태를 피하기 위해 시스템 자원 관리에 은행원 알고리즘을 적용해 시스템의 상태를 항상 안전 상태로 유지한다.
○ 시점 t에 시스템 내에 수행되고 있는 프로세스는 P_1, P_2, P_3, P_4, P_5밖에 없다.
○ 은행원 알고리즘이 관리하는 자원에는 R_1, R_2, R_3, R_4, R_5의 5가지밖에 없다.
○ 요구 행렬 C의 i행 j열 원소 C_{ij}는 P_i가 수행 도중 할당받아 점유할 수 있는 자원 R_j의 최대 개수이다.
○ 할당 행렬 A의 i행 j열 원소 A_{ij}는 P_i가 운영체제 X로부터 할당받아 해당 시점에 점유하고 있는 자원 R_j의 개수이다.
○ 가용 벡터 $V = (V_1, V_2, V_3, V_4, V_5)$의 i 번째 원소 V_i는 운영체제 X가 해당 시점에 임의의 프로세스에 할당해 줄 수 있는 가용 자원 R_i의 개수이다.
○ 운영체제 X를 대상으로 한 할당 요청 벡터 $(a_1, a_2, a_3, a_4, a_5)$는 5가지 자원 R_i 각각에 대해 a_i 개의 추가 자원을 일괄 할당해 달라는 특정 프로세스의 요청이다.
○ 운영체제 X를 대상으로 한 반환 요청 벡터 $(r_1, r_2, r_3, r_4, r_5)$는 5가지 자원 R_i 각각에 대해 이미 할당받아 점유하고 있는 자원 중 r_i 개의 자원을 일괄 반환하겠다는 특정 프로세스의 요청이다.
○ 자원 할당 요청이 거부될 경우 해당 프로세스는 블록 상태로 대기하게 되며, 이후 다른 프로세스가 반환한 자원에 의해 할당 요청이 수용될 수 있는 상황에 이르는 즉시 해당 자원을 할당받고 준비 상태로 전이된다. 이때 일괄 할당 가능한 프로세스가 다수일 경우 할당 요청 시점이 빠른 요청부터 할당한다.
○ 각 할당 요청에 대한 '실제 할당' 시점은 요청된 자원 모두가 그것을 요청한 프로세스에 실제로 할당되는 시점을 말한다. 시점 t_a에 할당 요청된 자원 모두가 바로 할당된 경우 시점 t_a가 '실제 할당' 시점이 되고, 시점 t_b에 할당 거부된 자원 요청이 이후 시점 t_c에 반환된 자원으로 가용 자원에 여유가 생겨 운영체제 X에 의해 수용·할당된다면 시점 t_b의 자원 요청에 대한 '실제 할당' 시점은 시점 t_c가 된다.
○ 시점 t 이후 ④의 요청이 있기까지 생성되거나 종료된 프로세스가 없고, (나)에 제시된 요청 이외에 자원 할당이나 반환과 관련된 다른 요청은 없다.

<작성 방법>

○ (가)의 시스템 상태는 안전 순서가 존재하는 안전 상태이다. 해당 안전 순서의 2번째 프로세스와 3번째 프로세스를 순서대로 쓸 것.
○ (나)의 할당 요청 중 '실제 할당' 시점이 가장 빠른 요청의 번호를 쓰고, 해당 할당 시점 직후 가용 벡터 V의 값을 쓸 것.

10. 다음은 프로그램 검사를 위한 테스트 케이스를 만들기 위해 제시된 순서도와 테스트 케이스 표이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]



순서도에 대한 테스트 케이스 표

케이스	입력		결과
	a	b	c
1	4	2	3
2	㉠	㉡	3
3	㉢	1	2
4	3	㉣	4

<조 건>

○ 표의 케이스 1, 2는 제시된 순서도에 대해 분기 테스팅에 따른 테스트 데이터이고, 표의 케이스 1, 2, 3, 4는 경로 테스팅에 따른 테스트 데이터이다.

○ 분기 테스팅(branch testing)은 각 분기점에서 참인 경우와 거짓인 경우 모두가 검사될 수 있도록 테스트 데이터를 만드는 기준이다. 위 순서도의 경우 최소 2가지의 a, b 값 조합으로 테스트 데이터를 만들 수 있다.

○ 경로 테스팅(path testing)은 각 분기점의 참·거짓 값에 따라 나타날 수 있는 모든 실행 경로가 검사될 수 있도록 테스트 데이터를 만드는 기준이다. 위 순서도의 경우 2개의 분기가 있으므로 모든 경로의 수는 $2 \times 2 = 4$가 된다.

<작성 방법>

○ 분기 테스팅에 따른 테스트 데이터의 ㉠에 해당하는 조건과 ㉡에 해당하는 값을 쓸 것.

○ 경로 테스팅에 따른 테스트 데이터의 ㉢에 해당하는 값과 ㉣에 해당하는 조건을 쓸 것.

11. 다음은 가네(R. Gagné)의 9단계 수업 단계에 따라 설계한 교수·학습 지도안이다. <작성 방법>에 따라 서술하시오. [4점]

대단원 (중단원)	Ⅳ. 컴퓨팅 시스템 (1. 컴퓨팅 시스템의 동작 원리)	소단원	1. 컴퓨터 하드웨어
학습 대상	중학교 1학년	학습 장소	컴퓨터실
교과 역량	협력적 문제해결력	교수·학습 방법	강의법
학습 목표	컴퓨터를 구성하는 하드웨어의 특징을 설명할 수 있다.		

수업 단계		교수·학습 활동
도입	㉠ 주의 집중	○ 학생의 호기심을 유발하기 위해 '컴퓨팅 시스템의 구성'에 대한 흥미로운 동영상 자료를 시청하게 한다.
	수업 목표 제시	○ "컴퓨터를 구성하는 하드웨어의 특징을 설명할 수 있다."를 학습 목표로 제시한다.
	사전 지식 회상	○ 컴퓨터의 기계적 장치에 대한 지식을 상기시킨다.
전개	학습 자극 자료 제시	○ 컴퓨터를 구성하는 하드웨어 장치를 보여주며 명칭을 알려 준다.
	학습 안내 제공	○ 각 장치의 기능을 찾아보도록 안내한다. ○ 각 장치를 기능에 따라 나누어 보도록 안내한다.
	학습 수행 유도	○ 각 장치의 명칭과 기능을 정리할 수 있도록 시간을 제공한다. ○ 교사가 제공한 활동지를 작성하도록 한다.
	피드백 제공	○ 작성한 활동지에 대한 정답 여부와 부연 설명을 제공한다.
정리	성취 행동 평가	○ 학습 성취도를 확인하기 위해 학생들에게 컴퓨터를 구성하는 하드웨어에 대한 퀴즈를 풀게 한다.
	㉡	○ ㉢ 컴퓨터를 구성하는 하드웨어의 특징을 다시 알려 준다. ○ ㉣ 배운 내용을 토대로 온라인 쇼핑몰에서 광고하는 컴퓨터 하드웨어 사양을 보고 컴퓨팅 시스템의 성능을 확인해 보는 활동을 수행하게 한다.

<작성 방법>

○ 켈러(J. Keller)의 학습 동기 이론 4가지 요소(ARCS) 중 밑줄 친 ㉠의 하위 요소 3가지를 쓸 것.

○ ㉡ 단계의 요소 중 밑줄 친 ㉢과 ㉣에 해당하는 요소의 명칭과 개념을 서술할 것(단, ㉢, ㉣의 순서로 제시할 것).

한국교육과정평가원

12. 다음은 리스트를 이용하여 허프만 트리를 구하는 C 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

```
#define N 5
typedef struct node *NODE;
struct node {
    int w;        /* 가중치 */
    NODE lc, rc; /* 트리의 자식 링크 */
    NODE next;   /* 트리를 구성 요소로 하는
                    리스트의 연결 링크 */
};
NODE list = NULL; /* 리스트 */

/* 리스트에서 가중치가 가장 작은 원소(트리)를 가져오는 함수 */
NODE least() {
    NODE t;
    if (list == NULL)
        return NULL;
    t = list;
    list = list->next;
    return t;
}

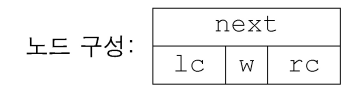
/* 리스트에 원소(트리)를 가중치 순서에 맞게 삽입하는 함수 */
void insert(NODE t) {
    NODE pt, old;
    if (list == NULL || t->w < list->w) {
        t->next = list;
        list = t;
    } else {
        old = list;
        pt = old->next;
        while (pt != NULL && pt->w < t->w) {
            old = pt;
            pt = pt->next;
        }
        [ ㉠ ];
        [ ㉡ ];
    }
}

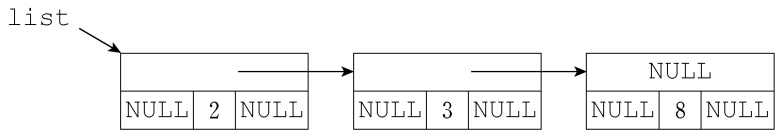
int main(void) {
    int i;
    NODE pt;
    init();   /* 리스트를 초기화 */
    for (i = 1; i < N; i++) {
        pt = newnode();
        pt->lc = least();
        pt->rc = least();
        pt->w = (pt->lc)->w + (pt->rc)->w;
        insert(pt);
    }
    return 0;
}
```

<조 건>

○ init() 함수는 초기 가중치를 받아 오름차순으로 정렬된 리스트를 만드는 함수이다. 예를 들면 초기 가중치가 3개인 자료 (3, 8, 2)의 경우 다음과 같은 리스트로 만든다.

노드 구성:





○ newnode() 함수는 노드를 생성하는 함수이다.

○ least() 함수는 리스트 list에 들어 있는 원소(트리) 가운데 가중치 값이 가장 작은 것을 가져오는 함수이다.

○ insert() 함수는 리스트 list에 새 원소(트리)를 정렬 순서에 맞게 삽입하는 함수이다.

<작성 방법>

○ 초기 가중치 값을 (15, 5, 8, 12, 4)로 하여 프로그램을 수행한 결과, 구성된 최종 이진트리를 전위(preorder) 순회할 때 1번째 방문 노드와 3번째 방문 노드의 가중치 값을 순서대로 쓸 것.

○ insert() 함수가 올바르게 동작하도록 ㉠, ㉡에 들어갈 코드를 순서대로 쓸 것.

<수고하셨습니다.>