2024학년도 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. (가)는 2022 개정 정보과 교육과정에 따른 인공지능 기초 과목에서 '인공지능의 사회적 영향' 영역의 '성취기준' 중 일부이고, (나)는 (가)에 해당하는 영역 '내용 체계' 중 일부이다. (다)는 (가)와 (나)를 근거로 제작한 수업 활동지이다. 괄호 안의 ㉠과 ㉡에 들어갈 용어를 순서대로 쓰시오. [2점]

(가)

[12인기03-04] 인공지능의 활용사례와 (㉠) 상황을 인공지능 (㉡) 관점에서 분석한다.

(나)

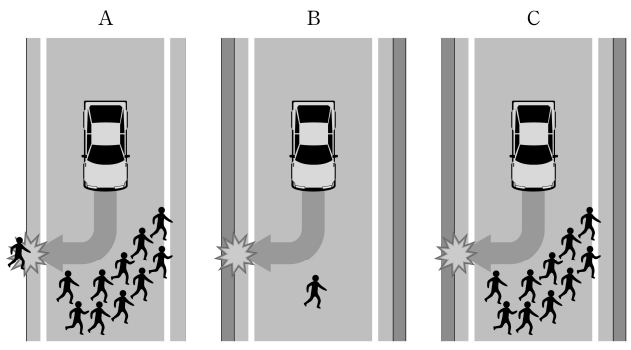
핵심 아이디어	인공지능을 올바르게 활용하기 위해서는 인공지능에 의해 발생할 수 있는 윤리적 문제에 대한 이해가 필요하다.
구분 / 범주	내용 요소
지식 · 이해	인공지능과 (㉡)
과정 · 기능	인공지능과 관련된 (㉠) 상황에 대해 논의하기
가치 · 태도	인공지능의 다양한 측면에 대한 비판적 자세와 윤리적 태도

(다)

수업 활동지

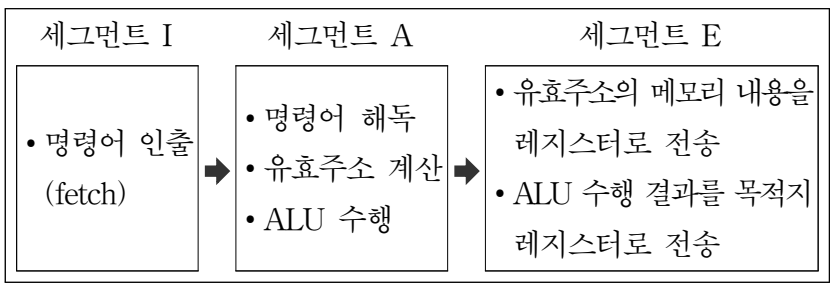
자율 주행 자동차가 주행 중에 다음과 같은 상황이 발생했을 경우 어떻게 해야 하는지, 자신의 생각을 작성해 보자.

A. 직진하게 되면 10명을 치게 되지만, 급격히 방향을 꺾으면 인도에 있던 1명을 치게 된다.
B. 직진하게 되면 보행자를 치게 되지만, 급격히 방향을 바꾸면 차에 타고 있는 사람이 크게 다치거나 죽게 된다.
C. 직진하게 되면 여러 사람이 죽거나 다치고, 급격히 방향을 꺾으면 차에 타고 있는 사람이 크게 다치거나 죽게 된다.



2. (가)는 RISC 파이프라인 컴퓨터의 명령어 세그먼트이고 (나)는 명령어 집합이다. (다)는 프로그램을 수행할 때 발생하는 데이터 해저드(hazard)를, 무연산(no-operation) 명령어를 삽입하는 지연된 적재(delayed load)로 해결한 결과이다. <작성 방법>에 따라 쓰시오. [2점]

(가)



(나)

○ LOAD R, M[X] : 주소 X의 메모리 내용을 레지스터 R에 적재
○ ADD R3, R1, R2 : 레지스터 R1과 R2의 내용을 더하여 레지스터 R3에 저장
○ NOP : 무연산(no-operation) 명령

(다)

프로그램	클럭 사이클(clock cycle) 번호									
	1	2	3	4	5	6	7	8	9	10
① LOAD R1, M[1]	I	A	E							
② LOAD R2, M[2]		I	A	E						
③ NOP			I	A	E					
④ ADD R3, R1, R2				I	A	E				
⑤ LOAD R4, M[4]					I	A	E			
⑥ LOAD R5, M[5]						I	A	E		
⑦ NOP							I	A	E	
⑧ ADD R6, R4, R5								I	A	E

<작성 방법>

○ 해당 프로그램에서 발생하는 데이터 해저드를 지연된 적재(delayed load) 대신 명령어 재배치(rearrangement)로 해결할 경우, 총 클럭 사이클(clock cycle) 수가 최소가 되는 명령어 번호 순서를 1가지 쓸 것. (단, 각 세그먼트는 한 클럭 사이클에 수행되고, 재배치된 프로그램은 명령어 ①번에서 시작하고 ⑧번으로 끝난다.)
○ 이때, 소요되는 총 클럭 사이클의 수를 쓸 것.

3. (가)는 2022 개정 중학교 정보과 교육과정에 제시된 '성취기준 적용 시 고려 사항'에 제시된 내용 중 일부이고 (나)는 '교수·학습의 방향'에 제시된 내용 중 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

문제를 해결하는 과정에서 문제 발견, 상태 정의, 핵심요소 추출 등의 추상화 단계를 거쳐 알고리즘을 설계하는 과정을 자연스럽게 경험할 수 있도록 교수·학습 절차를 설계하고, 문제 해결 과정 전반을 평가할 수 있도록 보고서나 포트폴리오 등을 활용하여 학생의 사고 과정을 누적하여 기록하도록 한다.

㉠ 학생의 수준을 고려하여 적합한 프로그래밍 언어를 선정하고, 초등학교 실과 과목에서 학습한 기초적인 프로그래밍 기능을 바탕으로 데이터를 순차적으로 저장하는 구조, 논리 연산, 중첩 제어 구조를 활용할 수 있도록 프로젝트의 수준을 적절하게 설정하도록 한다.

(나)

… (상략) …

교수·학습에 필요한 디지털 역량을 탐색하고 ㉡ 학생의 디지털 역량 수준을 파악하여 교수·학습을 진행하는 데 어려움이 없도록 추가적인 교육 기회를 제공한다.

<작성 방법>

- 2022 개정 정보과 교육과정에 따른 정보 과목의 영역 중 (가)에 해당하는 영역명을 쓸 것.
- 하인니히(R. Heinich)가 제시한 'ASSURE 모형'을 적용한 매체 활용 수업을 설계하려고 한다. 이 모형의 단계 중, 밑줄 친 (가)의 ㉠과 (나)의 ㉡에 공통으로 해당하는 단계의 명칭을 쓰고, 이 단계에서 중요하게 다루는 요소 2가지를 서술할 것.

4. (가)는 2022 개정 고등학교 정보과 교육과정에 따른 정보 과목의 배열 단원 수업을 위한 학습 지도안의 일부이고, (나)는 이를 근거로 작성한 수업 평가지의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

내용 요소	다차원 데이터 활용	수업 차시	1/3
성취 기준	[12정03-06] 다양한 데이터 (㉠)을/를 활용한 프로그램을 작성한다.		
학습 목표	○ 배열의 개념을 이해할 수 있다. ○ (㉡)		
도입	○ 배열과 변수의 차이점에 대해 알아보고, 일상생활 속에서 배열이 적용되어 있는 사례를 찾아본다. ○ 학습 목표를 확인한다.		
전개	1. 배열의 개념과 (㉠) • 배열의 개념에 대해 학습한다. • 배열의 자료형, 배열명, 배열 크기를 이해한다. • 1차원 배열과 2차원 배열을 비교하여 분석한다. - 정보 과목 성적을 계산하기 위해, 학생의 점수를 저장하기 위한 1차원 배열의 (㉠)은/는 다음과 같다.		

배열명 : score

	score[0]	score[1]	score[2]	…
	70	75	93	…

(나)

수업 평가지

1. 다음을 보고 변수와 배열의 차이점을 서술하시오.

변수

변수명	score0	score1	score2	score3	…
자료	90	85	95	80	…

배열

배열명	score				
인덱스	0	1	2	3	…
자료	90	85	95	80	…

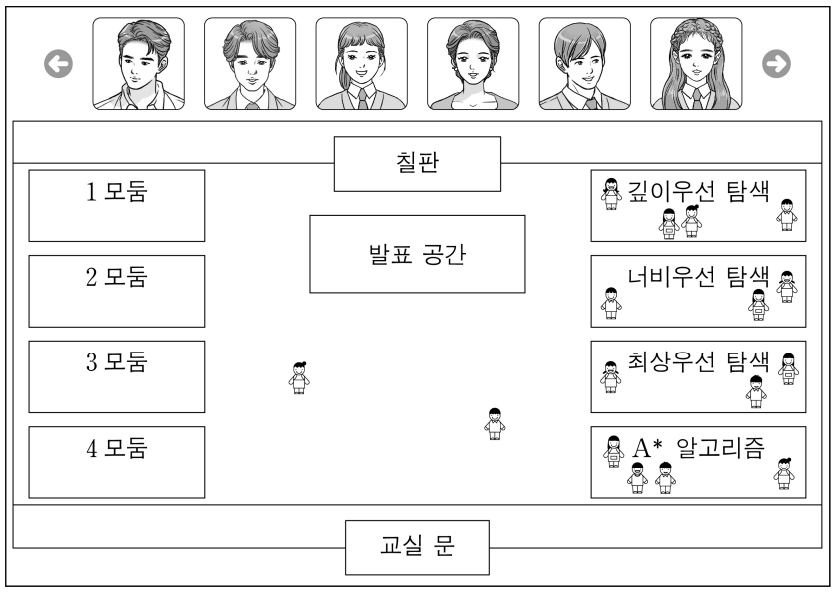
<작성 방법>

- 타일러(R. Tyler)의 학습 목표 진술 방법에 근거하여 괄호 안의 ㉠에 해당하는 용어를 사용하여 괄호 안의 ㉡을 서술할 것.
- 1차시 수업 중에 (나)를 활용하여 학습자 이해 수준을 확인하였다. 평가를 수행하는 시기의 관점에서 볼 때 이에 해당하는 평가 유형의 명칭을 쓸 것.
- 1학년 학생 100명의 정보 과목 점수가 (나)의 배열 score에 저장되어 있을 때, 배열명과 인덱스를 사용하여 모든 점수를 출력하는 다음 C언어 코드의 ⓐ에 들어갈 내용을 쓸 것.

```
for( i = 0; i < 100; i++ )
  printf("%d ", ⓐ );
```

5. (가)는 2022 개정 정보과 교육과정에 따른 인공지능 기초 과목에서 '인공지능의 이해' 영역을 온라인 환경에서 진행한 수업 화면이고, (나)는 (가) 환경에서 JIGSAW Ⅱ 협동 학습 수업을 하는 교사와 학생 간의 대화 내용의 일부이다. (다)는 (가)와 (나)의 수업을 위해 만든 전문가 집단 활동지의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)



(나)

교 사 : 오늘은 협동 학습을 이용하여 맹목적인 탐색과 정보 이용 탐색 방법을 비교·분석하고, 문제 해결을 위한 최적의 탐색 과정을 배워 보는 시간을 갖도록 하겠습니다.
학생들 : 네! 선생님.
교 사 : 현재 모둠 내에서 본인의 전문가 분야를 선택하셨나요? 지난 시간에 말한 대로, 전문가 분야는 깊이우선 탐색, 너비우선 탐색, 최상우선 탐색, A* 알고리즘입니다.
학생들 : 네! 선생님.
교 사 : 오른편에 있는 본인의 ㉠ 전문가 집단으로 이동하시고, 전문가 집단 내에서 각자의 역할을 정해 주세요. 각 전문가 집단별로 받았던 활동지를 확인한 후, 해당 전문가 분야에 대해 학습하도록 하세요.

교 사 : 모든 전문가 집단의 활동이 끝났으니 전문가들은 본인의 모둠으로 돌아가 주세요. 그리고 모둠별 활동지를 작성해 주세요.
학생들 : 네! 선생님.
교 사 : 각 모둠의 모둠장은 '발표 공간'으로 나와 활동지의 결과를 발표해 주세요.
학생 A : 선생님! 평가는 어떻게 이루어지나요?
교 사 : ㉡ 개인별 점수, ㉢ 개인별 향상 점수 그리고 ㉣ 모둠별 점수로 평가할 거예요. 개인별 향상 점수는 이전에 실시했던 퀴즈들의 평균 점수와 비교해서 얼마나 향상되었는지 평가하고, 모둠별 점수는 모둠 구성원 개인별 향상 점수의 평균으로 평가할 거예요.
학생 A : 네! 알겠습니다.

(다)

(㉤) **전문가 집단 활동지**

(㉤)은 목표 상태가 발견될 때까지 한쪽으로 가능한 모든 상태를 탐색하는 것이다. 더 탐색할 곳이 없으면 다시 이전 상태로 돌아와 다른 경로의 한쪽으로 계속 탐색한다.

문제 1. 그림의 상태 공간은 상태 A부터 F까지 트리 구조로 표현되어 있다. (㉤)으로 초기 상태 A부터 목표 상태 F까지 탐색해 보자. (단, 왼쪽 서브 트리부터 우선 탐색한다.)

탐색 순서 : (㉥)

<작성 방법>

○ JIGSAW Ⅰ 수업 방식과 JIGSAW Ⅱ 수업 방식을 비교했을 때, 두 수업 방식의 공통점을 (나)의 밑줄 친 ㉠~㉣ 부분에서 2가지를 찾아 그 기호를 쓸 것.
○ (다)의 괄호 안의 ㉤에 들어갈 내용을 (나)에 제시된 전문가 분야 중에서 1가지를 찾아 쓰고, 괄호 안의 ㉥에 들어갈 탐색 순서를 쓸 것.

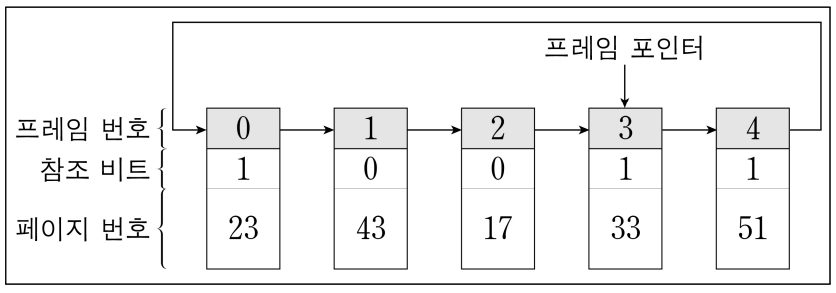
6. (가)는 가상 메모리의 페이지 교체를 위한 2차 기회 알고리즘(second chance algorithm)이다. (나)는 물리 메모리 프레임 구조의 현재 상태이고, (다)는 프로세스가 접근할 페이지의 번호들을 나열한 참조열이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

2차 기회 알고리즘(second_chance_algo)	
입력	프레임 포인터(frame_pointer)
출력	페이지 교체 대상 프레임 주소

```
typedef struct frame {
  int ref_bit; /* 참조 비트 */
  int page_num; /* 프레임에 저장된 페이지 번호 */
  struct frame* next; /* 다음 프레임 포인터 */
} Frame;
Frame* second_chance_algo(Frame* frame_pointer) {
  while (TRUE) { /* 무한 반복문 */
      /* frame_pointer가 가리키는 프레임의 참조 비트가 0
         이면, 해당 프레임의 참조 비트를 1로 설정하고
         해당 프레임 주소를 반환한다. */
      if(frame_pointer->ref_bit == 0) {
        frame_pointer->ref_bit = 1;
        return frame_pointer;
      }
      /* frame_pointer가 가리키는 프레임의 참조 비트가 1
         이면, 해당 프레임의 참조 비트를 0으로 설정하고 탐색을
         계속한다. */
      else {
        frame_pointer->ref_bit = 0;
        frame_pointer = frame_pointer->next;
      }
  }
}
```

(나)



(다)

73, 17, 51, 25, 23, 65, 23, 25, 51, 17, 89

<조 건>

- ○ 반입 정책은 요구 페이징(demand paging) 기법을 사용한다.
- ○ 본 시스템에서 프로세스의 페이지 적재를 위한 물리 메모리 프레임은 5개이다.
- ○ 프로세스가 참조하려는 페이지를 저장한 프레임이 존재하는 경우, 해당 프레임의 참조 비트를 1로 갱신한다.
- ○ 프로세스가 참조하려는 페이지를 저장한 프레임이 없는 경우, 페이지 폴트(page fault)가 발생한다.
- ○ 페이지 폴트 발생 시, 본 시스템의 페이지 교체 정책은 2차 기회 알고리즘을 사용하여 페이지 교체 대상 프레임을 선택하고, 해당 프레임에 프로세스가 참조하려는 페이지를 적재한다.
- ○ 2차 기회 알고리즘 실행에 따라 프레임 포인터는 가장 최근에 페이지 교체가 이루어진 프레임을 가리킨다.
- ○ (나)에서 시스템의 모든 프레임에 대한 페이지 교체 횟수는 0으로 초기화된 상태이다.
- ○ 프로세스는 (다)의 가장 왼쪽 페이지 번호에서 오른쪽 페이지 번호로 페이지 참조 순서를 가지며, 현재 시스템에서 프레임 포인터는 3번 프레임을 가리키고 있다.

<작성 방법>

- ○ 프로세스가 (다)의 순서대로 페이지를 참조할 때, 첫 번째로 페이지 교체가 일어난 프레임 번호를 쓸 것.
- ○ 프로세스가 (다)의 순서대로 페이지를 참조할 때, 65번 페이지를 프레임에 적재한 직후 시점까지 해당 프레임에서 발생한 페이지 교체 횟수를 쓸 것.
- ○ 프로세스가 (다)의 순서대로 모든 페이지에 대한 참조를 수행한 직후, 각 프레임 상태를 [프레임 번호, 참조 비트, 저장된 페이지 번호]의 순서화된 형식으로 연관 지어 쓸 것.

7. (가)에서 설명하는 요구사항을 기반으로 (나)의 SQL문을 통해 릴레이션(relation)을 생성하였다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

프로젝트 관리 시스템에 저장되는 직원과 관련한 정보는 사번, 이름, 연락처, 프로젝트명, 예산, 멘토사번, 평가부서이다. 사번은 회사 입사에 따라 직원에게 부여되는 고유한 값이고, 이름은 직원의 성명이며, 연락처는 각 직원별로 연락이 가능한 사무실 전화번호이다. 프로젝트명은 프로젝트를 구분하도록 부여한 고유한 이름이고, 예산은 프로젝트별로 소요되는 비용이다. 사내 직원 간 멘토링 활동을 위해 직원별로 한 명의 멘토가 존재하며, 멘토는 여러 명의 직원을 멘토링할 수 있다. 멘토링에 대한 평가 결과는 멘토가 소속된 부서에 제출하여 처리하기 때문에 시스템에서는 멘토의 소속부서가 평가부서가 된다.

(나)

```
CREATE TABLE 직원 (
  사번 VARCHAR(16),      이름 VARCHAR(16),
  연락처 VARCHAR(16),    프로젝트명 VARCHAR(32),
  예산 INT,              멘토사번 VARCHAR(16),
  평가부서 VARCHAR(16),
  PRIMARY KEY(사번, 프로젝트명)
);
```

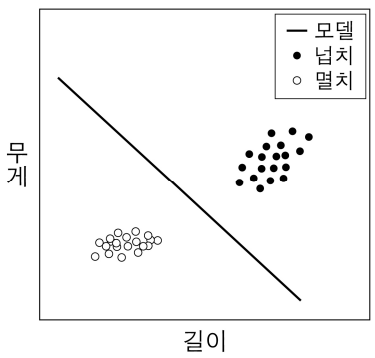
<조 건>

○ (가)에서 설명한 요구사항 이외의 조건은 고려하지 않는다.

<작성 방법>

○ (나)를 통해 생성한 릴레이션 내에 발생한 부분 종속(partial dependency)을 구성하는 속성(attribute)을 한 쌍 쓸 것.
○ (나)를 통해 생성한 릴레이션으로부터 부분 종속을 제거함에 따라 만족하는 정규형을 쓸 것.
○ (가) 요구사항에서 밑줄 친 부분을 '직원별로 한 명 이상의 멘토들이 존재하며'로 수정하는 경우, (나)를 통해 생성한 릴레이션을 BCNF(Boyce-Codd Normal Form)를 만족하도록 분해하기 위한 후보키를 2개 쓸 것.

8. 다음은 지도 학습(supervised learning) 중 대표적인 2가지 학습 문제 유형에 대한 특징이다. <작성 방법>에 따라 서술하시오. [4점]

학습문제유형	특징
(㉠)	• 주어진 입력 데이터-출력 데이터 쌍을 이용하여 학습한다. • 학습이 끝난 후, 임의의 입력 x가 주어지면 그것에 대한 출력 y를 예측할 수 있다. 
(㉡)	• 입력 데이터와 그 데이터가 어느 클래스(범주)에 속하는지를 나타내는 레이블이 학습 데이터로 주어진다. • 입력을 두 개 이상의 클래스로 구분한다. • 학습이 끝난 후, 입력 데이터가 주어지면 그 입력 데이터가 속하는 클래스를 예측할 수 있다.

<작성 방법>

○ 괄호 안의 ㉠과 ㉡에 들어갈 지도 학습의 대표적인 학습 문제 유형명을 순서대로 쓸 것.
○ ㉠과 ㉡ 중에서 다음 문제를 해결하는 데 적합한 지도 학습 문제 유형명을 쓰고, 학습 데이터 구성 방안을 서술할 것.

<문 제>

대학 캠퍼스 주위에 많은 원룸이 있다. 캠퍼스까지 걸어서 갈 수 있는 거리가 가까울수록 원룸 임대 가격이 높아진다. 캠퍼스 주위의 원룸 임대 가격과 캠퍼스까지의 거리를 조사하고, 이를 토대로 조사하지 않은 나머지 원룸 임대 가격을 예측하는 지도 학습 모델을 구성하고자 한다.

9. (가)는 주어진 문자열을 변환하여 암호문을 생성해 주는 C 프로그램이고, (나)는 아스키 코드 표의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

```
#include <stdio.h>
#include <string.h>
#define LENGTH   128

void code1(char enc[], int s) {
  int  i;
  for(i = 0; i < s; i++) {
    if(enc[i] >= 65 && enc[i] <= 90) enc[i] += 32;
    else if(enc[i] >= 97 && enc[i] <= 122) enc[i] -= 32;
  }
}

void code2(char enc[], int s, int n) {
  int i;
  char temp[LENGTH];
  strcpy(temp, enc);
  for(i = 0; i < s; i++) enc[i] = temp[(i + n) % s];
}

void main() {
  char enc[LENGTH] = "RedHat";
  int n = 2;
  int s;

  s = strlen(enc);
  code1(enc, s);
㉠ printf("%s \n", enc);

  code2(enc, s, n);
㉡ printf("%s \n", enc);
  ㉢ ;
  ㉣ ;

  printf("%s \n", enc);
}
```

(나)

문자	Dec	Hex	문자	Dec	Hex	문자	Dec	Hex	문자	Dec	Hex
A	65	41	N	78	4E	a	97	61	n	110	6E
B	66	42	O	79	4F	b	98	62	o	111	6F
C	67	43	P	80	50	c	99	63	p	112	70
D	68	44	Q	81	51	d	100	64	q	113	71
E	69	45	R	82	52	e	101	65	r	114	72
F	70	46	S	83	53	f	102	66	s	115	73
G	71	47	T	84	54	g	103	67	t	116	74
H	72	48	U	85	55	h	104	68	u	117	75
I	73	49	V	86	56	i	105	69	v	118	76
J	74	4A	W	87	57	j	106	6A	w	119	77
K	75	4B	X	88	58	k	107	6B	x	120	78
L	76	4C	Y	89	59	l	108	6C	y	121	79
M	77	4D	Z	90	5A	m	109	6D	z	122	7A

<작성 방법>

- ㉠의 출력 결과를 쓸 것.
- ㉡의 출력 결과를 쓸 것.
- ㉡이 실행된 이후, 배열 enc의 내용을 초깃값 "RedHat"로 복호화하기 위하여, ㉢과 ㉣에 필요한 함수 호출을 code1()과 code2() 함수를 사용하여 쓸 것.

10. (가)는 어떤 IPv4 데이터그램의 헤더 정보 중 일부이다. 이 데이터그램을 단편화(fragmentation)할 때, <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)

필드(field) 이름	값(value)
Total length	820
Identification	456
MF(More Fragments) flag bit	0
Fragment offset	0
… (생략) …	

<조 건>

- IPv4 데이터그램의 헤더 길이는 20바이트이다.
- 최대 전송 단위(MTU : Maximum Transmission Unit)는 500바이트이다.
- 단편화되는 데이터그램의 Total length는 마지막 데이터그램을 제외하고 모두 500이다.
- DF(Don't Fragment) flag bit는 0으로 설정되어 있다.

<작성 방법>

- (가)의 데이터그램이 몇 개의 데이터그램으로 단편화되는지 쓸 것.
- 단편화되는 데이터그램의 Total length 값들을 모두 더하면 얼마가 되는지 쓸 것.
- 단편화되는 데이터그램들의 헤더 정보를 (가)의 양식을 참고하여 표현할 것.

11. (가)는 0-1배낭채우기 알고리즘의 의사코드이고, (나)는 (가)에 대한 설명이다. (다)는 물건 7개에 대한 정보이다. (라)는 (가)의 알고리즘에 배낭의 총용량 8, 물건의 개수 7 그리고 (다)의 배열 w와 v를 입력하여 나온 결과인 배열 K이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

0-1배낭채우기 알고리즘

- 입력 : 배낭의 총용량 `C`, 물건의 개수 `n`, 각 물건의 용량이 저장된 배열 `w`, 각 물건의 가치가 저장된 배열 `v` (단, 물건 번호 `i`의 용량은 `w[i]`에, 가치는 `v[i]`에 저장된다.)
- 출력 : 배낭의 용량별로 선택할 수 있는 물건들의 조합의 최대 가치를 계산한 값을 저장한 2차원 배열 `K`

```
for 0 ≤ i ≤ n :  K[i][0]를 0으로 초기화
for 0 ≤ j ≤ C :  K[0][j]를 0으로 초기화
for 1 ≤ i ≤ n :
  for 1 ≤ j ≤ C :
    if [ ㉠ ] :
      K[i][j] ← K[i-1][j]
    else :
      K[i][j] ← max(K[i-1][j], K[i-1][j-w[i]] + v[i])
return K
```

(나)

○ 배낭의 총용량(C) 및 n개의 물건들에 대한 각각의 용량과 가치가 주어졌을 때, 배낭의 총용량을 넘지 않는 범위 내에서 가치를 합산한 값이 최대가 되도록 물건을 선택하여 담고, 그때의 합산한 가치, 즉 최대 가치를 구하는 알고리즘이다. (단, 어떤 물건도 그 물건의 일부만 분리하여 배낭에 담을 수는 없다.)

○ 다음은 (가) 알고리즘에 대한 관계식이다.

- j가 0이고, $0 \le i \le n$ 인 경우, $K[i, 0] = 0$
- i가 0이고, $0 \le j \le C$ 인 경우, $K[0, j] = 0$
- $1 \le i \le n$, $1 \le j \le C$ 이고, $w[i]$가 j보다 큰 경우, $K[i, j] = K[i-1, j]$
- $1 \le i \le n$, $1 \le j \le C$ 이고, $w[i]$가 j보다 작거나 같은 경우, $K[i, j] = max(K[i-1, j],\ K[i-1, j-w[i]] + v[i])$

○ 여기서, $max(a, b)$는 다음과 같이 정의한다.

- $a \ge b$인 경우, $max(a,\ b) = a$
- $a < b$인 경우, $max(a,\ b) = b$

(다)

물건의 번호 : i	1	2	3	4	5	6	7
용량이 저장된 배열 : w	7	1	5	8	2	4	3
가치가 저장된 배열 : v	8	2	9	7	6	8	9

(라)

i \ j	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	8	8
2	0	2	2	2	2	2	2	8	10
3	0	2	2	2	2	( )	11	11	11
4	0	2	2	2	2	9	11	11	11
5	0	2	6	8	8	9	11	15	17
6	0	2	6	8	8	10	14	16	17
7	0	2	6	9	11	15	17	17	19

<작성 방법>

○ (가)의 알고리즘이 올바르게 동작하도록 ㉠에 들어갈 코드를 쓸 것.

○ (라)의 `K[3][5]`에 있는 괄호 안에 들어갈 숫자를 쓰고, 그 계산 과정을 쓸 것.

○ (가)와 (다)에 근거하여 배낭의 총용량 8, 물건의 개수 7일 때 최대 가치를 만드는 물건들의 번호를 오름차순으로 쓸 것.

<수고하셨습니다.>