2026학년도 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 2022 개정 정보과 교육과정(교육부 고시 제2022-33호)을 토대로 교사들이 중학교와 고등학교 '정보' 교과의 목표를 비교·분석하기 위하여 정리한 내용 중 일부이다. ㉠~㉤ 중 학교급에 맞게 기술된 목표를 찾아 기호로 쓰고, 그 목표와 가장 관련 있는 '정보' 교과의 영역을 쓰시오. [2점]

중학교
㉠ 다양한 학문 분야의 문제 해결에 필요한 데이터의 관계를 모델링하고 알고리즘을 효율적으로 설계하여 프로그램으로 구현, 평가, 개선하는 과정에서 협력과 공유의 문화를 실천하는 태도를 기른다. ㉡ 지능 에이전트의 관점에서 인공지능을 이해하고, 기계학습을 통한 인공지능으로 문제를 해결하는 방법을 체득하고 적용하는 능력을 기른다. ㉢ 디지털 기술로 인한 사회의 발전과 변화를 이해하고, 정보 보호와 정보보안의 중요성을 인식하여 실천하는 태도와 능력을 기른다.
고등학교
㉣ 디지털 세상을 연결하는 컴퓨팅 시스템 간의 연결 원리를 파악하고, 정보를 다루는 시스템에 의해 처리·생성된 결과가 공유되도록 하는 시스템 제어 능력을 기른다. ㉤ 컴퓨터로 처리되는 정보의 원리를 이해하고 다양한 현상의 의미를 해석하는 데 도움이 되는 데이터의 중요성을 고려하여 데이터의 수집 및 분석, 처리를 위한 능력을 기른다. ㉥ 컴퓨팅을 활용한 실생활의 문제 해결을 위해 문제를 발견, 분석, 추상화하여 해결책을 구상하고, 프로그램을 설계·구현하는 과정에서 자동화의 필요성과 중요성을 이해하고 실천하는 태도를 기른다.

2. 다음은 회문(palindrome) 여부를 판단하는 파이썬 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 쓰시오. [2점]

```
def reverse(src):
   dst = []  # 비어 있는 리스트 생성
   n = len(src)
   for i in range(n):
      dst.append(src[ ㉠ ])
   return dst

def isSame(src, dst):
   n = len(src)
   for i in range(n):
      if (src[i] != dst[i]):
         ㉡
   return 1

s = ['R', 'E', 'F', 'E', 'R']
d = reverse(s)
if (isSame(s, d)):
   print("회문입니다.")
else:
   print("회문이 아닙니다.")
```

<조 건>

○ 회문은 문자열을 구성하는 문자의 순서가 앞에서부터 뒤로 읽거나 뒤에서부터 앞으로 읽어도 동일한 문자열을 의미하며, 'SOS', 'RADAR', 'LEVEL' 등이 이에 해당한다.
○ src와 dst가 리스트 자료형일 때, len(src)는 src 요소 개수를 반환하고, dst.append(x)는 dst 끝에 요소 x를 추가한다.
○ reverse()는 리스트 내의 문자가 저장된 순서를 역순으로 바꾼 결과를 반환한다.
○ isSame()은 두 문자열이 동일한지 여부를 반환한다.

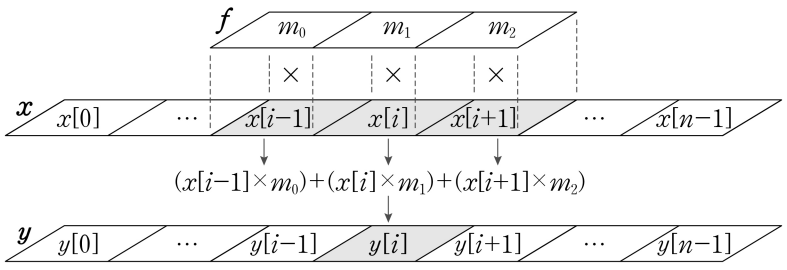
<작성 방법>

○ 빈칸 안의 ㉠에 들어갈 코드를 쓸 것.
○ 빈칸 안의 ㉡에 들어갈 코드를 쓸 것.

3. (가)는 데이터 필터링(filtering)에 대한 설명이고, (나)는 데이터 경계(edge)를 찾는 필터링을 구현한 C 프로그램의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

○ 그림에서 $\boldsymbol{x}$는 크기가 n인 입력 데이터, $\boldsymbol{f}$는 필터 계수의 값이 $[m_0, m_1, m_2]$인 필터 마스크(filter mask), 그리고 $\boldsymbol{y}$는 필터링된 결과를 나타낸다.



○ i 위치에서의 필터링 결과 $y[i]$는 다음과 같이 계산한다.

$$y[i] = (x[i-1] \times m_0) + (x[i] \times m_1) + (x[i+1] \times m_2)$$

○ 필터 마스크의 크기가 1×3이면 데이터의 양끝을 제외한 $i = 1$부터 $i = n-2$까지 $y[i]$를 계산한다. 이 과정을 데이터 필터링이라고 한다.

○ 데이터 경계는 임계치(threshold)보다 큰 $y[i]$ 값이 나타나는 위치(인덱스 i)이다.

(나)

```
… (상략) …
#define SIZE 8
typedef [ ㉠ ] char PIXEL;

void filtering(PIXEL x[], PIXEL y[], int f[]) {
   int i, j;

   for (i = 1; i < SIZE - 1; i++) {
      for (j = 0; j < 3; j++)
         [ ㉡ ];
   }
}

int main(void) {
   PIXEL x[SIZE] = {225, 225, 225, 225, 5, 5, 5, 5};
   PIXEL y[SIZE] = {0, 0, 0, 0, 0, 0, 0, 0};
   int f[3] = {1, 0, -1};
   int i;

   filtering(x, y, f);

㉢ for (i = 0; i < SIZE; i++)
      printf("%d ", y[i]);

   return 0;
}
```

<작성 방법>

○ 배열 x, y에 저장하는 값의 범위가 0~255일 때, 빈칸 안의 ㉠에 들어갈 키워드(keyword)를 쓸 것.

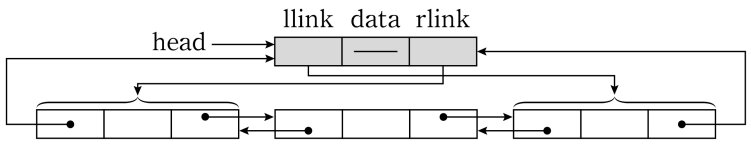
○ 빈칸 안의 ㉡에 들어갈 코드를 쓸 것.

○ ㉢의 출력 결과를 쓰고, 배열 y에서 경계가 나타나는 위치(인덱스)를 쓸 것. (단, 임계치는 100이다.)

한국교육과정평가원

4. (가)는 원형 이중연결리스트 구조이고, (나)는 원형 이중연결리스트에 노드를 삽입하는 C 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)



(나)

```
typedef struct Node {
   int data;
   [ ㉠ ] llink;
   [ ㉠ ] rlink;
} DListNode;

void init(DListNode *phead) {
   phead->llink = phead; phead->rlink = phead;
}

void printDlist(DListNode *phead) {
   DListNode *p;
   for (p = phead->rlink; p != phead; [ ㉡ ])
      printf("%d ", p->data);
}

void dinsertLeft(DListNode *phead, int data) {
   DListNode *newnode =
            (DListNode *)malloc(sizeof(DListNode));
   newnode->data = data;
   newnode->rlink = phead;
   newnode->llink = phead->llink;
   [ ㉢ ];
   phead->llink = newnode;
}

void dinsertRight(DListNode *phead, int data) {
   DListNode *newnode =
            (DListNode *)malloc(sizeof(DListNode));
   newnode->data = data;
   newnode->llink = phead;
   newnode->rlink = phead->rlink;
   phead->rlink->llink = newnode;
   phead->rlink = newnode;
}

int main(void) {
   DListNode *head =
      (DListNode *)malloc(sizeof(DListNode));
   init(head);
   dinsertRight(head, 10);  dinsertRight(head, 15);
   dinsertRight(head, 30);  dinsertRight(head, 20);
   dinsertLeft(head, 5);
㉣ printDlist(head);
   return 0;
}
```

<조 건>

○ head 노드에는 데이터가 없다.
○ printDlist()는 head 노드의 오른쪽 노드부터 출력한다.
○ dinsertRight()는 새 노드를 head의 오른쪽에, dinsertLeft()는 새 노드를 head의 왼쪽에 삽입한다.

<작성 방법>

○ 빈칸 안의 ㉠, ㉡, ㉢에 들어갈 코드를 순서대로 쓸 것.
○ 밑줄 친 ㉣의 실행 결과를 쓸 것.

5. (가)는 패리티 비트(parity bit)에 대한 설명이고, (나)는 5개의 데이터와 이에 대한 2차원 짝수 패리티의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

○ 데이터와 패리티 비트를 포함하여 1의 개수가 짝수가 되도록 만들어 주는 비트를 짝수 패리티 비트, 홀수가 되도록 만들어 주는 비트를 홀수 패리티 비트라고 한다.
○ 다음은 짝수 패리티의 예이다.



○ 2차원 패리티 방식에서는 데이터를 2차원 배열로 구성하여 행 방향(수평 방향)과 열 방향(수직 방향)으로 패리티 비트를 추가한다.
  • 수평 패리티 비트: 행 데이터에 대한 패리티 비트
  • 수직 패리티 비트: 열 데이터에 대한 패리티 비트
  • 전체 데이터 패리티 비트: 전체 데이터에 대한 패리티 비트

(나)

								수평 패리티 비트
데이터 1	1	1	0	0	1	1	0	0
데이터 2	0	1	1	0	0	1	1	0
데이터 3	1	0	0	0	0	1	1	1
데이터 4	1	1	1	1	0	0	0	0
데이터 5	0	0	0	1	1	0	0	0
수직 패리티 비트	(㉠)	(㉡)	0	(㉢)	0	(㉣)	0	(㉤)

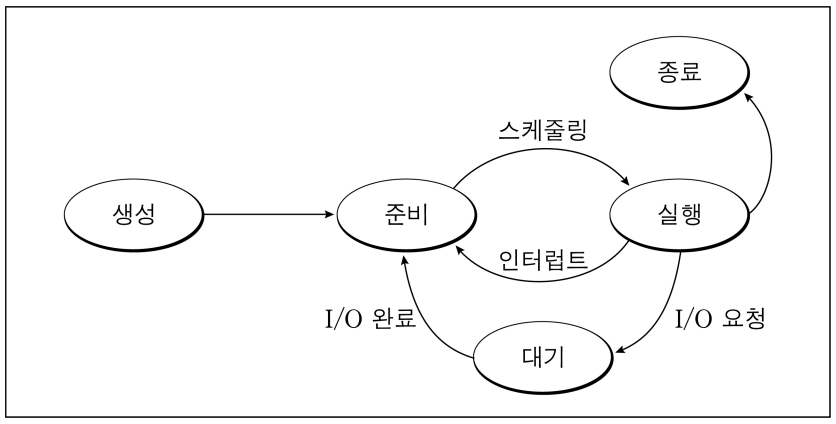
전체 데이터 패리티 비트

<작성 방법>

○ 괄호 안의 ㉠~㉣에 들어갈 패리티 비트 값을 순서대로 쓰고, 괄호 안의 ㉤에 들어갈 전체 데이터 패리티 비트 값을 쓸 것.
○ 한 종류의 게이트로 짝수 패리티 생성기를 구성할 때 필요한 논리 게이트를 쓸 것.
○ 짝수 패리티 결과를 Y라고 할 때 홀수 패리티를 구하기 위한 논리식을 쓸 것.

6. (가)는 프로세스의 상태 다이어그램이고, (나)는 리눅스 시스템에서 자식 프로세스를 생성하는 C 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)



(나)

라인번호	C 프로그램
1	`int main(void) {`
2	`  int status;`
3	
4	`  if (fork() == 0)`
5	`  printf("자식 프로세스.\n");`
6	`  else {`
7	`  printf("부모 프로세스.\n");`
8	
9	`  }`
10	`  return 0;`
11	`}`

<조 건>

○ 프로세스 상태는 (가)의 노드 명칭들로 한정한다.
○ (나)에서 fork()를 호출한 프로세스는 부모 프로세스, fork()를 통해 생성된 프로세스는 자식 프로세스라 한다.
○ 고아(orphan) 프로세스란 자식 프로세스보다 부모 프로세스가 먼저 종료된 상태에서 그 자식 프로세스를 의미한다.
○ 좀비(zombie) 프로세스란 자식 프로세스가 종료되었지만, 해당 종료 상태를 그 부모 프로세스가 확인하지 않는 상태의 자식 프로세스를 의미한다.
○ (나) 실행 시 오류는 발생하지 않으며, 고아 프로세스는 다른 프로세스(init 등)의 자식 프로세스가 될 수 없다.

<작성 방법>

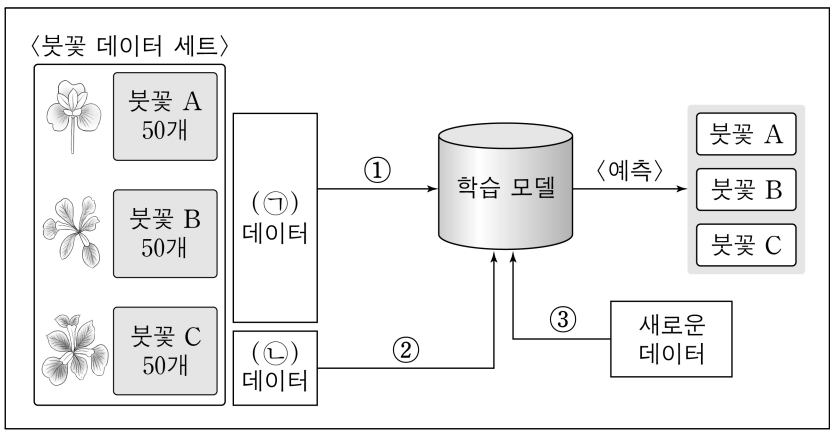
○ (나) 실행에 따른 고아 프로세스 발생 시, 부모 프로세스가 실행 완료한 코드의 라인 번호와 프로세스 상태, 그리고 자식 프로세스가 실행 완료한 코드의 라인 번호와 프로세스 상태를 순서대로 쓸 것.
○ (나) 실행에 따른 좀비 프로세스 발생 시, 부모 프로세스가 실행 완료한 코드의 라인 번호와 프로세스 상태, 그리고 자식 프로세스가 실행 완료한 코드의 라인 번호와 프로세스 상태를 순서대로 쓸 것. (단, (나)의 8번째 라인에 `wait(&status);` 코드를 추가하여 고려한다.)

7. (가)는 2022 개정 고등학교 정보과 교육과정(교육부 고시 제2022-33호)에 따른 인공지능 기초 과목에서 '인공지능과 학습' 영역의 수업에 대한 정보 교사와 예비 교사의 대화 내용이다. (나)는 '인공지능과 학습' 영역의 수업 자료의 일부이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

정보 교사 : 다음 차시 수업에는 어떤 내용을 준비하고 있나요?
예비 교사 : 지난 시간에는 문제 해결에 적합한 기계학습의 유형과 알고리즘에 대해 배웠습니다. 다음 시간에는 학생들이 (㉠) 데이터와 (㉡) 데이터를 이용해 기계학습 모델을 구현하는 수업을 할 예정입니다. 이것은 [12인기02-04] 성취기준에 해당합니다.
정보 교사 : 어떤 수업 교구와 활동을 고려하고 있나요?
예비 교사 : 네, 실습은 파이썬과 붓꽃 데이터 세트를 활용하여 (나)의 ①~③ 순서로 진행됩니다. 먼저, 데이터 세트를 (㉠) 데이터와 (㉡) 데이터로 나눕니다. 붓꽃 데이터에는 붓꽃 종류를 나타내는 레이블과 꽃받침, 꽃잎 정보가 들어 있습니다. 이 데이터를 이용하여 ㉢k-최근접이웃(k-NN) 알고리즘으로 기계학습 모델을 생성하고, 새로운 데이터를 입력하면 붓꽃의 종류를 예측할 수 있습니다.
정보 교사 : 그러면 기계학습 모델을 구현하기 위해 파이썬 프로그래밍 실습이 필요하겠네요. 계획하고 있는 교수·학습 방법이 있나요?
예비 교사 : 네, (㉣)을/를 생각하고 있습니다. ㉤프로그래밍 역량이 높은 학생과 낮은 학생이 짝을 지어 학습하는 형태입니다. 학습자가 다른 학습자를 돕고 가르치면서 학습하는 방법으로 학습자들이 서로 결과를 공유할 뿐만 아니라 그 결과를 얻기 위한 과정도 함께 진행하는 협력적 상황을 이룹니다.
정보 교사 : 좋은 수업이 될 것 같습니다.

(나)

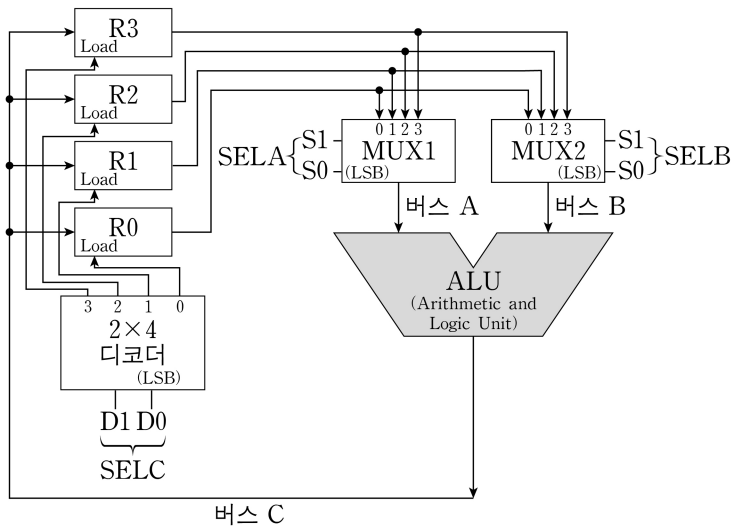


<작성 방법>

○ (가), (나)의 괄호 안의 ㉠, ㉡에 해당하는 용어를 쓸 것.
○ 2022 개정 고등학교 정보과 교육과정에 따른 인공지능 기초 과목의 '인공지능과 학습' 영역의 '성취기준 해설'에 근거하여 밑줄 친 ㉢에 해당하는 기계학습의 유형을 쓸 것.
○ (가)의 밑줄 친 ㉤에 해당하는 교수·학습 방법을 괄호 안의 ㉣에 쓸 것.

8. (가)는 4개의 레지스터를 가진 가상 컴퓨터의 버스 구조를 나타낸 것이고, (나)는 ADD 명령어 형식과 ADD 연산을 수행한 예이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

(가)



(나)

	4bits	4bits	4bits
ADD	Reg1	Reg2	Reg3

C_4 C_3 C_2 C_1 C_0

0 0 1 1 Reg1
\+ 1 0 0 0 Reg2
1 0 1 1 Reg3

<조 건>

- 멀티플렉서(MUX1, MUX2)는 선택 라인(SELA, SELB)을 이용하여 4개의 레지스터(R0~R3) 중 1개를 선택한다.
- 디코더는 선택 라인(SELC)을 이용하여 목적지 레지스터를 선택한다.
- (LSB)는 최하위 비트를 의미한다.
- 레지스터의 크기는 4비트이고, 최상위 비트는 부호 비트이다.
- 레지스터 내에 저장되는 정수는 2의 보수로 저장된다.
- (나)의 ADD 명령어는 Reg3 ← Reg1 + Reg2를 수행하는 3-주소 필드 명령어이다.
- Reg1은 MUX1, Reg2는 MUX2, Reg3은 디코더에서 선택된 레지스터 번호이다.
- C_0~C_4는 캐리이다.
- ALU에서 연산을 수행한 후 오버플로가 발생하면 상태 비트들 중 오버플로 비트(V)를 1로 세트시킨다.
- 레지스터 R1과 R2에는 각각 $(0110)_2$, $(0100)_2$ 값이 저장되어 있다.

<작성 방법>

- R3 ← R1 + R2를 수행할 때 MUX1과 디코더가 해당 레지스터를 선택할 수 있도록 SELA와 SELC의 값을 순서대로 쓸 것.
- R3 ← R1 + R2를 수행한 후 R3에 저장되는 값을 이진수로 쓸 것.
- R3 ← R1 + R2를 수행하면 오버플로가 발생한다. 오버플로 비트를 1로 세트하기 위한 논리식을 C_4, C_3을 이용하여 쓸 것.

9. (가)는 2022 개정 고등학교 정보과 교육과정(교육부 고시 제2022-33호) '내용 체계'의 일부이고, (나)는 '성취기준 적용 시 고려사항'이다. (다)는 이와 관련하여 교사가 준비한 수업 자료이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

핵심 아이디어	• 데이터의 (㉠)와/과 (㉡)은/는 데이터를 효율적으로 관리하고 보호하는 데 도움을 준다.
구분 / 범주	내용 요소
지식 · 이해	• 디지털 데이터 (㉠)와/과 (㉡)
과정 · 기능	… (생략) …
가치 · 태도	… (생략) …

(나)

○ (㉠)(이)나 (㉡)이/가 실제로 구현되는 과정을 프로그래밍 과정에서 확인할 수 있도록 '알고리즘과 프로그래밍' 영역과 연계하여 교수 · 학습 과정을 설계하도록 한다.

(다)

실생활 사례 (도입)	• (㉠) : 소리.mp3, 그림.jpg, 영상.mp4 • (㉡) : 보안 메일, 랜섬웨어
(㉠) 원리의 이해	• 주어진 문자열을 '문자와 그 문자가 연속된 개수'로 표현하면 문자열의 길이를 줄일 수 있음을 이해한다. (예: DDDDD는 D5로 표현된다.)
(㉠) 알고리즘 예제	① 문자열 str1 입력받기; ② 문자 ch ← str1[0]; count ← 0; i ← 0; ③ **while** (i <= strlen(str1)) **do** ④ **if** ch == str1[i] **then** { count++; i++; } **else** { 문자열 str2에 ch와 count를 추가하기; count ← 0; ch ← str1[i]; } ⑤ 문자열 str2를 출력하고, str2의 길이를 출력하기;
(㉡) 원리의 이해	… (생략) …
(㉡) 알고리즘 예제	… (생략) …
	… (하략) …

<작성 방법>

- 괄호 안의 ㉠, ㉡에 해당하는 용어를 2022 개정 고등학교 정보과 교육과정에 근거하여 순서대로 쓸 것.
- (다)의 ㉠ 알고리즘 예제의 ①에서, 입력된 문자열(str1)이 "AAABBBBCC"일 경우 ⑤의 실행 결과를 쓸 것.

10. 다음은 최소 힙(heap)을 이용하여 내림차순으로 정렬을 하는 C 프로그램의 일부이다. <조건>을 고려하여 <작성 방법>에 따라 서술하시오. [4점]

```
#include <stdio.h>

void makeHeap(int list[], int n, int root) {
   int child, key;
   int temp;
   key = list[root];
   child = 2 * root;
   while (child <= n) {
      if (          ㉠          )
         child++;
      if (key < list[child])
         break;
      else {
         list[child/2] = list[child];
         child *= 2;
      }
   }
   list[child/2] = key;
}

void printNodeIndex(int list[], int key) {
   for (int k = 1; k < 8; k++)
      if (list[k] == key)
         ⓑ printf("%d\n", k);
}

void heapSort(int list[], int n) {
   int temp;

   for (int i = n / 2; i > 0; i--)
      makeHeap(list, n, i) ;
ⓐ printf("%d\n", list[4]);

   for (int i = n - 1; i > 0; i--) {
      temp = list[1];
      list[1] = list[i + 1];
      list[i + 1] = temp;
          ㉡          ;
      if (temp == 20) printNodeIndex(list, 40);
   }
}

int main(void) {
   int list[8] = {0, 10, 30, 20, 40, 1, 50, 70};

   heapSort(list, 7);
   return 0;
}
```

<조 건>

○ 힙 정렬은 완전이진트리 형태의 힙 자료 구조를 이용하여 정렬을 수행한다.

○ 최소 힙은 다음 성질을 만족한다.
- 트리는 완전이진트리이어야 한다.
- 각 노드의 키 값은 자식 노드보다 작거나 같다.
- 최소 힙의 루트는 그 트리에서 가장 작은 키 값을 가진다.

○ 힙을 구현하기 위해 배열을 사용한다. 루트 노드의 위치는 1이고, 배열의 0번째 위치는 사용하지 않는다.

○ n개의 노드를 가진 완전이진트리의 인덱스 i를 갖는 노드의 왼쪽 자식 노드의 인덱스는 $2i$이고, $2i+1 \leq n$이면 오른쪽 자식 노드의 인덱스는 $2i+1$이다.

○ 함수 printNodeIndex(list, key)는 배열 list에서 key의 값을 저장하고 있는 인덱스를 출력한다.

<작성 방법>

○ 내림차순으로 정렬하기 위하여 괄호 안의 ㉠과 ㉡에 들어갈 코드를 순서대로 쓸 것.

○ 밑줄 친 ⓐ, ⓑ의 출력 결과를 쓸 것.

11. (가)는 교사가 학생에게 제시한 문제이고, (나)는 제시된 문제를 학생이 구현한 C 프로그램의 일부이다. (다)는 교사가 학생에게 코딩 규칙을 지도하기 위해서 작성한 C 프로그램이다. <작성 방법>에 따라 서술하시오. [4점]

(가)

[문제] switch-case 문을 이용하여 다음의 배터리 충전량에 따라 색상을 출력하는 프로그램을 구현하시오. (단, 배터리 충전량은 정수이며, 0%~100%이다.)

배터리 충전량(%)	출력 색상
80 이상 ~ 100 이하	Blue
60 이상 ~ 80 미만	Green
40 이상 ~ 60 미만	Yellow
0 이상 ~ 40 미만	Red

(나)

학생이 작성한 프로그램 코드

```
#include <stdio.h>

int main(void) {

   int cRate;
   scanf("%d", &cRate);

   switch( ㉠ )
   {
      case 5:
      case 4:
         printf("Blue\n");
      case 3:
         printf("Green\n");
      case 2:
         printf("Yellow\n");
      case 1:
         printf("Red\n");
   }
   return 0;
}
```

(다)

지도할 프로그램 코드

```
#include <stdio.h>

int main(void) {

   int cRate;
   printf("충전량 입력(0%~100% 범위): ");
   scanf("%d", &cRate);

   switch( ㉠ )
   {
      case 5:
         // 5일 때에도 4와 동일하게 적용
      case 4:
         printf("Blue\n");
         ㉡ ;
      case 3:
         printf("Green\n");
         ㉡ ;
      case 2:
         printf("Yellow\n");
         ㉡ ;
      ㉢ :
         printf("Red\n");
         ㉡ ;
   }
   return 0;
}
```

<작성 방법>

- 빈칸의 ㉠에 들어갈 코드를 쓸 것. (단, 배터리 충전량의 변수명은 cRate를 사용한다.)
- cRate가 75일 때, (나)의 실행 결과를 쓸 것.
- 빈칸 ㉡, ㉢에 들어갈 코드를 순서대로 쓸 것.

<수고하셨습니다.>