**[ C 언어 5차 1급 ]**

**문제 1.**

계단 n칸을 올라가는 방법의 수를 구하려고 합니다. 계단은 한 번에 1계단, 2계단, 3계단씩 오를 수 있습니다.

예를 들어, 계단 3칸을 오르는 방법은 다음과 같이 4가지가 있습니다.

1. 1계단 + 1계단 + 1계단

2. 1계단 + 2계단

3. 2계단 + 1계단

4. 3계단

계단 수 n이 매개변수로 주어질 때, 계단을 오르는 경우의 수를 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해주세요.

**▣ 매개변수 설명**

계단 수 n이 solution 함수의 매개변수로 주어집니다.

\* n은 3 이상 30 이하인 정수입니다.

**▣ return 값 설명**

계단을 오르는 경우의 수를 return 합니다.

**▣ 입출력 예**

| n | return |

|---|--------|

| 3 | 4 |

| 4 | 7 |

**▣ 입출력 예 설명**

예시 #1

문제에 나온 예와 같습니다.

예시 #2

계단 4칸을 오르는 방법은 다음과 같이 7가지가 있습니다.

1. 1계단 + 1계단 + 1계단 + 1계단

2. 1계단 + 1계단 + 2계단

3. 1계단 + 2계단 + 1계단

4. 2계단 + 1계단 + 1계단

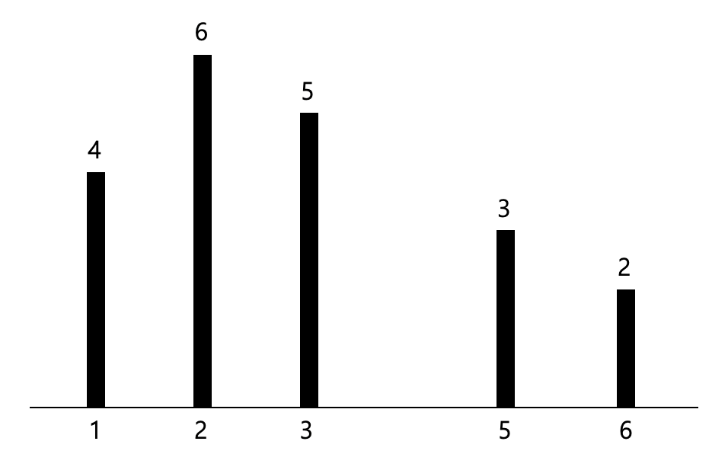
5. 1계단 + 3계단

6. 3계단 + 1계단

7. 2계단 + 2계단

**문제 2.**

아래와 같이 벽이 여러 개 있을 때, 벽 2개를 제외한 나머지 벽을 제거하여 물을 최대한 담으려 합니다. 물은 `두 벽 사이의 거리 x 두 벽 중 낮은 벽의 높이`리터 만큼 담을 수 있으며, 두 벽의 거리는 두 벽의 위치 차이입니다.



모든 벽의 위치와 높이를 담은 2차원 배열 walls과 walls의 길이 walls\_len이 매개변수로 주어질 때, 물을 최대 몇 리터나 담을 수 있는지 return 하도록 solution 함수를 작성했습니다. 그러나, 일부 코드가 잘못되어 코드가 바르게 동작하지 않습니다. 주어진 코드가 모든 입력을 바르게 처리하도록 코드를 수정해주세요. 코드는 \_\*\*한 줄\*\*\_만 수정해야 합니다.

**▣ 매개변수 설명**

모든 벽의 위치와 높이를 담은 2차원 배열 walls과 walls의 길이 walls\_len이 solution 함수의 매개변수로 주어집니다.

\* walls는 길이가 3이상 100이하인 2차원 배열입니다.

\* walls의 각 원소는 [벽의 위치, 벽의 높이] 이고, 벽의 위치를 기준으로 오름차순으로 정렬되어 있습니다.

\* 벽의 위치는 1 이상 100 이하입니다.

\* 벽의 높이는 1 이상 100 이하입니다.

**▣ return 값 설명**

물을 최대 몇 리터나 담을 수 있는지 return 합니다.

**▣ 입출력 예**

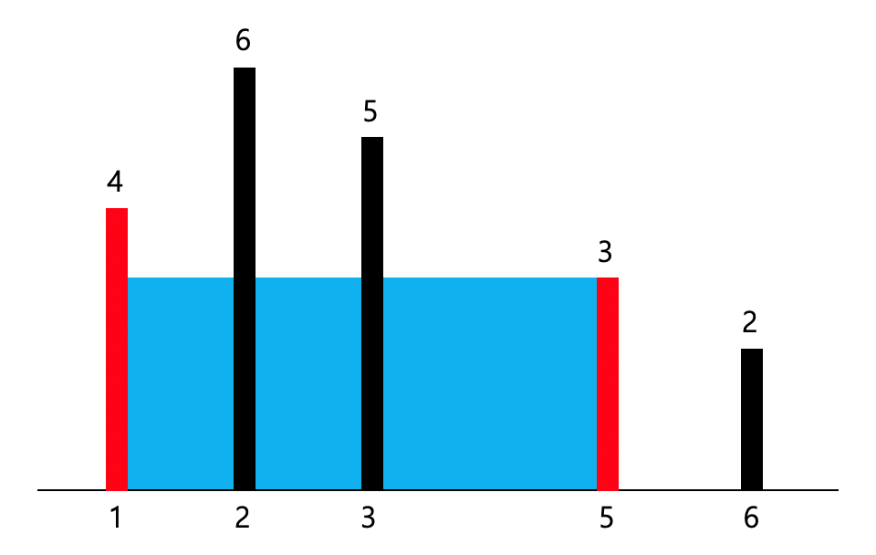
| walls | walls\_len | return |

|------------------------------------------|-----------|--------|

| [[1, 4], [2, 6], [3, 5], [5, 3], [6, 2]] | 5 | 12 |

**▣ 입출력 예 설명**

아래와 같이 벽을 남겼을 때 물을 최대로 담을 수 있습니다. 이 때 최대 12리터를 담을 수 있습니다.



**문제 3.**

서로 다른 정수 n개가 담긴 배열이 있습니다. 이 배열을 앞 `(n+1)/2`개 원소는 증가하고 뒤 `(n+1)/2`개 원소는 감소하도록 정렬하려고 합니다. 이때, 조건을 만족하는 배열이 여럿인 경우 사전순으로 가장 먼저 나오는 배열을 답으로 합니다. 예를 들어, 주어진 배열이 [7, 3, 4, 1, 2, 5, 6]라면 정렬한 이후에는 [1, 2, 3, 7, 6, 5, 4]가 됩니다.

정수 배열 numbers와 numbers의 길이 numbers\_len가 매개변수로 주어질 때, 문제의 조건에 맞게 정렬하여 return 하도록 solution 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되어있기 때문에, 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 \_\*\*한 줄\*\*\_만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

**▣ 매개변수 설명**

정수 배열 numbers와 numbers의 길이 numbers\_len가 solution 함수의 매개변수로 주어집니다.

\* numbers의 각 원소는 -1,000 이상 1,000 이하인 정수이며, 중복되는 수는 없습니다.

\* numbers\_len은 3 이상 10,000 이하인 홀수입니다.

**▣ return 값 설명**

주어진 배열을 앞 `(n+1)/2`개 원소는 증가하고 뒤 `(n+1)/2`개 원소는 감소하도록 정렬하여 return 합니다.

조건을 만족하는 배열이 여럿인 경우 사전순으로 가장 먼저 나오는 배열을 답으로 합니다.

**▣ 입출력 예**

| numbers | numbers\_len | return |

|-----------------------|-------------|-----------------------|

| [7, 3, 4, 1, 2, 5, 6] | 7 | [1, 2, 3, 7, 6, 5, 4] |

**▣ 입출력 예 설명**

문제에 나온 예와 같습니다.

**문제 4.**

주어진 숫자를 각 숫자와 숫자의 개수로 읽으려 합니다. 이때, 값이 큰 숫자를 먼저 읽어야합니다.

예를 들어, 2433은 2 한개 4 한개 3 두개로 이루어져 있기 때문에 "413221"로 읽습니다.

숫자 number가 매개변수로 주어질 때, 숫자를 읽어 문자열로 return 하도록 solution 함수를 작성했습니다. 그러나, 일부 코드가 잘못되어 코드가 바르게 동작하지 않습니다. 주어진 코드가 모든 입력을 바르게 처리하도록 코드를 수정해주세요. 코드는 \_\*\*한 줄\*\*\_만 수정해야 합니다.

**▣ 매개변수 설명**

숫자 number가 solution 함수의 매개변수로 주어집니다.

\* number는 0 이상 2,000,000,000 이하인 정수입니다.

**▣ return 값 설명**

주어진 숫자를 읽어 문자열 형태로 return 합니다.

**▣ 예시**

| number | return |

|--------|----------|

| 2433 | "413221" |

| 662244 | "624222" |

**▣ 예시 설명**

예시 #1

문제에 나온 예와 같습니다.

예시 #2

662244는 6 두개 2 두개 4 두개로 이루어져 있기 때문에 "624222"로 읽습니다.

**문제 5.**

XX게임에선 캐릭터는 자신과 공격력이 같거나 자신보다 공격력이 작은 몬스터에게 이깁니다. 내가 가진 캐릭터가 최대 몬스터 몇 마리를 이길 수 있는지 구하려 합니다. 단, 한 캐릭터는 한 번만 싸울 수 있습니다.

예를 들어, 세 몬스터의 공격력이 각각 [1, 4, 3]이고, 내가 가진 두 캐릭터의 공격력이 각각 [1, 3]이라면 첫 번째 캐릭터는 첫 번째 몬스터와, 두 번째 캐릭터는 세 번째 몬스터와 싸워서 이길 수 있습니다. 따라서 이길 수 있는 몬스터 수는 최대 2마리입니다.

모든 몬스터의 공격력을 담은 배열 enemies, enemies의 길이 enemies\_len, 내가 가진 모든 캐릭터의 공격력을 담은 배열 armies, armies의 길이 armies\_len가 매개변수로 주어질 때, 내 캐릭터로는 최대 몬스터 몇 마리를 이길 수 있는지 return 하도록 solution 함수를 작성해주세요.

**▣ 매개변수 설명**

모든 몬스터의 공격력을 담은 배열 enemies, enemies의 길이 enemies\_len, 내가 가진 모든 캐릭터의 공격력을 담은 배열 armies, armies의 길이 armies\_len가 solution 함수의 매개변수로 주어집니다.

\* 몬스터 수는 1마리 이상, 500마리 이하입니다.

\* 각 몬스터의 공격력은 1 이상 100 이하입니다.

\* 내가 가진 캐릭터 수는 1개 이상 500개 이하입니다.

\* 각 캐릭터의 공격력은 1 이상 100 이하입니다.

**▣ return 값 설명**

내가 가진 캐릭터로는 최대 몬스터 몇 마리를 이길 수 있는지 return 해주세요.

**▣ 입출력 예**

| enemies | enemies\_len | armies | armies\_len | return |

|-----------|-------------|--------------|------------|--------|

| [1, 4, 3] | 3 | [1, 3] | 2 | 2 |

| [1, 1, 1] | 3 | [1, 2, 3, 4] | 4 | 3 |

**▣ 입출력 예 설명**

입출력 예 #1

문제에 나온 예와 같습니다.

입출력 예 #2

캐릭터를 어떻게 보내도 모든 몬스터를 이길 수 있습니다.

**문제 6.**

p 진법으로 표현한 수란, 각 자리를 0부터 p-1의 숫자로만 나타낸 수를 의미합니다. p 진법으로 표현한 자연수 두개를 더한 결과를 q 진법으로 표현하려 합니다.

예를 들어, 3진법 수 112001과 12010을 더한 결과를 8진법으로 나타내면 1005입니다.

solution 함수의 매개변수로 p 진법 자연수를 담은 문자열 s1, s2와 두 수를 나타내는 진법의 기수 p, 두 수의 덧셈 결과를 표현할 진법의 기수 q가 매개변수로 주어집니다. p진법으로 표현된 두 수를 더한 결과를 q 진법으로 나타낸 값을 return 하도록 solution 함수를 완성해주세요.

**▣ 매개변수 설명**

p 진법으로 자연수를 담은 문자열 s1, s2와 두 수를 표현한 진법의 기수 p, 두 수의 덧셈 결과를 표현할 진법의 기수 q가 solution 함수의 매개변수로 주어집니다.

\* p와 q는 2 이상 10 이하인 자연수입니다.

\* s1과 s2의 길이는 1 이상 9 이하입니다.

\* s1과 s2의 원소는 '0', '1', '2', …, ‘p-1’로만 구성됩니다.23333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333521111111111111111111111111111111111111111111111

\* s1이나 s2가 ‘0’인 경우는 주어지지 않습니다.

**▣ return 값 설명**

두 수를 더한 결과를 q 진법으로 나타낸 값을 문자열로 return 하도록 solution 함수를 완성해주세요.

**▣ 예시**

| s1 | s2 | p | q | return |

|----------|---------|---|---|--------|

| "112001" | "12010" | 3 | 8 | "1005" |

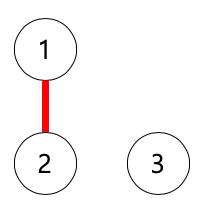
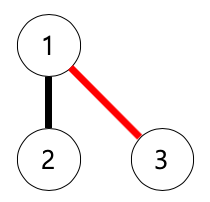
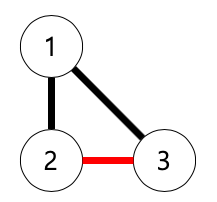
**▣ 예시 설명**

문제에 나온 예와 같습니다.

**문제 7.**

그래프의 노드 수와 노드 연결 순서가 주어질 때, 몇 번째 연결에 사이클이 생기는지 알고 싶습니다. 예를 들어, 노드가 3개이고 노드를 [[1, 2], [1, 3], [2, 3]] 순으로 연결한다면 아래 그림과 같습니다.

\* 1번째 연결 \* 2번째 연결 \* 3번째 연결

따라서 3번째 연결에서 사이클이 생깁니다.

그래프의 노드 수 n, 노드 연결 순서 connections와 connections의 길이 connections\_len이 매개변수로 주어질 때, 몇 번째 연결에 사이클이 생기는지 return 하도록 solution 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해주세요.

**▣ 매개변수 설명**

그래프의 노드 수 n, 노드 연결 순서 connections와 connections의 길이 connections\_len이 solution 함수의 매개변수로 주어집니다.

\* 그래프의 노드 수 n은 3 이상 10 이하입니다.

\* connections은 길이가 3 이상 50 이하인 배열입니다.

\* connections 배열의 각 행은 [a, b]는 a번 노드와 b번 노드를 연결한다는 의미입니다.

\* 항상 사이클이 생기는 경우만 주어집니다.

**▣ return 값 설명**

몇 번째 연결에서 사이클이 생기는지 return 합니다.

**▣ 예제**

| n | connections | connections\_len | return |

|---|--------------------------|-----------------|--------|

| 3 | [[1, 2], [1, 3], [2, 3]] | 3 | 3 |

**▣ 예제 설명**

문제에 나온 예와 같습니다.

**문제 8.**

세 수 a, b, c의 공약수가 몇 개인지 구하려고 합니다. 공약수란, 동시에 모든 정수의 약수인 정수를 뜻합니다. 예를 들어, 세 수 24, 9, 15의 공약수는 1, 3이고, 따라서 양의 공약수는 2개입니다.

세 수의 공약수가 몇 개인지 구하기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 세 수의 최대공약수를 구합니다.

2. 앞서 구한 최대공약수의 약수가 몇 개인지 구합니다.

세 수 a, b, c가 매개변수로 주어질 때, 세 수의 약수가 몇 개인지 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 func\_a, func\_b, func\_c 함수와 매개변수를 알맞게 채워주세요.

**▣ 매개변수 설명**

세 수 a, b, c가 매개변수로 주어집니다.

\* 세 수 a, b, c는 1 이상 1,000 이하인 정수입니다.

**▣ return 값 설명**

세 수의 약수가 몇 개인지 return 합니다.

**▣ 예제**

| a | b | c | return |

|----|---|----|--------|

| 24 | 9 | 15 | 2 |

**▣ 예제 설명**

문제에 나온 예와 같습니다.

**문제 9.**

스택 두개를 이용해 Queue 자료구조를 만들었을 때, Queue 자료 구조의 pop(또는 dequeue) 함수를 구현하려합니다. Queue란 먼저 삽입한 데이터를 먼저 빼내는 자료구조를 뜻합니다. pop 함수를 만들기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 스택2가 비었다면 스택1에 아무것도 남지 않을때까지 스택1을 pop한 값을 스택2에 push 한다.

2. 스택2를 pop한 값을 리턴한다.

배열 stack1과 정수 stack1\_idx, 배열 stack2과 정수 stack2\_idx가 solution 함수의 매개변수로 주어집니다. 이때, stack1\_idx는 stack1을 스택으로 이용했을 때 stack1의 top의 index를 의미하며, stack2\_idx는 stack2를 스택으로 이용했을 때 stack2의 top의 index를 의미합니다.

두 배열을 스택으로 이용해 Queue 자료 구조의 pop 함수를 구현하려합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 func\_a, func\_b, func\_c 함수와 매개변수를 알맞게 채워주세요.

※ 배열 index가 0인 부분을 스택의 bottom으로 생각합니다.

**▣ 매개변수 설명**

배열 stack1과 정수 stack1\_idx, 배열 stack2과 정수 stack2\_idx가 solution 함수의 매개변수로 주어집니다. 이때, stack1\_idx는 stack1을 스택으로 이용했을 때 stack1의 top의 index를 의미하며, stack2\_idx는 stack2를 스택으로 이용했을 때 stack2의 top의 index를 의미합니다.

\* stack1과 stack2는 길이가 0 이상 10 이하입니다.

\* stack1과 stack2의 길이가 모두 0인 경우는 주어지지 않습니다.

\* stack1과 stack2의 원소는 100 이하인 정수입니다.

\* stack1\_idx는 -1 이상 stack1의 길이 미만인 정수입니다.

\* stack2\_idx는 -1 이상 stack2의 길이 미만인 정수입니다.

**▣ return 값 설명**

stack1과 stack2로 구현한 큐를 pop(또는 dequeue)한 값을 return 해주세요.

\* 스택이 넘치는 일은 발생하지 않는다고 가정합니다.

**▣ 예시**

| stack1 | stack1\_idx | stack2 | stack2\_idx | result |

|-----------------------|------------|------------------------|------------|--------|

| [1,2,0,0,0,0,0,0,0,0] | 1 | [3,4,0,0,0,0,0,0,0,0] | 1 | 4 |

| [1,2,3,0,0,0,0,0,0,0] | 2 | [0,0,0,0,0,0,0,0,0,0] | -1 | 1 |

**▣ 예시 설명**

예시 #1

stack2가 비지 않았으므로 stack2를 pop 한 결과인 4를 리턴합니다.

예시 #2

1. stack2가 비었으므로 stack1의 원소를 전부 pop해 stack2로 넣습니다. 이 결과 stack2는 [3,2,1,0,0,0,0,0,0,0]이 됩니다.

2. stack2를 pop한 결과인 1을 리턴합니다.

**문제 10.**

두 수의 덧셈과 뺄셈을 하는 계산기를 구현하기 위해 구조체 Calculator와 함수를 다음과 같이 정의했습니다.

```c

typedef struct Calculator {

int (\*plus)(int, int);

int (\*minus)(int, int);

}Calculator;

```

```c

int plus(int a, int b) {

return a + b;

}

int minus(int a, int b) {

return a - b;

}

```

또한, 다음과 같이 프로그램 구조를 작성했습니다.

1. 덧셈과 뺄셈을 가지고 있는 계산기 구조체를 만듭니다.

2. 계산기 구조체 멤버에 함수의 메모리 주소를 할당합니다.

3. 주어진 두 수의 덧셈과 뺄셈 결과를 순서대로 배열에 담아 return 합니다.

두 수 a, b가 매개변수로 주어질 때, 두 수의 합과 차를 순서대로 배열에 담아 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸을 채워 전체 코드를 완성해주세요.

**▣ 매개변수 설명**

두 수 a, b가 solution 함수의 매개변수로 주어집니다.

\* a, b는 1이상 1,000이하인 정수입니다.

**▣ return 값 설명**

두 수의 합과 차를 배열에 담아 return 합니다.

**▣ 예시**

| a | b | return |

|---|---|---------|

| 3 | 5 | [8, -2] |