

2020학년도 중등학교교사 임용후보자 선정경쟁시험 대비 실전 모의고사

정보 · 컴퓨터

제1차 시험	3 교시 전공 B	11문항 40점	시험 시간 90분
--------	-----------	----------	-----------

응시자 유의 사항

1. 문제지 및 답안지의 전체 면수와 인쇄 상태를 확인하십시오.
◇ 답안지는 2면입니다.
2. 답안지 모든 면의 상단에 성명과 수험 번호를 기재하고, 검은색 펜을 사용하여 수험 번호의 해당란을 '●'로 표기하십시오.
◇ '●'로 표기한 부분을 수정하고자 할 경우에는 반드시 수정 테이프를 사용해야 합니다.
3. 답안의 초안 작성은 문제지 여백을 활용하십시오.
4. 각각의 문항에 대한 답안은 해당 문항의 답안란에 작성하십시오.
◇ 답안지에는 문항 내용을 기재하지 않습니다.
5. 답안은 지워지거나 번지지 않는 동일한 종류의 검은색 펜을 사용하여 작성하십시오.
◇ 연필이나 사인펜 종류는 사용할 수 없습니다.
6. 답안을 작성할 때, 가로 선을 그어 답안란의 줄을 추가하거나 세로 선을 그어 답안란을 다단으로 구분할 수 있으니, 필요한 경우에 활용하십시오.
◇ 단, 가로 선과 세로 선은 해당 답안란 내에서만 활용할 수 있습니다.
7. 답안을 수정할 때에는 반드시 두 줄(=)을 긋고 수정할 내용을 작성하십시오.
◇ 수정 테이프 또는 수정액을 사용하여 답안을 수정할 수 없습니다.
8. 문항에 대한 답안 내용 이외의 것(답안의 특정 부분을 강조하기 위한 밑줄이나 기호 등)은 일절 표시하지 마십시오.
◇ 단, 일반적인 글쓰기 교정 부호는 사용이 가능합니다.
9. 문항에서 요구하는 내용의 가짓수가 제한되어 있는 경우, 요구한 가짓수까지의 내용만 답안으로 작성하십시오.
◇ 첫 번째로 작성한 내용부터 문항에서 요구한 가짓수에 해당하는 내용까지만 순서대로 채점합니다.
10. 다음에 해당하는 답안은 채점하지 않으니 유의하십시오.
◇ 다른 문항의 답안란에 작성한 부분(문항 번호를 임의로 수정하는 경우, 맞바꿔 작성한 부분을 화살표로 표시하는 경우 등 포함)
◇ 답안란 이외의 공간(옆면, 뒷면 등)에 작성한 부분
◇ 내용이 지워지거나 번지는 등 식별이 불가능한 부분
◇ 연필로 작성한 부분, 수정 테이프 또는 수정액을 사용하여 수정한 부분
◇ 개인 정보를 노출하거나 암시하는 표시(성명 및 수험번호 기재란 제외)가 있는 답안지 전체
11. 답안지 교체가 필요한 경우에는 답안 작성 시간을 고려하십시오.
◇ 종료종이 울리면 답안을 일절 작성할 수 없으며, 답안지 교체 후에는 교체 전 답안지를 폐답안지로 처리합니다.
12. 시험 종료 전까지 답안 작성을 완료하십시오.
◇ 시험 종료 후 답안 작성은 부정행위로 간주합니다.
13. 답안을 작성하지 않은 빈 답안지에도 성명, 수험 번호를 기재·표기한 후, 답안지를 모두 제출하십시오.
14. 위의 사항을 위반하여 작성한 답안은 채점 시 불이익을 받을 수 있으니 유의하십시오.

※ 시험이 시작되기 전까지 표지를 넘기지 마시오.

3. 다음은 OO 중학교 정보과 컴퓨팅 시스템 영역의 <성취기준>과 <학습자와 학습 환경 분석> 결과이다. 이 사항들을 고려하여 피지컬 컴퓨팅 수업을 어떻게 설계해야 할지를 서술하시오. [4점]

<성취기준>

[9정05-01] 컴퓨팅 시스템을 구성하는 하드웨어와 소프트웨어의 역할을 이해하고 유기적인 상호 관계를 분석한다.

[9정05-02] 센서를 이용한 자료 처리 및 동작 제어 프로그램을 구현한다.

<학습자 분석>

처음으로 피지컬 컴퓨팅을 접해서 관심이 적고, 학습자의 수준이 낮아서 프로그램을 구현하기가 어려움이 있다.

<학습 환경 분석>

센서의 개수가 학습자 수의 절반 정도로 적다.

4. 다음은 중학교 정보과 수업에서 학생들에게 주어진 활동지 첫 부분이다. 비구조화된 문제 상황이 주어질 때, 컴퓨팅 사고력을 기반으로 문제를 해결하고자 한다. 괄호 안의 ㉠에 해당하는 학습 내용을 쓰고 그 이유를 서술하시오. [4점]

<성취기준>

[9정03-01] 실생활 문제 상황에서 문제의 현재 상태, 목표 상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석한다.

[9정03-02] 문제 해결에 필요한 요소와 불필요한 요소를 분류한다.

[9정03-03] 논리적인 문제 해결 절차인 알고리즘의 의미와 중요성을 이해하고 실생활 문제의 해결과정을 알고리즘으로 구상한다.

[9정03-04] 문제 해결을 위한 다양한 방법과 절차를 탐색하고 명확하게 표현한다.

<비구조화된 문제 상황>

1인당 929.9g. 한국인들이 매일 버리는 쓰레기 양이다. 음식물 쓰레기가 가장 많고, 종량제봉투에 넣어 버리는 것들도 절반 이상은 재활용할 수 있는 것들이다.

... (중략) ...

종량제봉투에 담긴 쓰레기 중에는 종이류(28.5%)가 가장 많았다. 화장지(21.1%)와 플라스틱(20.8%), 음식물(4.8%)도 많이 담겼다. 종이, 플라스틱, 유리, 금속, 건전지 등 재활용할 수 있는 자원이 종량제봉투 쓰레기의 53.7%를 차지했다.

[출처]

http://news.khan.co.kr/kh_news/khan_art_view.html?art_id=201803291201001#csidxd00a4b817657022a4ec7b04430ccd38



[문제 정의]

- 사람이 일일이 분리수거를 하기에는 번거로움이 있다. 자동으로 분리수거를 하도록 할 수는 없을까?



[문제 분석]

㉠

5. (가)는 고등학교 정보 과목의 성취기준이고, (나)는 수업단계이고, (다)는 수업에 활용할 동영상에 대한 정보이다. <작성 방법>에 따라 서술하시오. [4점]

(가) 성취기준

[12정보05-03] 문제 해결에 적합한 하드웨어를 선택하여 컴퓨팅 장치를 구성한다.

[12정보05-04] 퍼지컬 컴퓨팅 장치의 동작을 제어하기 위한 프로그램을 작성한다.

(나) 수업 단계 : 문제 발견 및 정의

(다) 동영상

제목 : 하절기 에너지 절약

내용 : 폭염대비 하절기 에너지 절약법을 미리 숙지하여 다가오는 여름에도 에너지 절약을 습관화합시다.



[출처]

2019 대구 서구청 공식 블로그 <https://blog.naver.com/happyseogu/221558682689>

<작성 방법>

- (1) 제시된 동영상을 활용한 교사의 활동을 서술할 것.
- (2) (1)의 결과로 정의된 문제 예를 1가지 제시하고 설명할 것.

6. 어떤 회사가 어느 ISP(Internet Service Provider)으로부터 210.150.64.0/23로 시작하는 클래스 없는(classless) IP 주소 블록을 부여받았다. 다음의 <조건>과 같이 16개의 서브넷이 필요하여 가변길이 서브넷을 구성하고자 한다. 부여받은 주소 블록을 첫 번째 그룹부터 시작하여 낮은 주소부터 차례로 할당할 때, <작성 방법>에 따라 서술하시오. [4점]

<조건>

- A 그룹은 각각 60개의 주소를 가지는 2개의 서브넷을 필요로 한다.
- B 그룹은 각각 30개의 주소를 가지는 2개의 서브넷을 필요로 한다.
- C 그룹은 각각 14개의 주소를 가지는 4개의 서브넷을 필요로 한다.
- D 그룹은 각각 7개의 주소를 가지는 8개의 서브넷을 필요로 한다.

<작성 방법>

- (1) B 그룹에 할당된 IP주소 블록의 첫 번째 IP주소와 마지막 IP주소를 CIDR 표기법으로 순서대로 쓸 것.
- (2) C 그룹에 할당된 IP주소 블록의 첫 번째 IP주소와 마지막 IP주소를 CIDR 표기법으로 순서대로 쓸 것.
- (3) D 그룹에 있는 마지막 서브넷의 서브넷 마스크를 쓸 것.
- (4) 부여받은 주소 블록에서 모든 서브넷의 주소를 할당한 후 여러분의 주소 개수를 구하여 쓸 것.

7. 순수한 페이지징 시스템은 4096바이트 크기의 페이지와 0부터 4095까지 번호가 매겨진 4096개 페이지의 가상 메모리가 있고, 0부터 511까지 번호가 매겨진 512개의 페이지 프레임을 가진 물리적 메모리를 가진다. 물리적 메모리의 현재 내용은 다음과 같다. <작성 방법>에 따라 기술하시오. [4점]

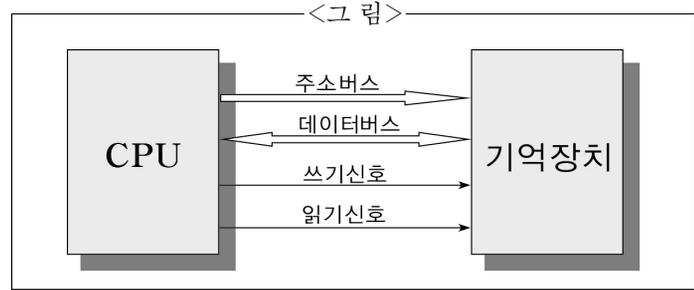
<물리적 메모리>

주소	페이지 프레임
0	페이지 테이블
4096	페이지 27
8192	페이지 15
	사용가능
20480	페이지 7
20576	페이지 109
	사용가능
28672	페이지 23
32768	페이지 13

<작성 방법>

- (1) 가상주소 28,687에 대한 참조되는 물리주소를 구할 것.
- (2) 물리주소 32,789에 대한 참조되는 가상주소를 구할 것.
- (3) 가상주소 12,301에 대해 참조할 경우 발생하는 문제점을 쓸 것.
- (4) 물리주소 46,521에 대한 페이지 프레임 번호와 변위(offset)의 크기를 구할 것. (단, 표기는 (f, d)로 한다)

8. 아래의 <그림>에서 주소 버스(address bus)는 12bit이고, 데이터 버스(data bus)는 16bit라 가정하고, 명령어는 연산코드(opcode)가 4bit이고 오퍼랜드(operand)가 12bit로 구성된다. <작성 방법>에 따라 기술하시오. [4점]



<작성 방법>

- (1) CPU가 수행할 수 있는 연산의 종류는 최대 몇 가지인지를 쓸 것.
- (2) 오퍼랜드가 2의 보수(complement) 데이터라면, 표현할 수 있는 데이터의 범위를 쓸 것. (단, 표현은 2의 지수승으로 표현한다)
- (3) 오퍼랜드가 기억장치 주소를 가리킨다면, 이 명령어에 의해 직접 주소지정 될 수 있는 기억 장소들의 수는 최대 몇 개인지를 쓸 것.
- (4) 기억장치의 주소가 바이트 별로 지정된다면, 최대 기억장치 용량을 구할 것.

9. 다음은 8-퍼즐 문제에 대해 A* 탐색 알고리즘을 이용해 초기 상태에서 목표상태로 이동하고자 한다. 다음의 <조건>을 보고 <작성 방법>에 따라 서술하시오. [4점]

— <조건> —

○ 초기상태와 목표상태는 다음과 같다.

2	4	3
6	7	5
8		1

 \Rightarrow

6	2	3
4		5
8	7	1

초기상태

목표상태

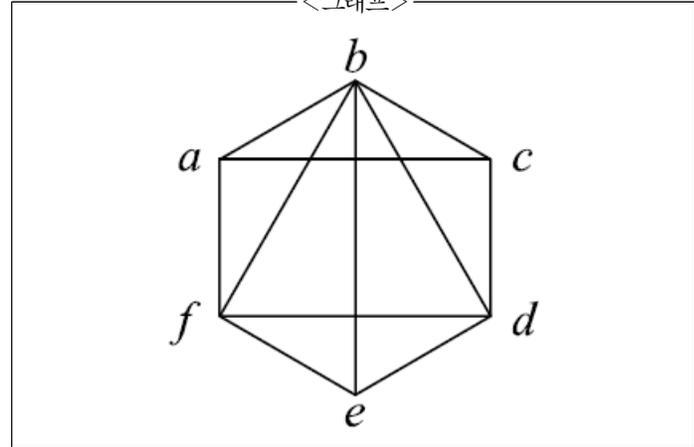
- 평가함수는 $f(n) = g(n) + h(n)$ 이다.
 - $g(n)$: 초기상태에서 n 노드까지의 최단거리이다.
 - $h(n)$: 목표 상태와 같지 않은 위치에 있는 타일 수로 한다.
- 평가함수 값을 작을수록 우선순위가 높다.
- 연산자는 위(up), 아래(down), 좌(left), 우(right)이다.
- 제약조건은 한 번에 한 칸씩만 퍼즐의 내부에서 움직인다.
- 초기상태부터 시작하여 한 번 평가된 퍼즐은 더 이상 평가하지 않는다.

— <작성 방법> —

- (1) 시작상태에서 두 번째로 선택된 8-퍼즐의 평가함수 값을 구할 것.
- (2) 시작상태에서 세 번째로 선택된 8-퍼즐의 상태를 그릴 것.
- (3) 시작상태에서 목표상태까지 총 이동거리를 구할 것.
- (4) 시작상태에서 목표상태까지 평가된 총 8-퍼즐의 개수를 구할 것.

10. 다음과 같은 <그래프 G>를 보고 아래의 <작성 방법>에 따라 기술하시오. [4점]

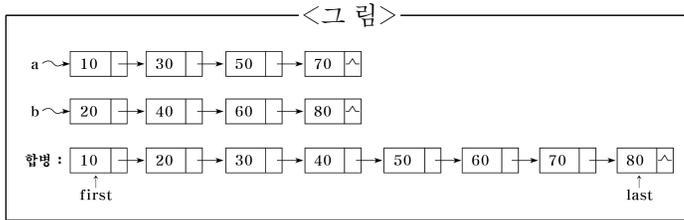
— <그래프> —



— <작성 방법> —

- (1) 그래프 G에서 정점의 수와 간선의 수를 이용하여 영역의 수를 구할 것.
- (2) 그래프 G에서 평면 그래프인 동형 그래프를 그릴 것.
- (3) 평면 그래프는 오일러 그래프인지 해밀턴 그래프인지를 판별할 것.

11. 다음 <그림>은 포인터 a, b가 가리키는 각각의 리스트를 노드 단위로 하나씩 합병(Merge)하여 새로운 리스트를 리턴하기 위한 것이다. 다음 <알고리즘>의 빈칸 ㉠, ㉡, ㉢, ㉣, ㉤에 들어갈 내용을 순서대로 쓰시오. (단, 새로운 리스트의 첫 노드를 가리키는 포인터를 first, 마지막 노드를 가리키는 포인터를 last라고 한다.) [4점]



<알고리즘>

```

struct node
{ int key ;
  struct node *next ; } ;

Merge(struct node a, struct node b)
{ struct node first, last, temp ;
  first = last =  ㉠ ;
  while( a != NULL && b != NULL )
  {
    if ( a != NULL )
    {
      temp = malloc(node) ;
       ㉡ ;
      temp->next = NULL ;
      if( first==NULL )
        first = temp ;
      if( last != NULL )
         ㉢ ;
      last = temp ;
      a = a->next ;
    }
    if ( b != NULL )
    {
      temp = malloc(node) ;
       ㉣ ;
      temp->next = NULL ;
      if( first==NULL )
        first = temp ;
      if( last != NULL )
         ㉤ ;
      last = temp ;
      b = b->next ;
    }
  }
  return(first) ;
}

```

<수고하셨습니다.>

이 면은 여백입니다.