

15회 E-PPER 프로그래밍 문제

객관식 2 + 주관식 8

제한시간 : 140분, 총 100점

출제자 : 최윤정, 천지영

난이도	상	중	하	총
기출	1	1	1	3
신규	2	1	2	5
	3	2	3	8

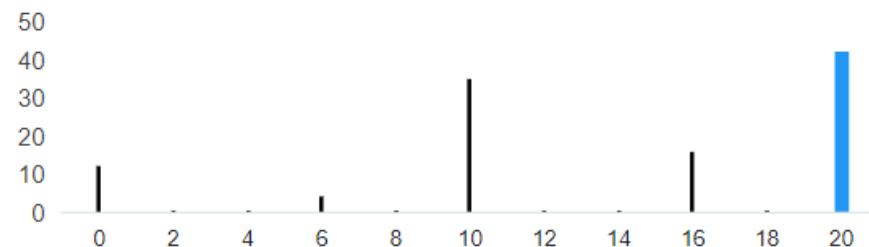
객관식 /주관식 점수

- 객관식 각 10점 , 주관식 각 10점 : 80점

	객관식	주관식							
	1,2번	3번	4번*	5번	6번*	7번	8번	9번*	10번
난이도	하	하	하	하	중	중	상	상	상
풀이학생(명)	96	102	86	57	67	17	17	35	14
비율	88%	94%	80%	53%	62%	16%	16%	32%	13%
평균1	13.4	9.1	7.2	4.4	4.5	0.3	0.3	2.3	0.2
평균2 부분점수 부여	13.4	10.0	8.0	5.4	5.8	0.7	0.5	5.3	0.9

1. 객관식

지원자 점수 분포(총 108명)



- [E-PPER-1] 1차배열의 동적할당 malloc()

지문 부분 점수 반영

C언어에서, `double num[10];` 과 같은 크기로 동적할당을 하려고 할 때 올바른 코드를 모두 고르세요

-32bit 컴파일환경, 부분점수 있음

-부분점수 식 : ((맞춘 개수 - 틀린 개수) ÷ 전체 정답 개수) . 최하 0점.

보기

- ① (✓) `double *num = (double *)malloc(80);`
- ② `int *num = (int *)malloc(sizeof(int) * 10);`
- ③ `double *num = (double *)malloc(sizeof(double *) * 10);`
- ④ (✓) `double *num = (double *)malloc(sizeof(double) * 10);`

-출제의도 : 언어적 기본사항 - C프실,자료구조 및 Solution() 함수기초

2. 객관식

- [E-PPER-2] 함수의 프로토타입

10

지문 부분 점수 반영

C언어에서, 정수형 1차배열 `num`과 `num`의 크기를 인자로 받고, 문자열(문자형 1차배열)을 리턴하는 의미를 가진 `solution` 함수의 원형 (prototype)으로 적합한 것을 모두 고르세요.

- 32bit 컴파일환경, 부분점수 있음
- size_t는 unsigned int 와 같은 의미.
- 리턴하게 되는 문자열 변수는 solution함수내에서 크기 100으로 동적할당 되었다고 가정
- 부분점수 식 : ((맞춘 개수 - 틀린 개수) ÷ 전체 정답 개수) . 최하 0점.

보기

- ① `int *solution(int *num, size_t size);`
- ② (✓) `char *solution(int *num, size_t size);`
- ③ `char []solution(int num[], unsigned int size);`
- ④ (✓) `char *solution(int num[], size_t size);`

-출제의도 : 언어적 기본사항 - C프실,자료구조 기초, Solution() 함수기초

3. 재고없는 날

[문제](#) [초기 코드](#) [테스트 케이스](#)

이화쇼핑에서는 노트북이 하루에 1개씩 판매됩니다. 현재 N개의 노트북을 가지고 있고, M일에 한 번씩 도매점으로부터 노트북이 1개씩 입고됩니다. 노트북의 재고가 0이 될 때까지 며칠이 걸리는지 return 하는 solution 함수를 완성해 주세요. 단, 재고가 0이 될 때, 도매점으로부터 입고되는 날이면 재고가 다시 1개 늘어납니다.

[제한사항]

- N : 100 이하의 자연수
- M : 2보다 크거나 같고 100보다 작거나 같은 자연수

Solution.java

```
1 class Solution {
2     public int solution(int n, int m) {
3         int answer = 0;
4         return answer;
5     }
6 }
```

[문제](#) [초기 코드](#) [테스트 케이스](#)

초기 코드

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  int solution(int n, int m) {
6      int answer = 0;
7      return answer;
8  }
```

4. 100만들기 (2018-JAVA기출문제)

[문제](#) [초기 코드](#) [테스트 케이스](#)

주어진 9개의 숫자 중에서 7개의 숫자의 합이 100이 되는 수를 찾아내야 합니다.

9개의 숫자들은 모두 1이상 99 이하이며 빈칸으로 구분되어 주어지며 이들 중 같은 숫자가 반복되는 경우는 없으며, 답은 항상 유일합니다.

[제한사항]

- 정수의 배열 numbers의 크기는 9입니다.
- 정수 배열 numbers에 저장된 각 수들은 1이상 99이하이며 같은 크기의 숫자가 반복되는 경우는 없습니다.
- 100을 만들 수 있는 7개의 숫자들을 answer에 저장합니다.
- c언어를 배열을 리턴할 때에 answer는 동적할당을 사용해야 하며, 크기를 고려하여 malloc() 문장을 작성합니다.

[입출력 예]

numbers	answer
[1 2 3 4 5 6 7 8 79]	[1 2 3 4 5 6 79]

입출력 예 설명

입출력 예#1 : 9개의 수가 저장된 정수배열 numbers는 오름차순으로 정렬되어 있으며, 이 중 7과 8을 제외하면 100을 만들 수 있습니다.

answer에 저장되는 수는 7개이며 numbers에 있던 7과 8이 제거되었습니다.

Solution.java

```
1 class Solution {
2     public int[] solution(int[] numbers) {
3         int[] answer = {};
4         return answer;
5     }
6 }
```

초기 코드

solution.c

```
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include <stdlib.h>
4
5 // 정수 배열 numbers의 길이는 9 입니다.
6 int* solution(int numbers[]) {
7     // return 값은 malloc 등 동적 할당을 사용해주세요.
8     int* answer = (int*)malloc(1);
9     return answer;
10 }
```

5. 문자열 압축

문제 초기 코드 테스트 케이스

문자열에 등장하는 문자의 종류가 한정되어 있는 경우에는 연달아 등장하는 문자의 출현 횟수를 기록함으로써 문자열의 저장에 소요되는 기억 공간을 줄일 수 있습니다. 특히, 등장하는 문자들이 두 가지 뿐일 때에는 두 가지의 문자가 번갈아서 나타난다고 볼 수 있기 때문에 기억 공간을 더욱 줄일 수 있습니다.

“0”과 “1”로만 이루어진 비트열의 경우에는 같은 비트가 연속해서 등장하는 횟수들을 기록하여 저장 공간을 줄일 수 있습니다. 예를 들어, 첫 문자가 “0”으로 시작하는 “00011110”의 경우, “CDA”라고 표기하여 저장 공간을 줄일 수 있는데, “C”는 “0”이 3개 연속으로 등장하였다는 의미이고, “D”는 “1”이 4개, “A”는 “0”이 1개가 연속하여 등장했다는 의미입니다.

만약, 비트열이 “0”이 아니라 “1”로 시작하는 경우에는 저장 공간의 제일 앞에 “1”을 붙여서 혼돈을 방지하도록 합니다. 예를 들어, “110100”의 경우에는 “1BAAB”로 표시하면 됩니다. 주어진 문자열이 “111100100011” 일 때 출력할 문자열은 “1DBACB” 입니다. 이 문자열 압축 알고리즘을 구현하는 함수 solution() 을 완성하세요.

[제한사항]

- 동일한 문자가 1번 등장하는 경우에는 “A”, 2번 등장하는 경우에는 “B”, ..., 동일한 문자가 26회 연달아 등장하는 경우에는 “Z”로 대응되며, 동일한 문자가 27회 이상 연달아 등장하는 경우는 입력으로 주어지지 않습니다.
- 입력으로 주어지는 문자열 src 의 길이는 1 이상 10,000 이하이며, 이 문자열에는 “0” 과 “1” 외의 문자는 등장하지 않음을 가정합니다.
- 출력되는 문자열에는 대문자가 사용됩니다.

[입출력 예]

src	answer
"000"	"C"
"11111"	"1E"
"00011110"	"CDA"
"111100100011"	"1DBACB"

- 입출력 예 설명 #1

"0"이 연이어 세 번 등장하므로 출력할 문자열 answer 는 "C"입니다.

- 입출력 예 설명 #2

"1" 이 다섯 번 등장하므로 출력할 문자열 answer 는 "1E" 입니다.

- 입출력 예 설명 #3

"0"으로 시작하는 문자열이며 "0"은 연이어 세 번 등장하므로 출력할 문자열 answer의 첫 글자는 3 을 나타내는 "C" 입니다. 그 뒤에는 "1" 이 네 번, "0" 이 한 번 등장하였으므로 "DA" 를 덧붙입니다. 출력할 문자열 answer 의 전체 내용은 "CDA"입니다.

- 입출력 예 #4

"1"로 시작하는 문자열이며, "1"은 연이어 네 번 등장하므로 출력할 문자열 answer는 "1D"로 시작합니다. 그 뒤에는 "0"이 두 번, "1"이 한 번, "0"이 세 번, "1"이 두 번 등장하므로 출력할 문자열 answer는 규칙에 의해 "1DBACB"가 됩니다.

초기 코드	Solution.java
solution.c <pre> 1 #include <stdio.h> 2 #include <stdbool.h> 3 #include <stdlib.h> 4 #include <string.h> 5 6 // 파라미터로 주어지는 문자열은 const로 주어집니다. 변경하려면 문자열을 복사해서 사용하세요. 7 char* solution(const char* s) { 8 // return 값은 malloc 등 동적 할당을 사용해주세요. 할당 길이는 상황에 맞게 변경해주세요. 9 char* answer = (char*)malloc(1); 10 return answer; 11 }</pre>	<pre> 1 class Solution { 2 public int solution(int m, String input) { 3 int answer = 0; 4 return answer; 5 } 6 }</pre>

6. 후위표기법 (2019-C 기출문제)

문제 초기 코드 테스트 케이스

A + B와 같이 피연산자 A와 B 중간에 연산자 +가 위치하는 계산방식을 중위표기법(Infix Notation)이라 부릅니다. 후위표기법(Postfix Notation)은 A B +와 같이 피연산자 A와 B의 뒤에 연산자 +가 위치한 표기법을 말합니다. 수식 (5 + 8) * 2를 후위표기법으로 바꾸면 5 8 + 2 *로 표현됩니다.

후위표기법으로 주어진 수식을 올바르게 계산한 후, 결과를 정수로 리턴하는 함수 solution()을 완성하세요.

[제한사항]

- 첫번째 인자로 전달되는 값 m은 총 입력되는 연산자와 피연산자의 개수의 합으로 $3 \leq m \leq 11$ 입니다.
- 그 다음으로, 하나의 빈칸으로 구분된 연산자와 피연산자 m개가 문자열의 형태로 주어지므로 이 문자열 input의 길이는 5 이상 21 이하입니다. (m + 빈칸개수 m-1)
- 피연산자 x는 $0 \leq x \leq 9$ 의 범위를 갖는 정수이며, 연산자는 사칙연산인 * / + - 입니다.
- 0으로 나누는 경우는 없습니다.

[입출력 예]

****주의:** 입력으로 사용되는 정수와 문자열은 solution함수의 인자로 전달됩니다.

m	input	answer
3	2 3 +	5
3	4 1 -	3
3	9 4 /	2
7	2 3 + 2 * 2 /	5

- 입출력 예 설명 #1 피연산자와 연산자의 개수는 3이며, $(2 + 3) = 5$ 이므로 answer는 5입니다.
- 입출력 예 설명 #2 피연산자와 연산자의 개수는 3이며, $(4 - 1) = 3$ 이므로 answer는 3입니다.
- 입출력 예 설명 #3 피연산자와 연산자의 개수는 3이며, $(9 / 4) = 2$ 이므로 answer는 2입니다.
- 입출력 예 설명 #4 피연산자와 연산자의 개수는 7이며, $(2 + 3) * 2 / 2 = 5$ 이므로 answer는 5입니다.

초기 코드

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  // 파라미터로 주어지는 문자열은 const로 주어집니다.
6  int solution(int m, const char* input) {
7      int answer = 0;
8      return answer;
9  }
```

7. 도서관 예약

문제 초기 코드 테스트 케이스

공학도서관에는 인기가 많은 두 좌석이 있습니다. 학생들은 E-PPER를 보기 전 날, 이 두 좌석 중 하나를 사용하기 위해 사용을 원하는 희망 시작시간과 희망종료시간을 적은 신청서를 도서관 사서에게 제출합니다. 그런데 이 두 좌석은 인기가 너무 많은 나머지, 신청하는 학생들이 너무 많아서 앉지 못하는 학생들이 생길 수 있습니다. 이런 문제들을 최소화하기 위해 도서관 사서는 최대한 많은 학생들이 좌석을 이용할 수 있도록 예약해주려고 합니다. 두 좌석은 똑같이 인기가 많아서 학생들은 둘 중 어디에 앉아도 불만이 없으며, 당연히 한 자리에는 한번에 한 명씩만 앉을 수 있습니다. 이화인은 매우 썩아서 종료시간과 시작시간이 같은 두 학생이 있다면 문제 없이 교대해서 앉을 수 있습니다. 이 사서는 최대 몇 명의 학생에게 좌석을 예약해줄 수 있을까요? 최대한 예약할 수 있는 학생의 수를 return하도록 `solution 함수`를 완성해보세요.

[입력]

- 학생들의 희망시작시간이 저장된 배열 `s`와 희망종료시간이 저장된 배열 `e`가 주어집니다. 여기서, 배열 `s`의 `i`번째 값에는 `i`번째 학생의 희망시작시간이, 배열 `e`의 `i`번째 값에는 `i`번째 학생의 희망종료시간이 저장되어 있습니다.
- 희망시작시간과 희망종료시간의 범위는 0초부터 86399초 사이이며, 각 학생별로 희망시작시간은 희망종료시간보다 항상 이른 시간입니다.

[출력]

최대로 예약할 수 있는 학생의 수를 리턴합니다.

[제한 사항]

- 공학도서관 자리를 희망하는 학생은 0명 이상 1,000명 이하입니다.
- 한 사람은 최대 한 개의 신청서만 낼 수 있습니다.
- 희망시작시간과 희망종료시간은 0초 이상 86399초 이하의 정수입니다.

[입출력 예]

- 입력 예

s[]	e[]
[0, 6, 3, 1, 1, 2]	[3, 7, 10, 5, 9, 8]

- 출력 예

4

- 입출력 예 설명:

첫 번째 학생의 희망시작시간은 0초, 희망종료시간은 3초,
두 번째 학생의 희망시작시간은 6초, 희망종료시간은 7초,
세 번째 학생의 희망시작시간은 3초, 희망종료시간은 10초,
네 번째 학생의 희망시작시간은 1초, 희망종료시간은 5초,
다섯 번째 학생의 희망시작시간은 1초, 희망종료시간은 9초,
여섯 번째 학생의 희망시작시간은 2초, 희망종료시간은 8초일 때,
최대로 예약할 수 있는 학생의 수는 4명입니다.
한 좌석에 1번, 3번 학생이 앉고, 다른 한 좌석에 4번, 2번 학생이 앉는 경우입니다.

초기 코드

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  // s_len은 배열 s의 길이입니다.
6  // e_len은 배열 e의 길이입니다.
7  int solution(int s[], size_t s_len, int e[], size_t e_len) {
8      int answer = 0;
9      return answer;
10 }
```

초기 코드

Solution.java

```
1  class Solution {
2      public int solution(int[] s, int[] e) {
3          int answer = 0;
4          return answer;
5      }
6  }
```

8. e-Queen

[문제](#) [초기 코드](#) [테스트 케이스](#)

n-Queen은 너무나 유명한 문제이다. 혹시 잘 모르는 학생들을 위해 체스판에서 Queen이 어떻게 움직이는지부터 설명해보면, $n \times n$ 사이즈의 체스판에서 Queen은 현재 위치를 기준으로 가로, 세로, 대각선 중 한 방향을 골라 원하는 만큼 이동하면서 만나는 것을 공격할 수 있다. n-Queen은 $n \times n$ 사이즈의 체스판에 n개의 Queen들이 서로를 공격할 수 없도록 배치하는 경우가 총 몇 가지인지를 계산하는 문제이다.

예를 들어, $n = 4$ 인 경우는 아래와 같은 두 가지의 배치가 가능하다. 여기서 Q가 Queen이 있는 칸이고, X는 아무것도 없는 칸이다.

X Q X X
X X X Q
Q X X X
X X Q X

X X Q X
Q X X X
X X X Q
X Q X X

e-Queen은 지금까지 설명한 n-Queen을 변형한 문제이다. n-Queen과의 유일한 차이는 Queen을 절대로 놓을 수 없는 k개의 위치 정보가 추가로 주어진다. 제일 왼쪽 상단 자리를 (1, 1)이라고 하고, 제일 오른쪽 하단 자리를 (n, n)이라고 하자. 만약 위의 예에서 (1, 2)와 (1, 3)이 추가로 주어졌다면, 이 위치에는 Queen을 놓을 수 없으므로 n = 4인 경우 Queen이 서로를 공격할 수 없는 위치에 놓여있도록 구성할 수 없게 된다. 즉, 경우의 수가 0가지이다.

n과 k개의 위치 정보가 주어졌을 때, Queen이 서로를 공격할 수 없도록 배치하는 경우가 총 몇 가지인지를 return하도록 `solution` 함수를 완성해보세요.

[입력]

- n, k, 그리고 k개의 (x, y) 위치정보가 주어진다.
- n은 14 이하의 자연수이고, k는 n 이하의 자연수이다.
- (x, y) 위치 정보는 각각 $1 \leq x, y \leq n$ 을 만족하는 정수이다. 즉, 체스판은 왼쪽 상단인 (1, 1)부터 오른쪽 하단인 (n, n)까지의 위치 정보가 존재한다.
- 함수의 입력으로는 편의상 x 좌표의 값들을 모은 배열 X와, x 좌표에 대응되는 y 좌표의 값들을 모은 배열 Y가 주어진다. 즉, 배열 x의 i 번째 값 x[i]와 배열 y의 i 번째 값 y[i]는 (x[i], y[i])에 Queen을 놓을 수 없음을 의미한다.
- k개의 위치 정보는 서로 모두 다르다.

[출력]

Queen을 절대 놓을 수 없는 k개의 위치 정보가 주어졌을 때, Queen이 서로를 공격할 수 없게 배치하는 경우가 총 몇 가지인지를 리턴한다.

[부분점수 Tip] - 완전히 해결할 수 없는 경우 참고하세요.

- #1: $k = 0$ 인 경우에 대해서만 해결하면 2점을 받을 수 있다.
- #2: $k \leq 1$ 인 경우에 대해서만 해결하면 5점을 받을 수 있다.
- 당연히 #2는 #1을 모두 포함한다.

[입출력 예]

- 입력 예

n	k	X[]	Y[]
4	2	[1,1]	[2,3]

- 출력 예

0

- 입출력 예 설명

4 X 4 사이즈의 체스판에서 Queen을 절대 놓을 수 없는 2개의 위치 정보인 (1, 2)와 (1, 3)이 주어졌을 때, Queen이 서로를 공격할 수 없게 배치할 수 있는 경우는 없다.

초기 코드

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  // x_len은 배열 x의 길이입니다.
6  // y_len은 배열 y의 길이입니다.
7  int solution(int n, int k, int x[], size_t x_len, int y[], size_t y_len) {
8      int answer = 0;
9      return answer;
10 }
```

Solution.java

```
1  class Solution {
2      public int solution(int n, int k, int[] x, int[] y) {
3          int answer = 0;
4          return answer;
5      }
6  }
```

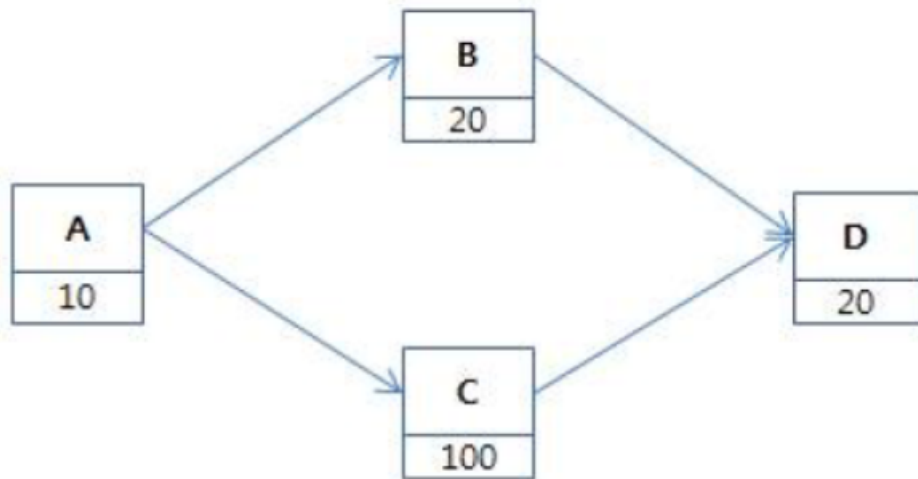

9. N개의 작업공정

(2018-JAVA기출문제)

n개의 작업 공정이 있습니다. 공정마다 소요되는 시간이 있고, 각 공정들 끼리 선후 관계가 존재할 때는 반드시 선행 공정이 끝나야만 다음 공정으로 넘어갈 수 있습니다.

예를 들어 아래 공정을 보면, A공정에 10이 소요되고 난 후, B와 C가 동시에 진행이 됩니다. 이 때 B공정이 끝나는 시점은 30이 되고, C공정은 110에 끝나게 됩니다. 이 때 D공정은, B와 C가 모두 끝나는 시점인 (즉,C가 끝나는 시점) 110에 시작하게 되고, 따라서 D공정은 130에 끝나게 됩니다.

이러한 원칙을 적용하여, 임의로 주어지는 공정에 대해 목표되는 공정까지 소요되는 최소시간을 구하는 프로그램을 작성하세요



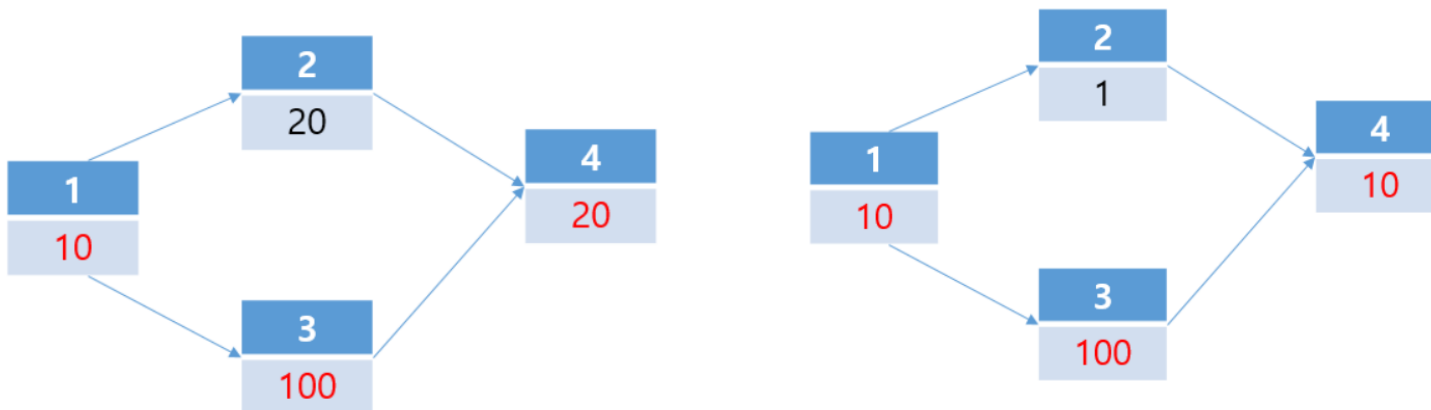
[제한사항]

- 공정수 n 과 n 사이의 관계수는 모두 1이상 100이하의 정수입니다.
- n 개의 각 공정에서 소요되는 시간을 저장하는 배열 N 은 1차정수배열입니다.
- n 개의 공정간의 관계가 저장된 Relation은 2차정수배열입니다.
- 이해를 돕기위해 그림에서는 공정번호를 A,B,C,D로 사용하여 설명하였으나 n 개의 공정번호 역시 정수를 사용합니다.
- 공정간의 관계는 [앞공정번호 뒤공정번호] 순이며, 그림에서 $A \rightarrow B$, $B \rightarrow C$, $A \rightarrow C$ 의 관계가 성립할 때 $[[1,2], [2,3], [1,3]]$ 로 주어져며, 이때 Relation의 행크기는 3, 열크기는 2입니다.
- n (공정수), N 의 크기, Relation의 행크기(공정간의 관계수)는 모두 1이상 100이하의 정수임을 다시 한번 강조합니다.

[입출력 예]

N	Relation	goal	answer
[10, 20, 100, 20]	[[1, 2], [1, 3], [2, 4], [3, 4]]	4	130
[10, 1, 100, 10]	[[1, 2], [1, 3], [2, 4], [3, 4]]	4	120

- 입출력 예 설명 #1 : (그림)



초기 코드

C

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  // N_len은 배열 N의 길이입니다.
6  // Relation_rows는 2차원 배열 Relation의 행 길이, Relation_cols는 2차원 배열 Relation의 열 길이입니다.
7  int solution(int N[], size_t N_len, int** Relation, size_t Relation_rows, size_t Relation_cols, int goal) {
8      int answer = 0;
9      return answer;
10 }
```

Solution.java

```
1  class Solution {
2      public int solution(int[] N, int[][] Relation, int goal) {
3          int answer = 0;
4          return answer;
5      }
6  }
```

10. 랜덤 질문

문제 초기 코드 테스트 케이스

한 교실에 100명의 학생들이 10 X 10 형태로 반듯하게 앉아있다. 이 교실에 있는 모든 학생들은 50점의 기본 점수를 가지고 있다. 교수님은 이 학생들 중 한 명씩 랜덤으로 선택하여 질문을 한다. 그 학생이 정답을 말했을 경우, 그 학생의 가로줄과 세로줄에 있는 학생들이 모두 1점씩 획득한다. 물론 맞춘 학생도 1점을 획득한다. 그런데 그 학생이 만약 오답을 말했을 경우, 그 학생의 가로줄과 세로줄에 있는 모든 학생들이 1점씩 감점을 당한다. 당연히 오답을 말한 학생도 1점 감점을 당한다.

예를 들어, 제일 왼쪽 상단 자리를 (1, 1)이라고 하고, 제일 오른쪽 하단 자리를 (10, 10)이라고 하자. 처음 시작했을 때의 점수는 아래와 같다.

50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50

만약 교수님이 (2, 5)에 앉은 학생에게 질문을 했고, 그 학생이 정답을 맞추었다면 아래와 같은 점수가 된다.

50 50 50 50 51 50 50 50 50 50
51 51 51 51 51 51 51 51 51 51
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50
50 50 50 50 51 50 50 50 50 50

위의 상황에 이어서 이번에는 교수님이 (5, 2)에 앉은 학생에게 질문을 했고, 그 학생이 정답을 맞추지 못했다면 아래와 같은 점수가 된다.

50 49 50 50 51 50 50 50 50 50
51 50 51 51 51 51 51 51 51 51
50 49 50 50 51 50 50 50 50 50
50 49 50 50 51 50 50 50 50 50
49 49 49 49 50 49 49 49 49 49
50 49 50 50 51 50 50 50 50 50
50 49 50 50 51 50 50 50 50 50
50 49 50 50 51 50 50 50 50 50
50 49 50 50 51 50 50 50 50 50
50 49 50 50 51 50 50 50 50 50

[입력]

10 X 10 사이즈의 최종 점수판이 2차원 배열 형태로 주어진다. 각 점수는 0 이상 100 이하의 정수이다.

[출력]

최종 점수판이 주어졌을 때 질문을 맞춘 학생 자리에는 3, 질문을 맞추지 못한 학생 자리에는 1, 질문을 받지 않은 학생 자리에는 2를 저장하여 2차원 배열로 리턴하는 함수를 작성하시오.

[부분점수 Tip] - 완전히 해결할 수 없는 경우 참고하세요.

- #1 : 질문을 받은 학생의 수가 1명 이하인 경우에 대해서만 문제를 해결하면 3점을 받을 수 있다.
- #2 : 질문을 받은 학생이 모두 정답을 말하거나, 질문을 받은 학생이 모두 오답을 말하는 경우에 대해서만 문제를 해결하면 6점을 받을 수 있다. 예를 들어, 질문을 받은 학생이 3명인데 3명 모두 정답을 말한 경우, 질문을 받은 학생이 5명인데 5명 모두 오답을 말한 경우 등이 이 경우에 해당된다.
- 당연히 #2는 #1을 모두 포함한다.

[입출력 예]

입출력 예 설명

(2, 5)에 앉은 학생이 정답을 맞추었고 (5,2)에 앉은 학생이 정답을 맞추지 못한 경우에 대한 점수이므로, (2, 5)에는 3을 그리고 (5,2)에는 1을 출력하고, 나머지 학생들은 질문을 받지 않았으므로 2를 출력한다.

• 입력 예

50	49	50	50	51	50	50	50	50	50
51	50	51	51	51	51	51	51	51	51
50	49	50	50	51	50	50	50	50	50
50	49	50	50	51	50	50	50	50	50
49	49	49	49	50	49	49	49	49	49
50	49	50	50	51	50	50	50	50	50
50	49	50	50	51	50	50	50	50	50
50	49	50	50	51	50	50	50	50	50
50	49	50	50	51	50	50	50	50	50
50	49	50	50	51	50	50	50	50	50

• 출력 예

2	2	2	2	2	2	2	2	2	2
2	2	2	2	3	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	1	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2

solution.c

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  // point_rows는 2차원 배열 point의 행 길이, point_cols는 2차원 배열 point의 열 길이입니다.
6  int** solution(int** point, size_t point_rows, size_t point_cols) {
7      // return 값은 malloc 등 동적 할당을 사용해주세요. 할당 길이는 상황에 맞게 변경해주세요.
8      int** answer = (int**)malloc(sizeof(int*) * point_rows);
9      int i, j;
10     // 적절히 변수를 추가로 선언해주세요.
11
12     for(i = 0 ; i < point_rows ; i++)
13     {
14         answer[i] = (int*) malloc(sizeof(int) * point_cols);
15
16         for(j = 0 ; j < point_cols ; j++)
17             answer[i][j] = 0;
18     }
19     // 10번째 줄에서 변수를 선언하고, 여기 밑으로 코드를 작성해주세요.
20
21     return answer;
22 }
```

초기 코드

Solution.java

```
1  class Solution {
2      public int[][] solution(int[][] point) {
3          int[][] answer = {};
4          return answer;
5      }
6  }
```

기출문제 풀이 제공 : 상3,중3,하3 총 9개



2020년도 제 15회 E-PPER 대비 기출문제 풀이

문제	난이도	권장 풀이 시간
2018년도 JAVA 부문 7번	하	5분
2018년도 JAVA 부문 9번	중	10분
2018년도 JAVA 부문 10번	상	15분
2019년도 C 부문 7번	중	10분
2019년도 C 부문 8번	중	10분
2019년도 C 부문 9번	상	10분
2019년도 JAVA 부문 3번	하	5분
2019년도 JAVA 부문 8번	상	15분
2019년도 JAVA 부문 10번	중	10분



TOGETHER.
TOMORROW.
EWHW

총 영상 길이는 약 90분입니다.

참고: <https://cafe.naver.com/ewhaqueenp>

사이버캠퍼스 : E-Class

<https://cyber.ewha.ac.kr/course/view.php?id=85238>

기출문제풀이-영상/실시간튜터링



제 15회 E-PPER 경진대회 응시전 유의사항 615.4KB



2020년도 15회 E-PPER 응시전 유의사항 안내 08:23



2020년도 15회 E-PPER 경진대회를 위한 기출문제 풀이영상 01:26:43

*영상속의 소리가 작은 구간이 있는데 양해부탁드립니다.

(각 문제에 대한 pdf 및 스크립트도 업로드 될 예정입니다.)

기출문제풀이는 18,19년 문제 중 대표적인 문제들(상,중,하) 9개로 구성되었으며 이 중 3문제는 15회 경진대회 문제로 출제됩니다.

각 문제들마다 여러가지 방법들이 존재하며, 단 하나의 정답코드가 존재하지는 않습니다.

원스탑튜터가 제안하는 풀이방법과 해결과정을 참고하시고 자주 나오는 부분은 잘 익혀두기 바랍니다!



15회 E-PPER 경진대회 기출문제풀이 발표자료

15회 E-PPER 경진대회 기출문제풀이 발표자료입니다.

코드는 별도로 공개되지 않습니다. 영상을 통해 확인해주세요.



15회 E-PPER 경진대회 기출문제풀이 발표 스크립트

스크립트는 영상과 비슷한 내용을 담고 있으나 일부 차이가 있을 수 있습니다.

구름 IDE 등의 추가적인 자료는 이화 퀸즈 카페에서 확인해주세요.

<https://cafe.naver.com/ewhaqueenp>



15회 E-PPER 경진대회-사전준비안내(기출문제풀이+a))

기출문제풀이 동영상/실시간튜터링

기출문제풀이 동영상/실시간튜터링



기출문제에 대한 질문은 2020-원스탑클래스를 이용하세요^^~

<https://cyber.ewha.ac.kr/course/view.php?id=84685>

이화퀸포

이론 및 예제 - C	▼
이론 및 예제 - JAVA	▼
이론 및 예제 - 자료구조	▼
이론 및 예제 - 알고리즘	▼
단계별 문제 - C	▼
단계별 문제 - JAVA	▼
단계별 문제 - 통합	▼
스터디	▼
컴퓨터공학전공	▼
사이버보안전공	▼
학과 동아리	▼
대외 활동	▼



<https://cafe.naver.com/ewhaqueenp/3809>

카페 소식	제목	작성자	작성일	조회
공지사항	공지 15회 E-PPER 응시자의 신분증 및 화상기기 등록 요청	최윤정 m	2020.10.06.	106
이화뉴스 N	공지 15회 응시자 등록 할 때 - 안드로이드 폰은 "크롬" 권장	최윤정 m	2020.09.29.	76
활동 사진	공지 15회 E-PPER 응시자 등록 및 사전모의테스트 시작!!	최윤정 m	2020.09.29.	192
추천 정보	공지 15회 E-PPER 신청자 마감알림 및 일정공지(메일과 동일)	최윤정 m	2020.09.25.	116
공모전	공지 15회 E-PPER 프로그래밍 경진대회-기출문제풀이영상 upload!	최윤정 m	2020.09.25.	87
취업, 인턴 N	공지 15회 E-PPER 온라인시험관련 기본 안내 및 문의내용-답변3(9/22-개인정보동의서(수정) 업로드요청)	최윤정 m	2020.09.22.	135
강연, 교육	공지 15회 E-PPER 온라인시험관련 기본 안내 및 문의내용-답변2(9/14-이미지문제수정)	최윤정 m	2020.09.14.	237
기타	공지 15회 E-PPER 프로그래밍 경진대회 공지 및 안내	최윤정 m	2020.09.14.	420
TIP과 Q&A	공지 15회 E-PPER 온라인시험관련 기본 안내 및 문의내용-답변1(확인된 내용 포함-9/12)	최윤정 m	2020.09.10.	265
C TIP	15회 E-PPER 프로그래밍 경진대회 - 2018 java 기출문제 풀이 (7번)	18정혜민 s	2020.09.26.	58
JAVA TIP	15회 E-PPER 프로그래밍 경진대회 - 2018 java 기출문제 풀이 (9번)	18정혜민 s	2020.09.26.	36
Python TIP	15회 E-PPER 프로그래밍 경진대회 - 2018 java 기출문제 풀이(10번)	18정혜민 s	2020.09.26.	38
기타 TIP	15회 E-PPER 프로그래밍 경진대회 - 2019 java 기출문제 풀이(10번)	18윤하정 s	2020.09.25.	35
건의 게시판	15회 E-PPER 프로그래밍 경진대회 - 2019 java 기출문제 풀이(8번)	18윤하정 s	2020.09.25.	40
E-PPER 경진대회	15회 E-PPER 프로그래밍 경진대회 - 2019 java 기출문제 풀이(3번)	18윤하정 s	2020.09.25.	39
2020 경진대회	15회 E-PPER 프로그래밍 경진대회 - c부문 기출문제 풀이 (9번)	17권지현 s	2020.09.25.	33
2019 경진대회	15회 E-PPER 프로그래밍 경진대회 - c부문 기출문제 풀이 (8번)	17권지현 s	2020.09.25.	36
2018 경진대회	15회 E-PPER 프로그래밍 경진대회 - c부문 기출문제 풀이 (7번)	17권지현 s	2020.09.25.	52
2017 경진대회	15회 E-PPER 프로그래밍 경진대회-기출문제풀이영상 upload!	최윤정 m	2020.09.25.	87
2016 경진대회				
2015 경진대회				
2014 경진대회				
2013 경진대회				
2012 경진대회				

실시간 튜터링 시간표

<https://cyber.ewha.ac.kr/course/view.php?id=84685>



2020-2 소프트웨어학부 컴퓨터공학전공/사이버보안전공 OneStop 튜터링

튜터 시간표(웹페이지에서 확인하세요)



교시 시간	월			화			수			목			금		
언어	C	JAVA	python/ 리눅스	C	JAVA	python/ 리눅스	C	JAVA	python/ 리눅스	C	JAVA	python/ 리눅스	C	JAVA	python/ 리눅스
3교시 11:00~		정혜민		윤가영						권지현			권지현		이소은
4교시 12:30~		정혜민		윤가영			유호민	윤하정		권지현			권지현		
5교시 14:00~	유호민	윤하정			한근영		유호민	윤하정			정혜민				한다현
6교시 15:30~	유호민	윤하정			한근영	한다현			이소은		정혜민				
7교시 17:00~	윤가영													한근영	
19:00~ 20:00	윤가영									튜터 회의				한근영	



공지사항



Ewha QueenP(카페 ...



C



Java



Python



Linux



기타 Q&A