2020학년도 중등학교교사 임용후보자 선정경쟁시험

정보 · 컴퓨터

수험 번호 : (　　)　　**성　명 : (　　)**

제1차 시험	3교시 전공B	11문항 40점	시험 시간 90분

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

1. 다음은 문자열을 처리하는 C 프로그램이다. 실행 결과를 쓰시오. [2점]

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

int main () {
   char data[3][20] = {"education",
                       "computer",
                       "algorithm"};
   int k, count;
   char *p;
   count = strlen (data[0]);
   p = (char *) data;
   printf ("%s", &data[1][0]);
   printf ("\n");
   for (k = 0; k < count; k++)
      printf ("%c", *(p + count - (k + 1)));
   return 0;
}
```

2. (가)는 '마트' 데이터베이스의 '제품' 테이블이다. (나)는 마트에 제품을 납품하는 회사별 제품의 평균가격이 300을 초과하는 회사의 회사코드와 평균가격을 검색하는 SQL문이다. <조건>을 고려하여 ㉠에 해당하는 내용을 쓰고, (가)테이블에 대해 (나)SQL문을 실행하였을 때 결과 튜플(tuple)의 개수를 순서대로 쓰시오. [2점]

(가)

제품 (기본키 : 제품코드)

제품코드	이름	가격	회사코드
1001	생수	100	10
1002	파이	400	20
1003	사탕	100	10
1004	음료수	300	10
1005	과자	200	30
1006	연필	200	30
1007	치약	500	20
1008	가방	900	10

(나)

```
select 회사코드, avg(가격) as 평균가격
from 제품
group by 회사코드
having [ ㉠ ] ;
```

<조 건>

○ '마트'는 관계형 데이터베이스이다.
○ '제품' 테이블의 각 속성은 순서대로 제품 코드, 제품 이름, 제품 가격, 그 제품을 납품하는 회사 코드를 의미한다.

3. (가)는 A 교사가 고등학교 정보 과목 수업의 토론 활동에서 활용한 읽기 자료이고, (나)는 2015 개정 고등학교 정보 과목 교육과정(교육부 고시 제2015-74호)의 '성취기준' 중 일부를 순서에 상관없이 제시한 것이다. (다)는 성취기준 [12정보01-05]에 대한 수업 후, 학생이 작성한 조사 보고서의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

(㉠) 기술은 인간의 지능이 가지는 학습, 추론 등의 기능을 컴퓨터가 가질 수 있도록 하는 기술로, 최근 주목받고 있는 기계학습(machine learning)도 이 기술의 한 분야로 발전하고 있다. (㉠) 기술을 핵심으로 활용하는 분야에는 전문가 시스템(expert system), 데이터 마이닝(data mining), 패턴인식(pattern recognition), 자연어 처리(natural language processing), 컴퓨터 비전(computer vision), 음성 인식(speech recognition), 로봇 공학(robotics) 등이 있다.

(나)

① 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.
② 정보사회에서 정보과학의 지식과 기술이 활용되는 분야를 탐색하고 영향력을 평가한다.
③ 동일한 정보가 다양한 방법으로 디지털로 변환되어 표현될 수 있음을 이해하고 정보 활용 목적에 따라 보다 효율적인 방법을 선택한다.
④ 운영체제의 개념과 기능을 이해하고 운영체제를 활용하여 컴퓨팅 시스템의 자원을 효율적으로 관리한다.
⑤ 유무선 네트워크의 특성을 이해하고 사용하는 컴퓨팅 시스템의 네트워크 환경을 설정한다.

(다)

교육과정 성취기준	[12정보01-05] (㉡) 보호 제도 및 방법을 알고 올바르게 활용한다.
학습목표	(㉡) 보호 제도 및 방법을 알고 이를 올바르게 실천할 수 있다.
○○ 조사 보고서	
… (중략) … 4차 산업혁명시대에는 소프트웨어의 역할이 점점 확대될 것이고, 나아가 그 가치를 보호하기 위한 제도가 강화 될 것이다. … (중략) … 조사한 주요 결과는 다음과 같다. 1. 소프트웨어의 종류 소프트웨어에는 ㉢ 상용 소프트웨어, 공개 소프트웨어, 오픈 소스 소프트웨어 등이 있다. … (하략) …	

<작성 방법>

○ (가)의 ㉠에 해당하는 용어를 쓰고, (가)를 읽기 자료로 활용한 A 교사의 수업과 가장 관련 깊은 성취기준의 번호를 (나)에서 찾아 쓸 것.
○ (다)의 밑줄 친 ㉢의 의미를 ㉡의 관점에 근거하여 서술할 것.

4. (가)는 2015 개정 중학교 정보과 교육과정(교육부 고시 제2015-74호)을 참고하여 A 교사가 성취 평가를 위해 교육과정 성취기준에 대한 평가기준을 계획한 것이다. (나)는 이에 관해 학생이 제출한 활동지의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

교육과정 성취기준	평가기준	
[9정03-01] 실생활 문제 상황에서 문제의 현재 상태, 목표 상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석한다.	상	실생활 문제 상황을 분석하여 문제의 현재 상태와 목표 상태 모두를 정확히 정의할 수 있다.
	중	실생활 문제 상황에서 문제의 현재 상태와 목표 상태 중 하나만 정확히 정의할 수 있다.
	하	실생활 문제 상황이 무엇인지 설명할 수 있다.

(나)

활 동 지

○○ 중학교
이름 : ○○○

[**문제 상황**] 집에서 출발해서 도서관에 들러 책을 반납하고 학교에 도착하는 최단 경로는?

… (중략) …

Q1. ㉠ 제시된 문제 상황과 지도를 보고 현재 상태와 목표 상태를 정의해 봅시다.

- 현재 상태 : 집에서 도서관을 들러 학교에 도착하는 최단 경로를 모르는 상태
- 목표 상태 : 집에서 학교에 도착하는 최단 경로를 아는 상태

… (중략) …

Q5. ㉡ Q1부터 Q4까지의 활동을 얼마나 잘 했는지 스스로 '평가지 1'에 평가해 봅시다.

Q6. 각자의 활동 결과를 발표한 뒤에 ㉢ 친구의 활동에 대해 '평가지 2'에 평가해 봅시다.

… (하략) …

<작성 방법>

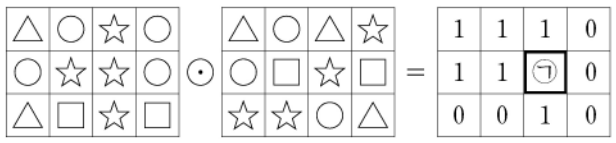
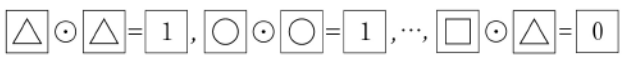
○ 성취기준 중심 학생 평가의 관점에서 A 교사가 준비한 (가)에 나타난 문제점 1가지를 구체적으로 서술할 것.
○ ㉠에 대해 학생이 작성한 답은 (가)의 평가기준에서 상, 중, 하 어디에 해당하는지 쓸 것.
○ (나)의 Q5와 Q6을 분석하여 2015 개정 중학교 정보과 교육과정(교육부 고시 제2015-74호)의 '교수·학습 및 평가의 방향'에 제시된 ㉡, ㉢의 평가방법을 서술할 것.

5. (가)는 A 교사가 중학교 정보과 수업을 위해 준비한 가상 로봇의 '기능'과 '연산 조건'에 대한 설명이고, (나)는 A 교사가 수업 아이디어를 구상한 내용이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

기능	• 입력 : ○, □, △, ☆ 기호로만 구성된 입력 카드를 2장 읽음 • 출력 : '연산 조건'에 따라 연산하여 결과 카드에 출력함
연산 조건	• 연산(⊙)은 2장의 카드를 입력 받아 동일한 위치의 셀끼리 연산함 예) 입력 카드 1의 (1,1) 셀의 값 ⊙ 입력 카드 2의 (1,1) 셀의 값 = 결과 카드 (1,1) 셀에 이진수로 출력 • 연산 결과는 입력 카드와 동일한 크기의 결과 카드에 이진수로 출력됨 • ○, □, △는 같은 기호끼리 연산되면 1, 아니면 0이 출력됨 • ☆은 다른 기호와 연산될 때 ☆의 규칙이 우선 적용되며 다음과 같은 규칙을 따름 … (하략) …

(나)

학습 주제	• 로봇이 실행되는 규칙을 이해하고 알고리즘 설계하기
활동 아이디어	
문제 이해	• 로봇이 다음과 같이 처리할 때, 로봇이 처리하는 규칙 찾아보기 (입력 카드 1) (입력 카드 2) (결과 카드) 
문제 분해	• 작은 문제로 분해하기 
규칙 찾기	• 불필요한 요소를 제거하여 핵심 요소를 추출하기 … (중략) … • ㉡가상 로봇의 '연산 조건'에 포함된 모든 규칙 찾기 - ○, □, △는 같은 기호끼리 연산되면 1, 아니면 0이 출력됨 - ☆는 다른 기호와 연산될 때 ☆의 규칙이 우선 적용되며 다음과 같은 규칙을 따름 1) 첫 번째 입력 기호 ☆이 두 번째 입력 기호 ☆을 만나면 0을 출력 2) (㉢) 3) (㉣)
알고리즘 설계	… (하략) …

<작성 방법>

○ 2015 개정 중학교 정보과 교육과정(교육부 고시 제2015-74호)의 '문제 해결과 프로그래밍' 영역 중 (나)를 분석하여 제시한 '활동 아이디어'에서 고려하지 <u>않은</u> '핵심 개념'이 무엇인지 서술할 것.

○ ㉡을 완성하기 위하여 필요한 ☆에 대한 규칙 ㉢, ㉣을 서술할 것.

○ (나)의 결과 카드에 있는 ㉠에 출력되는 값을 쓸 것.

6. 전송 데이터의 오류 검출을 위해 순환 중복 검사(CRC : Cyclic Redundancy Code)를 이용하는 프레임이 있다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

<조 건>

○ 송신단과 수신단에서 사용하는 CRC 생성 다항식은 $x^3 + x^2 + 1$ 이다.

○ 송신단은 데이터 필드(field)와 FCS(Frame Check Sequence) 필드로 구성된 8 비트 길이의 프레임을 수신단으로 전송한다.

○ 프레임을 비트열로 표시하면, 데이터 필드는 왼쪽에 위치하고 FCS 필드는 오른쪽에 위치한다.

○ 프레임을 비트열로 표시하면, 왼쪽 첫 번째 비트가 MSB(Most Significant Bit)이다.

○ FCS 필드의 길이는 최소로 한다.

○ 수신단에서는 수신 프레임을 생성 다항식으로 나누어 나머지가 0이면 전송 오류가 검출되지 않는 것으로 판단한다.

<작성 방법>

○ 오류가 검출되지 않은 경우, 수신 프레임에 포함된 데이터 필드의 길이를 쓸 것.

○ 비트열 10101111로 표시되는 프레임 A와 비트열 10100111로 표시되는 프레임 B가 수신되었다. 이때, 각 프레임의 오류 검출 여부를 쓰고, 오류가 검출되지 않은 프레임에 대해서는 FCS를 제외한 데이터 필드를 쓸 것.

7. 다음은 수식 트리(expression tree)를 이용하여 산술 연산식을 계산하는 C 프로그램이다. 프로그램 코드와 <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

```
#include <stdio.h>
typedef struct tree_node {
   int data;
   struct tree_node *left, *right;
} TreeNode;

TreeNode n1 = {6, NULL, NULL};
TreeNode n2 = {3, NULL, NULL};
TreeNode n3 = {'/', &n1, &n2};
TreeNode n4 = {4, NULL, NULL};
TreeNode n5 = {'-', [ ㉠ ], [ ㉡ ] };
TreeNode n6 = {2, NULL, NULL};
TreeNode n7 = {5, NULL, NULL};
TreeNode n8 = {'*', &n6, &n7};
TreeNode n9 = {'+', &n5, &n8};

int Expression_tree (TreeNode *node) {
   int op1, op2, op3;
   if (node == NULL)
      return 0;
   if (node->left == NULL && node->right == NULL)
      return node->data;
   else {
      op1 = Expression_tree (node->left);
      op2 = Expression_tree (node->right);
      if (node->data == (int) '/')
         op3 = op1 / op2;
      if (node->data == (int) '*')
         op3 = op1 * op2;
      if (node->data == (int) '+')
         op3 = op1 + op2;
      if (node->data == (int) '-')
         op3 = op1 - op2;
      return op3;
   }
   return 0;
}

int main () {
   printf ("%d", Expression_tree (&n9));
   return 0;
}
```

<조 건>

- ○ 주어진 수식 트리는 이진트리(binary tree)이다.
- ○ 수식 트리에서 피연산자는 단말(leaf) 노드에 저장되고, 산술 연산자는 루트(root) 노드 또는 내부(internal) 노드에 저장된다.
- ○ TreeNode의 data는 피연산자 또는 산술 연산자 {'+', '-', '*', '/'}를 갖는다.
- ○ 주어진 수식 트리를 루트 노드 n9부터 전위 순회한 결과는 다음과 같다.

+, -, /, 6, 3, 4, *, 2, 5

<작성 방법>

- ○ 주어진 수식 트리를 루트 노드 n9부터 후위 순회한 결과를 쓸 것.
- ○ ㉠, ㉡에 해당하는 내용을 순서대로 쓸 것.
- ○ 프로그램의 실행 결과를 쓸 것.

한국교육과정평가원

8. (가)는 어떤 컴퓨터 시스템에서 프로세스와 관련된 현재 메모리 상태이고 (나)는 앞으로의 페이지 참조 순서이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

○ 프로세스 Pa와 Pb만 현재 실행 중이다.
○ a1, a2, a3, a4, a5는 Pa의 페이지이다.
○ b1, b2, b3, b4, b5는 Pb의 페이지이다.
○ 현재 시스템의 물리 메모리 상태는 다음 표와 같다. (단, 참조 시점이 클수록 최근 참조된 페이지를 의미한다.)

프레임번호	적재된 페이지	참조시점
1	a2	1
2	a3	2
3	b2	3
4	b3	4
5	b5	5

(나)

○ 프로세스 Pa와 Pb가 다음과 같은 순서로 페이지를 참조한다.

참조시점 / 프로세스	6	7	8	9	10	11	12	13
Pa	a5		a2	a3	a5			a2
Pb		b2				b3	b2	

<조 건>

○ 반입 정책은 요구 페이징(demand paging)을 사용한다.
○ 교체 정책은 LRU(Least Recently Used)를 사용한다.
○ 적재 집합 관리는 전역 교체이다. 즉, 교체 대상 페이지를 선택할 때 프로세스를 구분하지 않는다.
○ 프로세스의 페이지 적재를 위한 시스템의 물리 메모리 프레임 수는 5이다.
○ 1개의 프레임에는 1개의 페이지만 적재 가능하다.
○ 프로세스가 참조하려는 페이지가 물리 메모리에 없으면 페이지 폴트(page fault)가 발생하고 이때 페이지 교체가 이루어진다.
○ 페이지를 물리 메모리 프레임에 적재하는 데 소요되는 시간은 0이라고 가정한다.

<작성 방법>

○ 참조시점 6에서 Pa가 페이지 a5를 참조할 때 페이지 교체가 이루어진다. 이때 a5와 교체되는 페이지를 쓰고, 그 이유를 서술할 것.
○ Pa와 Pb가 (가)의 상태에서 (나)와 같은 순서로 페이지 참조를 하는 동안 발생한 Pa의 페이지 폴트 횟수와 Pb의 페이지 폴트 횟수를 순서대로 쓸 것. (단, <조건>에서 설명하지 않은 페이지 폴트는 고려하지 않음.)

9. 다음은 트랜잭션 T_1, T_2, T_3의 연산이고, 각각은 독립적인 트랜잭션이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

T_1	T_2	T_3
$r_1(x)$ $w_1(x, 30)$	$r_2(x)$ $w_2(y, 60)$ $r_2(z)$ $w_2(z, 90)$	$r_3(x)$ $r_3(y)$ $w_3(x, 40)$

<조 건>

○ 데이터 x, y, z의 초깃값은 각각 20, 50, 80이다.
○ $r_i(a)$는 트랜잭션 T_i가 데이터 a를 읽는 읽기 연산이며, $w_i(a, 10)$은 트랜잭션 T_i가 데이터 a에 10을 기록하는 쓰기 연산이다. c_i는 트랜잭션 T_i가 완료하였음을 의미한다.
○ 로그에서 <T_i start>와 <T_i commit>은 트랜잭션 T_i의 시작과 완료를 각각 의미하며, <checkpoint>는 검사점 기록 로그이다. <T_i, a, 10, 20>은 '트랜잭션 T_i가 데이터 a의 현재 값 10을 20으로 쓰기 연산 하였음'을 의미하는 로그이다.
○ 회복(recovery) 가능한 스케줄이란 어떤 트랜잭션 T가 읽은 데이터 x에 대해, 읽기 연산 이전에 데이터 x에 쓰기 연산을 수행한 모든 트랜잭션들이 완료되기 전까지는 T가 완료되지 않은 스케줄을 의미한다.

<작성 방법>

○ 트랜잭션 T_1과 T_2가 스케줄 s의 순서로 실행되었을 때, 스케줄 s의 회복 가능 여부를 쓰고, 그 이유를 서술할 것. (단, 스케줄 s 이전에 실행한 트랜잭션들은 모두 종료되었다고 가정함.)

s : $r_1(x)$ $r_2(x)$ $w_2(y, 60)$ $r_2(z)$ $w_1(x, 30)$ $w_2(z, 90)$ c_2 c_1

○ 트랜잭션 T_1, T_2, T_3이 병행적으로 실행되는 도중 시스템 실패가 발생하였으며, 그 동안 다음과 같은 로그가 기록되었다. 지연 갱신 기법에서 이 로그를 사용하여 회복 작업을 진행할 때, undo와 redo가 필요한 트랜잭션을 각각 쓸 것. (단, 필요한 트랜잭션이 없을 경우 '없음'이라고 쓸 것.)

로그 번호	로그 레코드
1	<T_1 start>
2	<T_1, x, 20, 30>
3	<T_1 commit>
4	<T_2 start>
5	<T_2, y, 50, 60>
6	<checkpoint>
7	<T_3 start>
8	<T_3, x, 30, 40>
9	<T_2, z, 80, 90>
10	<T_3 commit>
<-- 시스템 실패 발생 -->	

10. 다음은 학생의 학번(sID)을 연결 리스트에 입력 순서대로 저장하는 C 프로그램이다. 프로그램 코드와 <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node {
   int sID;
   struct node *next;
} Node;

Node *findLast (Node *start) {
   Node *p;
   if (start == NULL)
      return NULL;
   p = [ ㉠ ];
   while (p->next != NULL) {
      printf ("findLast : %d \n", p->sID);
      p = p->next;
   }
   return p;
}

void insert (Node *start, int s) {
   Node *n, *p;
   p = findLast (start);
   if (p != NULL) {
      printf ("insert : %d \n", p->sID);
      n = (Node *) malloc (sizeof (Node));
      n->sID = s;
      n->next = NULL;
      [ ㉡ ] = n;
   }
}

int main () {
   Node *start = NULL;
   start = (Node *) malloc (sizeof (Node));
   start->sID = 10000;
   start->next = NULL;
   insert (start, 20000);
㉢ insert (start, 30000);
   return 0;
}
```

<조 건>

- main 함수의 start 변수는 연결 리스트의 첫 번째 노드를 가리킨다.
- findLast 함수는 연결 리스트의 마지막 노드의 주소를 반환한다.
- insert 함수는 연결 리스트의 마지막 노드 다음에 새로운 학생의 학번이 저장된 노드를 추가하는 함수이다.

<작성 방법>

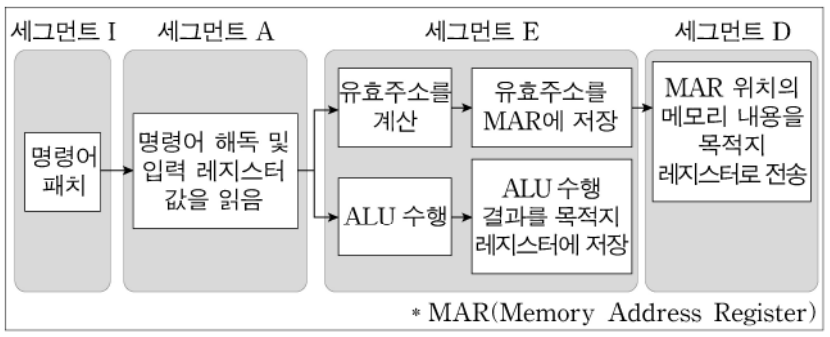
- ㉠, ㉡에 해당하는 코드를 순서대로 쓸 것.
- 프로그램의 실행 결과를 쓸 것.
- ㉢ 위치의 코드를 아래와 같이 변경한 후 실행했을 때, 연결 리스트의 마지막 노드에 저장된 학생의 학번을 쓸 것.

```
insert (start->next, 30000);
```


한국교육과정평가원

11. (가)는 RISC 파이프라인 컴퓨터의 명령어 세그먼트이고, (나)는 관련 명령어이다. (다)는 대표적인 해저드(hazard) 3가지에 대한 설명이다. (라)는 해당 컴퓨터의 메모리에 저장되어 있는 프로그램이다. (라)를 실행하면 해저드가 발생하는데, 무연산을 삽입하여 이 문제를 해결하고자 한다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)



(나)

명령어	설명
LOAD R[Y], M[X] DEC R[Y] ADD R[Y1], R[Y2], R[Y3]	M[X]를 R[Y]로 전송 R[Y]에서 1을 뺀 후 R[Y]에 저장 R[Y2]와 R[Y3]을 더한 후 R[Y1]에 저장
* M[X] : 메모리 주소 X의 내용 * R[Y] : 레지스터 파일의 Y번째 레지스터의 내용	

(다)

○ 자원 해저드(resource hazard) : 파이프라인 내의 명령어들이 동일한 클럭 사이클에 같은 메모리 모듈을 접근하는 경우 발생
○ 데이터 해저드(data hazard) : 이전 명령어의 결과가 목적지 레지스터에 저장되기 전에 다음 명령어가 해당 레지스터에 저장된 값을 사용하는 경우 발생
○ 분기 해저드(branch hazard) : 분기 명령어 이후의 잘못된 명령어들이 파이프라인에 들어온 경우 발생

(라)

삽입 위치	프로그램
	LOAD R[1], M[100]
①	LOAD R[2], M[101]
②	DEC R[1]
③	ADD R[2], R[2], R[3]
④	ADD R[1], R[1], R[3]
⑤	DEC R[1]

<조 건>

○ 각 명령어 세그먼트는 1클럭 사이클 동안 수행된다.
○ 세그먼트 D는 LOAD 명령어 경우에만 수행된다.
○ 명령어와 데이터는 서로 다른 메모리 모듈에 저장되어 동시에 접근 가능하다.
○ ALU 수행 결과와 전송된 메모리 내용은 목적지 레지스터에 값이 저장된 후, 다음 클럭 사이클부터 이용이 가능하다.
○ (라)는 첫 줄부터 순차적으로 실행된다.
○ (라)의 실행에 필요한 데이터는 실행 전에 저장되어 있다.
○ 삽입되는 무연산의 수는 최소로 하고 무연산삽입 이외의 방식은 사용할 수 없다.

<작성 방법>

○ (라)를 실행할 때 데이터가 저장된 메모리 모듈을 몇 번 접근하는지 쓸 것.
○ (다)의 3가지 해저드 중에서 (라)를 실행할 때 발생하는 해저드 1가지를 쓸 것.
○ (라)를 실행할 때 발생하는 해저드를 해결하기 위해 무연산을 삽입해야 하는 위치를 ①~⑤ 중에서 2가지 쓸 것.

<수고하셨습니다.>