CODE FESTIVAL

JEJU CODING BASE CAMP

자바스크립트 JAVA SCRIPT 100제 2

이호준 강민정 김유진 김대현 김혜원 이현창 웨이드 장하림 정윤하 차경림 최원범

JavaScript로 푸는 100개의 코딩 문제
제주 코딩 베이스 캠프 코드 페스티벌로
당신의 실력을 테스트해보세요!

intro

소개글

우리는 우리가 배운 지식을 나누고, 정리하며 성장하기 위해 이 프로젝트를 시작하였습니다. 누구든지 저작권 없이, 값을 지불하지 않고, 그러면서도 양질의 콘텐츠가 필요한 곳에 이 수업자료가 쓰이길 원합니다.

또 이 책을 Notion으로 통으로 오픈하는 이유는, 보다 좋은 내용과 자료가 우리에게 생겼을 때 노션을 업데이트하기 위함입니다. 우리는 이 책에 Comment를 열어놓겠습니다. 오셔서 언제든 Comment를 달아주세요. 나눌수록, 우리는 더욱 양질의 콘텐츠를 가지게 될 것입니다.

무료 책 프로젝트는 계속 이어나가겠습니다.
감사합니다.

저자소개

이호준

(現) 바울랩아이씨티기술연구원 대표이사
(現) 바울랩아이씨티컴퓨터학원 대표이사
(現) 사도출판 대표이사
(現) 바울랩미디어 대표이사
(現) 주식회사 위니브 대표이사
(現) GDG(Google Developers Group) JEJU Organizer
(現) 제주스타트업협회 부회장
(現) 제주대학교 풀스택 수업 강사
(現) 제주코딩베이스캠프 운영진

직책과 감투가 많지만, 사회에 공헌하며 미래를 꿈꾸는 아이들 가르치며 소박하게 살고 싶은 제주 도민입니다.

강민정

(前) 바울랩아이씨티기술연구원 연구원
제주코딩베이스캠프 2기 수료
카카오와 함께하는 제주 코딩 베이스캠프 7&8기 교재 집필
코딩도장 튜토리얼로 배우는 Python 문제풀이 외 5권 집필

김대현

(前) 제주대학교 전산통계학과 전공
(現) 빅데이터 전략 마에스트로 교육생
(現) 바울랩아이씨티기술연구원 연구원

김유진

(前) 제주대학교 컴퓨터공학 전공
(現) 바울랩아이씨티기술연구원 연구원
제주코딩베이스캠프 Code Festival: Python 100제 집필

김혜원

(前) 제주대학교 컴퓨터공학 전공
(現) 바울랩아이씨티기술연구원 연구원
카카오와 함께하는 제주 코딩 베이스캠프 11기 스탭
코딩도장 튜토리얼로 배우는 Python 문제풀이 외 4권 집필

웨이드

(前) dktechin 검색개발팀 팀원
Javascript Tutorials 집필
BLOCKCHAIN DAPP PLATFORMS FOR JEJU 집필

이현창

(前) 제주대학교 전산통계학과 전공
(現) 빅데이터 전략 마에스트로 교육생
(現) 바울랩아이씨티기술연구원 연구원

정윤하

(前) 제주대학교 관광개발학과, 관광융합소프트웨어 전공
(現) 바울랩아이씨티기술연구원 연구원
카카오와 함께하는 제주 코딩 베이스캠프 7&8기 교재 집필
바울랩 네이버 블로그 운영 및 '노션 좀 잘 써볼까' 연재 중

차경림

(前) 제주대학교 산업디자인학부 문화조형디자인 전공
(現) 바울랩아이씨티기술연구원 연구원

최원범

(現) 제주대학교 컴퓨터공학 전공
(現) 바울랩아이씨티기술연구원 연구원
제주코딩베이스캠프 Code Festival: Python 100제 집필

교단에서 강의할 때 한 학생이 찾아왔습니다. 가정 사정이 어려운데 SW를 배울 수 있는 학원은 가격이 너무 비싸고 그렇다고 혼자 공부할 수 있는 공부는 아니라는 막막함에 찾아온 학생이었습니다.

소프트웨어는 이제 삶에 한 부분이고 무엇을 꿈꾸던지 소프트웨어를 배워야 한다면 그 진입장벽 또한 낮아져야 합니다. 물론 SW 전문 교육 기관을 운영하는 입장에서 어려운 과제임이 확실합니다. 그러나 어른들의 숙제가 이제 막 꿈꾸기 시작한 학생들에게 전가되어서는 안됩니다.

또한 쉽게, 다양하게 배울 수 있어야 합니다. 학생에게 두꺼운 책과 끝내지 못한 책은 그 자체로 진입장벽이 됩니다. 저희가 출판하는 책은 여러분이 끝낸 한 권의 책이 되어 더 깊은 학문으로 가기 위한 발판이 되길 원합니다.

가정 형편이 어려워도 최고의 교육을 누구나 누릴 수 있도록, 이미 주어진 환경이 아닌 노력과 열정과 꿈의 크기만큼 성장할 수 있도록, 배움의 기쁨과 문제 해결의 쾌감을 누릴 수 있도록. 우리는 노력하겠습니다.

이호준

대상 독자

이 책은 다양한 유형을 살펴보며 자바스크립트 활용을 통해 문제를 풀 수 있게 하고, 문제 해결능력을 키우는데 목적을 두었습니다.

자바스크립트에 입문한 사람, 문법은 알고 있지만 문제해결을 어떤 식으로 해야하는지 모르는 사람들을 위한 책입니다. 이론과 개념 위주의 책을 통해 기본을 쌓았다면, 스스로 부족한 부분을 점검하는 과정이 필요합니다. JavaScript 100제를 통해 다시 기초를 다지고 문제 해결능력을 기를 수 있습니다.

이 책의 내용

자바스크립트로 풀 수 있는 100문제를 만들었습니다. 50문제씩 1부(초급, 중급)와 2부(중급, 고급)로 나누어져 있습니다.

1부

- 1-50번 까지의 문제가 있습니다.
- 자바스크립트 기초를 다시 한번 돌아보기 좋은 문제들로 구성되어있습니다.
- 입문자들에게 추천합니다.

2부

- 51-100번까지의 문제가 있습니다.
- 초반부는 1부와 마찬가지로 기초지식이 있다면 해결할 수 있는 문제로 구성하였습니다.
- 중,후반부는 어느 정도의 문제 해결능력을 갖추어야 풀 수 있도록 구성하였습니다.

이 책 안에는 답안지가 없습니다.

문제 및 답안지는

www.paullab.co.kr/codefestival.html 에 있으며 JavaScript기본 강좌와 PDF, Notion 페이지를 함께(Duplicate 가능합니다.) 제공합니다.

문제를 풀어보지 않고 답안지만 보시면 아니되십니다!

해설 강좌는 아래의 플랫폼에서 구매가 가능하세요.

책의 형태로 받고 싶으신 분은 리디북스, 반디엔루니스, 교보문고 등 여러분이 편하신 플랫폼에서 무료로 받으실 수 있습니다.

Contents

100문항 문제 2부

Chapter 2-1

문제 51 – merge sort를 만들어보자

문제 52 – quick sort

문제 53 – 괄호문자열

문제 54 – 연속되는 수

문제 55 – 하노이의 탑

문제 56 – 객체의 함수 응용

문제 57 – 1의 개수

문제 58 – 콤마 찍기

문제 59 – 빈칸 채우기

문제 60 – 번호 매기기

100문항 문제

문제51 : merge sort를 만들어보자

병합정렬(merge sort)은 대표적인 정렬 알고리즘 중 하나로 다음과 같이 동작합니다.

> 1. 리스트의 길이가 0 또는 1이면 이미 정렬된 것으로 본다. 그렇지 않은 경우에는
>
> 2. 정렬되지 않은 리스트를 절반으로 잘라 비슷한 크기의 두 부분 리스트로 나눈다.
>
> 3. 각 부분 리스트를 재귀적으로 합병 정렬을 이용해 정렬한다.
>
> 4. 두 부분 리스트를 다시 하나의 정렬된 리스트로 합병한다.

출처 : 위키피디아

다음 코드의 빈칸을 채워 병합정렬을 완성해 봅시다.

```javascript
function mergeSort(arr){
  if (arr.length <= 1){
    return arr;
  }

  const mid = Math.floor(arr.length / 2);
  const left = arr.slice(0,mid);
  const right = arr.slice(mid);

  return merge(mergeSort(left), mergeSort(right));
}

function merge(left, right){
  let result = [];

  while (/*빈칸을 채워주세요*/ && /*빈칸을 채워주세요*/){
    if (/*빈칸을 채워주세요*/){
      result.push(left.shift());
    } else {
      result.push(right.shift());
    }
  }
  while (left.length) {
    /*빈칸을 채워주세요*/
  }
  while (right.length) {
    /*빈칸을 채워주세요*/
  }

  return result;
}

const array = prompt('배열을 입력하세요').split(' ').map(n => parseInt(n, 10));

console.log(mergeSort(array));
```

JavaScript

문제52 : quick sort

다음 빈 칸을 채워 퀵 정렬을 완성해주세요.

```javascript
function quickSort(arr){
  if (arr.length <= 1){
    return arr;
  }

  const pivot = arr[0];
  const left = [];
  const right = [];

  for (let i=1; i<arr.length; i++){
    if(/*빈칸을 채워주세요*/){
      left.push(arr[i]);
    } else {
      right.push(arr[i]);
    }
  }
  return /*빈칸을 채워주세요*/
}

const array = prompt('배열을 입력하세요').split(' ').map(n => parseInt(n, 10));

console.log(quickSort(array));
```

JavaScript

문제53 : 괄호 문자열

괄호 문자열이란 괄호 기호인 '{', '}', '[', ']', '(', ')' 와 같은 것을 말한다. 그중 괄호의 모양이 바르게 구성된 문자열을 **바른 문자열**, 그렇지 않은 문자열을 **바르지 않은 문자열**이라 부르도록 하자.

(())와 같은 문자열은 바른 문자열이지만 ()() 와 같은 문자열은 바르지 않은 문자열이다.
(해당 문제에서는 소괄호만 판별하지만, 중괄호와 대괄호까지 판별해 보세요.)

입력으로 주어진 괄호 문자열이 바른 문자열인지 바르지 않은 문자열인지 "YES"와 "NO"로 구분된 문자열을 출력해보자.

문제54 : 연속되는 수

은주는 놀이공원 아르바이트를 하고 있다. 은주가 일하는 놀이공원에서는 현재 놀이공원 곳곳에 숨겨진 숫자 스탬프를 모아 오면 선물을 주는 이벤트를 하고 있다. 숫자 스탬프는 매일 그 수와 스탬프에 적힌 숫자가 바뀌지만 그 숫자는 항상 연속된다.
그런데 요즘 다른 날에 찍은 스탬프를 가지고 와 선물을 달라고 하는 손님이 늘었다.

스탬프에 적힌 숫자가 공백으로 구분되어 주어지면 이 숫자가 연속수인지 아닌지 "YES"와 "NO"로 판별하는 프로그램을 작성하시오

입력1
```
1 2 3 4 5
```

출력1
```
YES
```

입력2
```
1 4 2 6 3
```

출력2
```
NO
```

JavaScript

문제55 : 하노이의 탑

하노이의 탑은 프랑스 수학자 에두아르드가 처음으로 발표한 게임입니다. 하노이의 탑은 A, B, C 3개의 기둥과 기둥에 꽂을 수 있는 N 개의 원판으로 이루어져 있습니다. 이 게임에서 다음의 규칙을 만족해야 합니다.

> 1. 처음에 모든 원판은 A 기둥에 꽂혀 있다.
> 2. 모든 원판의 지름은 다르다.
> 3. 이 원반은 세 개의 기둥 중 하나에 반드시 꽂혀야 한다.
> 4. 작은 원반 위에 큰 원반을 놓을 수 없다.
> 5. 한 번에 하나의 원판(가장 위에 있는 원판)만을 옮길 수 있다.

이 규칙을 만족하며 A 기둥에 있는 원반 N 개를 모두 C 원반으로 옮기고 싶습니다.
모든 원반을 옮기기 위해 실행되어야 할 최소 원반 이동 횟수를 계산하는 프로그램을 완성해 주세요.

```javascript
const route = [];

function hanoi(num, start, end, temp){
  //원판이 한 개일 때에는 바로 옮기면 됩니다.
  if (num === 1) {
    route.push([start, end]);
    return NaN;
  }

  //원반이 n-1개를 경유기둥으로 옮기고
  hanoi(/*내용을 채워주세요.*/);
  //가장 큰 원반은 목표기둥으로
  route.push(/*내용을 채워주세요.*/);
  //경유기둥과 시작기둥을 바꿉니다.
  hanoi(/*내용을 채워주세요.*/);
}

hanoi(3, 'A', 'B', 'C');
console.log(route);
console.log(route.length);
```

JavaScript

문제56 : 객체의 함수 응용

다음의 객체가 주어졌을 때 한국의 면적과 가장 비슷한 국가와 그 차이를 출력하세요.

데이터

```
nationWidth = {
      'korea': 220877,
      'Rusia': 17098242,
      'China': 9596961,
      'France': 543965,
      'Japan': 377915,
      'England' : 242900,
}
```

출력

```
England 22023
```

JavaScript

문제57 : 1의 개수

0부터 1000까지 1의 개수를 세는 프로그램을 만들려고 합니다. 예를 들어 0부터 20까지 1의 개수를 세어본다면 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19에 각각 1이 들어가므로 12개의 1이 있게 됩니다. 11은 1이 2번 들어간 셈이죠.

그렇다면 0부터 1000까지 수에서 1은 몇 번이나 들어갔을까요? 출력해 주세요.

문제58 : 콤마 찍기

원범이는 편의점 아르바이트가 끝난 후 정산을 하고자 합니다.
정산을 빨리하고 집에 가고 싶은 원범이는 프로그램을 만들려고 합니다.

숫자를 입력받고 천 단위로 콤마(,)를 찍어주세요.

예를 들어, 123456789를 입력받았으면 123,456,789를 출력해야 합니다.

문제59 : 빈칸채우기

총 문자열의 길이는 50으로 제한하고 사용자가 문자열을 입력하면 그 문자열을 가운데 정렬을 해주고, 나머지 빈 부분에는 '='을 채워 넣어주세요.

입력
```
hi
```

출력
```
========================hi========================
```

JavaScript

문제60 : 번호 매기기

새 학기가 되어 **이름을 가나다 순서대로 배정하고 번호를 매기려고 합니다.**
데이터에 입력된 이름을 아래와 같이 출력해 주세요.

데이터

```
students = ['강은지','김유정','박현서','최성훈','홍유진','박지호','권윤일',
            '김채리','한지호','김진이','김민호','강채연']
```

출력

```
번호: 1, 이름: 강은지
번호: 2, 이름: 강채연
번호: 3, 이름: 권윤일
번호: 4, 이름: 김민호
번호: 5, 이름: 김유정
번호: 6, 이름: 김진이
번호: 7, 이름: 김채리
번호: 8, 이름: 박지호
번호: 9, 이름: 박현서
번호: 10, 이름: 최성훈
번호: 11, 이름: 한지호
번호: 12, 이름: 홍유진
```

JavaScript

Chapter 2-2

문제 61 – 문자열 압축하기

문제 62 – 20190923 출력하기

문제 63 – 친해지고 싶어

문제 64 – 이상한 엘리베이터

문제 65 – 변형된 리스트

문제 66 – 블럭탑쌓기

문제 67 – 민규의 약수

문제 68 – 버스 시간표

문제 69 – 골드바흐의 추측

문제 70 – 행렬 곱하기

100문항 문제

문제61 : 문자열 압축하기

문자열을 입력받고 연속되는 문자열을 압축해서 표현하고 싶습니다.

입력
```
aaabbbbcdddd
```

출력
```
a3b4c1d4
```

JavaScript

문제62 : 20190923출력하기

`20190923`을 출력합니다. 아래 기준만 만족하면 됩니다.

1. 코드 내에 숫자가 없어야 합니다.
    - 예) console.log(20190923)이라고 하시면 안됩니다.
2. 파일 이름이나 경로를 사용해서는 안됩니다.
3. 시간, 날짜 함수를 사용해서는 안됩니다.
4. 에러 번호 출력을 이용해서는 안됩니다.
5. input을 이용해서는 안됩니다.

문제63 : 친해지고 싶어

한국 대학교의 김한국 교수님은 학생들과 친해지기 위해서 딸에게 줄임말을 배우기로 했습니다.
딸은 '**복잡한 세상 편하게 살자**'라는 문장을 '**복세편살**'로 줄여 말합니다.

교수님이 줄임말을 배우기 위해 아래와 같이 **어떤 입력이 주어지면 앞 글자만 줄여 출력**하도록 해주세요.
입력은 한글 혹은 영어로 입력되며, 띄어쓰기를 기준으로 하여 짧은 형태로 출력합니다.

입력
복잡한 세상 편하게 살자

출력
복세편살

JavaScript

문제64 : 이상한 엘레베이터

정량 N에 정확히 맞춰야만 움직이는 화물용 엘리베이터가 있습니다.
화물은 7kg, 3kg 두 가지이며 팔이 아픈 은후는 가장 적게 화물을 옮기고 싶습니다.

예를 들어 정량이 24kg이라면 3kg 8개를 옮기는 것보다는
7kg 3개, 3kg 1개 즉 4개로 더 적게 옮길 수 있습니다.

입력
정량 N이 입력됩니다.

출력
가장 적게 옮길 수 있는 횟수를 출력합니다.
만약 어떻게 해도 정량이 N이 되지 않는다면 -1을 출력합니다.

문제65 : 변형된 리스트

a = [1, 2, 3, 4]
b = [a, b, c, d]
이런 리스트가 있을 때 **[[1, a], [b, 2], [3, c], [d, 4]]** 이런 식으로 a, b 리스트가 번갈아가면서 출력되게 해주세요.

문제66 : 블럭탑쌓기

탑을 쌓기 위해 각 크기별로 준비된 블럭들을 정해진 순서에 맞게 쌓아야 합니다.
순서에 맞게 쌓지 않으면 무너질 수 있습니다.

예를 들면 정해진 순서가 BAC 라면 A 다음 C가 쌓아져야 합니다.
선행으로 쌓아야 하는 블럭이 만족된 경우라면 탑이 무너지지 않습니다.

- B를 쌓지 않아도 A와 C를 쌓을 수 있습니다.
- B 다음 블럭이 C가 될 수 있습니다.

쌓아져 있는 블럭 탑이 순서에 맞게 쌓아져 있는지 확인하세요.

1. 블럭은 알파벳 대문자로 표기합니다.
2. 규칙에 없는 블럭이 사용될 수 있습니다.
3. 중복된 블럭은 존재하지 않습니다.

입력
```
탑 = ["ABCDEF", "BCAD", "ADEFQRX", "BEDFG", "EFGHZ"]
규칙 = "ABD"
```

출력
```
["가능", "불가능", "가능", "가능", "가능"]
```

JavaScript

문제67 : 민규의 악수

광장에서 모인 사람들과 악수를 하는 행사가 열렸습니다.
참가자인 민규는 몇명의 사람들과 악수를 한 후 중간에 일이 생겨 집으로 갔습니다.

이 행사에서 진행된 악수는 총 n번이라고 했을 때,
민규는 몇 번의 악수를 하고 집으로 돌아갔을까요?
그리고 민규를 포함한 행사 참가자는 몇 명일까요?

- 악수는 모두 1대 1로 진행이 됩니다.
- 민규를 제외한 모든 참가자는 자신을 제외한 참가자와 모두 한번씩 악수를 합니다.
- 같은 상대와 중복된 악수는 카운트 하지 않습니다.
- 민규를 제외한 참가자는 행사를 모두 마쳤습니다.

예를들어 행사에서 59회의 악수가 진행되었다면 민규는 4번의 악수를 하였고 민규를 포함한 참가자는 12명이다.

행사에서 진행된 악수 횟수(n)를 입력으로 받으면 민규의 악수 횟수와 행사 참가자 수가 출력됩니다.

```
입력
59

출력
[4, 12] //[악수 횟수, 행사 참가자 수]
```

JavaScript

문제68 : 버스 시간표

학교가 끝난 지원이는 집에 가려고 합니다. 학교 앞에 있는 버스 시간표는 너무 복잡해서 버스 도착 시간이 몇 분 남았는지 알려주는 프로그램을 만들고 싶습니다.

버스 시간표와 현재 시간이 주어졌을 때 버스 도착 시간이 얼마나 남았는지 알려주는 프로그램을 만들어주세요.

- 버스 시간표와 현재 시간이 입력으로 주어집니다.
- 출력 포맷은 "00시 00분"입니다.
  만약 1시간 3분이 남았다면 "**01시간 03분**"으로 출력해야 합니다.
- 버스 시간표에 현재 시간보다 이전인 버스가 있다면 "**지나갔습니다.**"라고 출력합니다.

```
입력
["12:30", "13:20", "14:13"]
"12:40"

출력
['지나갔습니다', '00시간 40분', '01시간 33분']
```

JavaScript

문제69 : 골드바흐의 추측

골드바흐의 추측(Goldbach's conjecture)은 오래전부터 알려진 정수론의 미해결 문제로, 2보다 큰 모든 짝수는 두 개의 소수(Prime number)의 합으로 표시할 수 있다는 것이다. 이때 하나의 소수를 두 번 사용하는 것은 허용한다. - 위키백과

위 설명에서 2보다 큰 모든 짝수를 두 소수의 합으로 나타낸 것을 골드바흐 파티션이라고 합니다.

예)
100 == 47 + 53
56 == 19 + 37

2보다 큰 짝수 n이 주어졌을 때, 골드바흐 파티션을 출력하는 코드를 작성하세요.

* 해당 문제의 출력 형식은 자유롭습니다. 가능하시다면 골드바흐 파티션 모두를 출력하거나, 그 차가 작은 것을 출력하거나 그 차가 큰 것 모두 출력해보세요.

문제70 : 행렬 곱하기

행렬 2개가 주어졌을 때 곱할 수 있는 행렬인지 확인하고 곱할 수 있다면 그 결과를 출력하고, 곱할 수 없다면 -1을 출력하는 프로그램을 만들어주세요.

입력

```
a = [[1, 2],
     [2, 4]]

b = [[1, 0],
     [0, 3]]
```

출력

```
[[1, 6], [2, 12]]
```

JavaScript

Chapter 2-3

문제 71 – 깊이 우선 탐색

문제 72 – 너비 우선 탐색

문제 73 – 최단 경로 찾기

문제 74 – 최장 경로 찾기

문제 75 – 이상한 369

문제 76 – 안전한 땅

문제 77 – 가장 긴 공통 부분 문자열

문제 78 – 원형 테이블

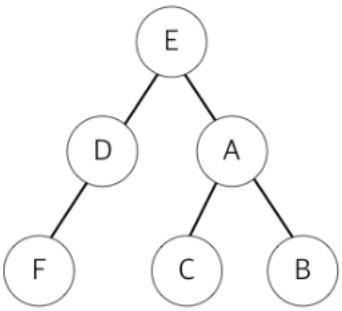
문제 79 – 순회하는 리스트

문제 80 – 순열과 조합

100문항 문제

문제71 : 깊이 우선 탐색

깊이 우선 탐색이란 목표한 노드를 찾기 위해 가장 우선순위가 높은 노드의 자식으로 깊이 들어갔다가 목표 노드가 존재하지 않으면 처음 방문한 노드와 연결된 다른 노드부터 그 자식 노드로 파고드는 검색 방법을 말합니다.



다음과 같이 리스트 형태로 노드들의 연결 관계가 주어진다고 할 때 깊이 우선 탐색으로 이 노드들을 탐색했을 때의 순서를 공백으로 구분하여 출력하세요.

데이터

```
graph = {'E': ['D', 'A'],
         'F': ['D'],
         'A': ['E', 'C', 'B'],
         'B': ['A'],
         'C': ['A'],
         'D': ['E','F']}
```

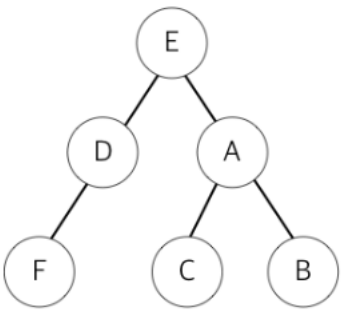
출력

```
E D F A C B
```

JavaScript

문제72 : 너비 우선 탐색

너비 우선 탐색이란 어떤 노드를 방문하여 확인한 후, 목표한 노드가 아니면 그 노드와 연결된 정점들 중에서 우선순위가 동일한 다른 노드를 찾고 그 순위에 없으면 그다음 순위 노드를 차례대로 찾는 방법이다.



다음과 같이 입력이 주어질 때 **너비 우선 탐색을 한 순서대로 노드의 인덱스를 공백 구분으로 출력하세요.**

데이터

```
graph = {'E': ['D', 'A'],
         'F': ['D'],
         'A': ['E', 'C', 'B'],
         'B': ['A'],
         'C': ['A'],
         'D': ['E','F']}
```

출력

```
E D A F C B
```

JavaScript

문제73 : 최단 경로 찾기

다음과 같이 노드의 연결 관계가 리스트 형태로 주어집니다. 그다음 경로를 구할 두 정점이 공백으로 구분되어 주어질 것입니다.

두 정점 사이를 이동할 수 있는 최단 거리를 출력하는 프로그램을 작성해 주세요.

이때 최단 거리란, 정점의 중복 없이 한 정점에서 다른 정점까지 갈 수 있는 가장 적은 간선의 수를 의미합니다.

데이터

```
graph = {'A': ['B', 'C'],
         'B': ['A', 'D', 'E'],
         'C': ['A', 'F'],
         'D': ['B'],
         'E': ['B', 'F'],
         'F': ['C', 'E']}
```

입력

```
A F
```

출력

```
2
```

JavaScript

문제74 : 최장 경로 찾기

다음과 같이 노드의 연결 관계가 주어집니다.
입력으로는 경로를 구할 두 정점의 번호가 공백으로 구분되어 주어집니다.
우리는 **이 두 정점으로 가기 위한 최대 거리**를 구하고자 합니다.

최대 거리란, 정점의 중복 없이 한 정점에서 다른 정점까지 경유할 수 있는 가장 많은 간선의 수를 뜻합니다.

데이터

```
graph = {1: [2, 3, 4],
         2: [1, 3, 4, 5, 6],
         3: [1, 2, 7],
         4: [1, 2, 5, 6],
         5: [2, 4, 6, 7],
         6: [2, 4, 5, 7],
         7: [3, 5, 6]}
```

입력

```
1 7
```

출력

```
6
```

JavaScript

문제75 : 이상한 369

369 게임을 하는데 조금 이상한 규칙이 있습니다. 3이나 6, 9 일 때만 박수를 쳐야합니다. 예를 들어 13, 16과 같이 3과 6, 9 만으로 된 숫자가 아닐 경우엔 박수를 치지 않습니다.
수현이는 박수를 몇 번 쳤는지 확인하고 싶습니다. 36일 때 박수를 쳤다면 박수를 친 횟수는 5번입니다.

n을 입력하면 박수를 몇 번 쳤는지 그 숫자를 출력해주세요.

👏	순서	👏	순서
3	1	39	6
6	2	63	7
9	3	66	8
33	4	69	9
36	5	93	10

```
입력
'93'

출력
10
```

JavaScript

문제76 : 안전한 땅

전쟁이 끝난 후, A 나라에서는 폐허가 된 도시를 재건하려고 한다. 그런데 이 땅은 전쟁의 중심지였으므로 전쟁 중 매립된 지뢰가 아직도 많이 남아 있다는 것이 판명되었다.
정부는 가장 먼저 지뢰를 제거하기 위해 수색반을 꾸렸다.

수색반은 가장 효율적인 지뢰 제거가 하고 싶다. 수색반은 도시를 격자무늬로 나눠놓고 자신들이 수색할 수 있는 범위 내에 가장 많은 지뢰가 매립된 지역을 가장 먼저 작업하고 싶다.

가장 먼저 테스트 케이스의 수를 나타내는 1이상 100 이하의 자연수가 주어진다.
각 테스트 케이스의 첫 줄에는 수색할 도시의 크기 a와 수색반이 한 번에 수색 가능한 범위 b가 주어진다. (a와 b 모두 정사각형의 가로 또는 세로를 나타낸다. 예를 들어 10이 주어지면 10x10칸의 크기를 나타낸다.)

그 후 a 줄에 걸쳐 도시 내 지뢰가 있는지의 여부가 나타난다.
0은 지뢰가 없음 1은 지뢰가 있음을 뜻한다.

각 테스트 케이스에 대해 수색 가능한 범위 bxb 내에서 찾아낼 수 있는 가장 큰 지뢰의 개수를 구하라.

입력

```
1
5 3
1 0 0 1 0
0 1 0 0 1
0 0 0 1 0
0 0 0 0 0
0 0 1 0 0
```

출력

```
3
```

JavaScript

문제77 : 가장 긴 공통 부분 문자열

가장 긴 공통 부분 문자열(Longest Common Subsequence)이란 A, B 두 문자열이 주어졌을 때 두 열에 공통으로 들어 있는 요소로 만들 수 있는 가장 긴 부분열을 말합니다. 여기서 부분열이란 다른 문자열에서 몇몇의 문자가 빠져 있어도 순서가 바뀌지 않은 열을 말합니다.

예를 들어 S1 = ['T', 'H', 'I', 'S', 'I', 'S', 'S', 'T', 'R', 'I', 'N', 'G', 'S'] S2 = ['T', 'H', 'I', 'S', 'I', 'S']라는 두 문자열이 있을 때 둘 사이의 부분 공통 문자열의 길이는 ['T', 'H', 'I', 'S', 'I', 'S']의 6개가 됩니다.

이처럼 **두 문자열이 주어지면 가장 긴 부분 공통 문자열의 길이를 반환하는 프로그램**을 만들어 주세요.

두 개의 문자열이 한 줄에 하나씩 주어집니다. 문자열은 알파벳 대문자로만 구성되며 그 길이는 100글자가 넘어가지 않습니다.

출력은 이 두 문자열의 가장 긴 부분 공통 문자열의 길이를 반환하면 됩니다.

- Test Case -

입력
THISISSTRINGS
THISIS

출력
6

-

입력
THISISSTRINGS
TATHISISKKQQAEW

출력
6

-

입력
THISISSTRINGS
KIOTHIKESSISKKQQAEW

출력
6

-

입력
THISISSTRINGS
TKHKIKSIS

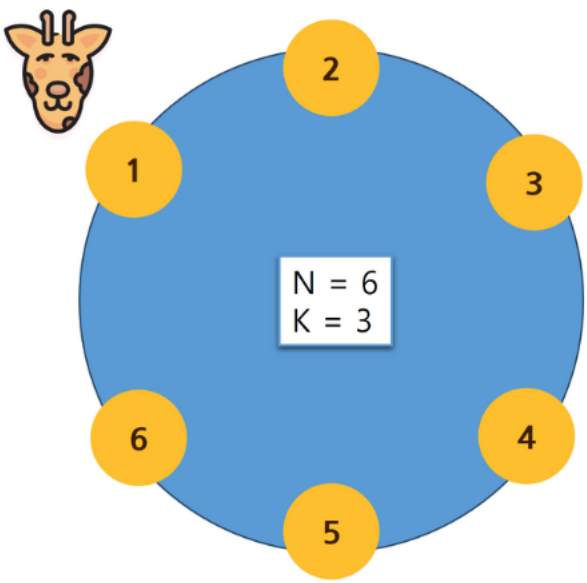
출력
3

문제78 : 원형 테이블

기린은 중국집에서 친구들과 만나기로 하고, 음식을 시켰습니다.
음식이 나오고 한참을 기다렸지만 만나기로 한 친구 2명이 오지 않았어요.

기린은 배가 너무 고파 혼자 음식을 먹기 시작합니다. 원형 테이블에는 N 개의 음식들이 있습니다.
한 개의 음식을 다 먹으면 그 음식의 시계방향으로 K 번째 음식을 먹습니다.
하지만 아직 오지 않은 친구들을 위해 2개의 접시를 남겨야 합니다.

마지막으로 남는 음식은 어떤 접시인가요?



입력은 2개의 정수로 이루어지며 공백으로 구분되어 입력됩니다.
첫 번째 숫자가 음식의 개수 N, 두 번째 숫자가 K입니다.
첫 번째 가져가는 음식이 K 번째 음식이며 나머지는 첫 번째 음식으로부터 시계방향으로 가져갑니다.

입력
```
6 3
```

남은 음식들의 번호를 배열의 형태로 출력합니다.

출력
```
[3, 5]
```

JavaScript

문제79 : 순회하는 리스트

다음의 값이 주어졌을 때

```
l = [10, 20, 25, 27, 34, 35, 39]
```

Plain Text

n번 순회를 결정합니다. 예를 들어 2번 순회면

```
l = [35, 39, 10, 20, 25, 27, 34]
```

Plain Text

여기서 변하기 전 원소와 변한 후 원소의 값의 차가 가장 작은 값을 출력하는 프로그램을 작성하세요.

예를 들어 2번 순회했을 때 변하기 전의 리스트와 변한 후의 리스트의 값은 아래와 같습니다.

순회전_리스트 = [10, 20, 25, 27, 34, 35, 39]
순회후_리스트 = [35, 39, 10, 20, 25, 27, 34]
리스트의_차 = [25, 19, 15, 7, 9, 8, 5]

39와 변한 후의 34 값의 차가 5이므로 리스트의 차 중 최솟값입니다. 따라서 39와 34의 인덱스인 6과 39와 34를 출력하는 프로그램을 만들어주세요.

```javascript
const l = [10, 20, 25, 27, 34, 35, 39]; //기존 입력 값
const n = parseInt(prompt('순회횟수는?'), 10);

function rotate(inL, inN){

    <빈칸을 채워주세요>
}

rotate(l, n)
```

JavaScript

입력
순회횟수는 : 2

출력
index : 6
value : 39, 34

JavaScript

문제80 : 순열과 조합

조합이란 원소들을 조합하여 만들 수 있는 경우의 수이며 원소의 순서는 신경 쓰지 않습니다.
순열이란 원소의 값이 같더라도 순서가 다르면 서로 다른 원소로 취급하는 선택법입니다.

한글의 자모 24자 중 자음은 총 14개입니다.
이 중 입력받은 자음을 n 개를 선택하여 나올 수 있는 모든 조합과, 조합의 수를 출력하고 싶습니다.

'한글 맞춤법'의 제2장 제4항에서는 한글의 기본 자모 24자 "ㄱ(기역), ㄴ(니은), ㄷ(디귿), ㄹ(리을), ㅁ(미음), ㅂ(비읍), ㅅ(시옷), ㅇ(이응), ㅈ(지읒), ㅊ(치읓), ㅋ(키읔), ㅌ(티읕), ㅍ(피읖), ㅎ(히읗), ㅏ(아), ㅑ(야), ㅓ(어), ㅕ(여), ㅗ(오), ㅛ(요), ㅜ(우), ㅠ(유), ㅡ(으), ㅣ(이)"를 제시

나올 수 있는 모든 조합을 아래와 같이 출력해 주세요.

<--**요구 조건**-->
1. 첫 번째 입력으로 선택할 한글 자음이 주어집니다.
2. 두 번째 입력으로 조합의 수가 주어집니다.
3. 주어진 조합의 수에 따라 조합과 조합의 수를 출력해 주세요.

```
입력
ㄱ,ㄴ,ㄷ,ㄹ
3

출력
['ㄱㄴㄷ', 'ㄱㄴㄹ', 'ㄱㄷㄹ', 'ㄴㄷㄹ']
4
```

JavaScript

Chapter 2-4

문제 81 – 지뢰 찾기

문제 82 – 수학 괄호 파싱

문제 83 – 수학 괄호 파싱 2

문제 84 – 숫자 뽑기

문제 85 – 숫자 놀이

문제 86 – 회전 초밥

문제 87 – 천하제일 먹기 대회

문제 88 – 지식이의 게임 개발

문제 89 – 지식이의 게임 개발 2

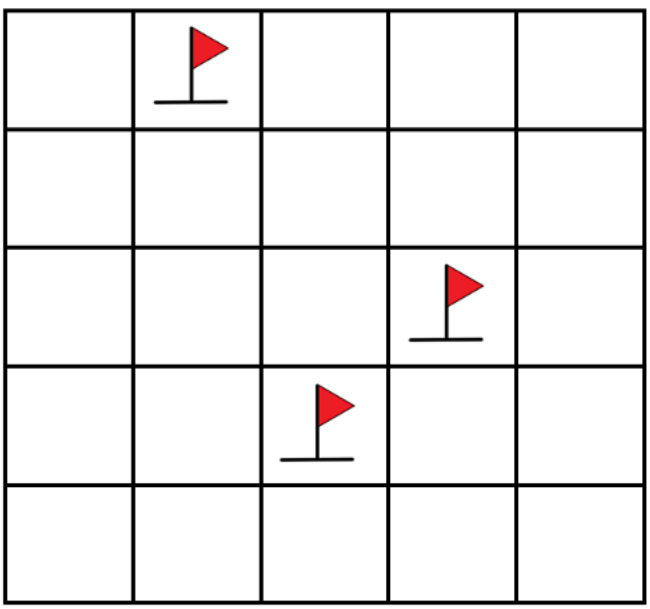
문제 90 – 같은 의약 성분을 찾아라

100문항 문제

문제81 : 지뢰찾기

지뢰를 찾는 문제입니다. 다음 그림처럼 깃발 주위에는 지뢰가 사방으로 있습니다. **깃발의 위치를 입력받아 지뢰와 함께 출력 해주는 프로그램**을 만드세요.

- 아래 Case 외 예외 사항은 고려하지 않습니다.
  (예를 들어 깃발이 붙어 있을 경우는 고려하지 않습니다.)
  실력이 되시는 분들은 깃발이 붙어있을 상황까지 고려해 주세요.



데이터

```
let flag = []; //지뢰 없이 깃발만 있는 리스트
let minesweeper = []; //지뢰를 찾은 리스트
let count = 0;

console.log(flag);
console.log(minesweeper);
```

입력

```
0 1 0 0 0
0 0 0 0 0
0 0 0 1 0
0 0 1 0 0
0 0 0 0 0
//"0 1 0 0 0\n0 0 0 0 0\n0 0 0 1 0\n0 0 1 0 0\n0 0 0 0 0"
```

출력

```
* f * 0 0
0 * 0 * 0
0 0 * f *
0 * f * 0
0 0 * 0 0
```

JavaScript

문제82 : 수학 괄호 파싱

수학공식이 제대로 입력이 되었는지 판단하는 코드를 작성하려 합니다. 괄호는 소괄호밖에 없습니다.

입출력 예시

```
데이터 입력(1), 프로그램 종료(2) : 1
데이터를 입력하세요: 3 + 5
True


데이터 입력(1), 프로그램 종료(2) : 1
데이터를 입력하세요: 5 + 7) * (3 * 5)
False
```

JavaScript

```javascript
function math(e){

    //코드를 작성해주세요

}

while (1){
    let order = prompt('데이터 입력(1), 프로그램 종료(2)');
    if (order === 1){
        const ex = prompt('데이터를 입력하세요');
        console.log(math(ex));
    } else {
        break;
    }
}
```

JavaScript

문제83 : 수학 괄호 파싱 2

수학공식이 제대로 입력이 되었는지 판단하는 코드를 작성하려 합니다.
괄호는 소괄호와 중괄호가 있습니다.

입출력 예시

```
데이터 입력(1), 프로그램 종료(2) : 1
데이터를 입력하세요: 5 + 7 * {(3 * 5)}
True

데이터 입력(1), 프로그램 종료(2) : 1
데이터를 입력하세요: 5 + 7){ * (3 * 5)
False

데이터 입력(1), 프로그램 종료(2) : 2
```

Plain Text

```javascript
function math(e){

    //코드를 작성해주세요
}

while (1){
    const order = prompt('데이터 입력(1), 프로그램 종료(2)');
    if (order == '1'){
        const ex = input('데이터를 입력하세요');
        console.log(math(ex));
    } else {
        break;
    }
}
```

JavaScript

문제84 : 숫자뽑기

소정이는 어떤 숫자에서 k개의 수를 뽑았을 때 가장 큰 수를 찾는 놀이를 하고 있습니다.
예를 들어, 숫자 1723에서 두 개의 수를 뽑으면 [17, 12, 13, 72, 73, 23] 을 만들 수 있습니다.
이 중 가장 큰 수는 73입니다.

위 예시처럼 **어떤 수 n에서 k개의 수를 선택하여 만들 수 있는 수 중에서 가장 큰 수**를 찾아 주세요.

문제85 : 숫자놀이

일정한 규칙을 가지고 있는 숫자를 나열하는 놀이를 하는 중입니다.
이전 숫자에서 각 숫자의 개수를 나타내어 숫자로 만들고 다시 그 숫자를 같은 규칙으로 만들며 나열합니다.
이 놀이는 1부터 시작합니다.
다음 수는 1이 1개이기 때문에 '11'이 되고,
'11'에서 1이 2개이기 때문에 그다음은 '12'가 됩니다.

즉,
1. 1 → (1)
2. 11 → (1이 1개)
3. 12 → (1이 2개)
4. 1121 → (1이 1개 2가 1개)
5. 1321 → (1이 3개 2가 1개)
6. 122131 → (1이 2개 2가 1개 3이 1개)
7. 132231 → (1이 3개 2가 2개 3이 1개)

위와 같이 진행되는 규칙을 통해 진행 횟수 N을 입력받으면 해당되는 수를 출력하세요.

입력
6

출력
122131

JavaScript

문제86 : 회전 초밥

쉔은 회전 초밥집에 갔습니다.
초밥집에 간 쉔은 각 초밥에 점수를 매기고 낮은 점수의 순서로 초밥을 먹으려 합니다.
이때 n위치에 놓여진 초밥을 먹고자 할 때 접시가 몇 번 지나가고 먹을 수 있을지 출력하세요.

1. 초밥은 놓여진 위치에서 옮겨지지 않습니다.
2. 지나간 초밥은 나머지 초밥이 지나간 후에 다시 돌아옵니다.
3. 초밥은 1개 이상 존재합니다.

예)
A, B, C, D, E 초밥이 있고 각 점수가 1, 1, 3, 2, 5 일 때 3번째(C초밥)을 먹게 되는 순서는
점수가 1인 초밥 A와 B를 먹고 다음으로 점수가 2인 D 초밥을 먹어야 .
A B C D E 의 순서로 접시가 도착하지만 C가 도착했을때 먹지 못하는 상황이 옵니다.
2점을 주었던 D를 먼저 먹어야 C를 먹을 수 있습니다.
즉, A B C D E C 의 순서로, 접시가 5번 지나가고 먹게 된다.

입력
point = [1,1,3,2,5]
dish = 3

출력
5

입력
point = [5,2,3,1,2,5]
dish = 1

출력
10

```
//point 각 접시별 점수가 들어있는 배열
//dish 먹고자하는 접시의 위치
```

JavaScript

문제87 : 천하제일 먹기 대회

천하제일 먹기 대회가 개최되었습니다.
이 대회는 정해진 시간이 끝난 후 음식을 먹은 그릇 개수를 파악한 후 각 선수들의 등수를 매깁니다.

1. 같은 이름의 선수는 없습니다.
2. 접시의 수가 같은 경우는 없습니다.

입력 예1)

```
손오공 야모챠 메지터 비콜로
70 10 55 40
```

JavaScript

출력 예1)

```
{'손오공': 1, '메지터': 2, '비콜로': 3, '야모챠': 4}
```

JavaScript

입력 예2)

```
["홍길동","엄석대","연개소문","김첨지"]
[2, 1, 10, 0]
```

JavaScript

출력 예2)

```
{'연개소문': 1, '홍길동': 2, '엄석대': 3, '김첨지': 4}
```

JavaScript

문제88 : 지식이의 게임 개발

지식이는 게임을 만드는 것을 좋아합니다. 하지만 매번 다른 크기의 지도와 장애물을 배치하는데 불편함을 겪고 있습니다. 이런 불편함을 해결하기 위해 **지도의 크기와 장애물의 위치, 캐릭터의 위치만 입력하면 게임 지형을 완성해 주는 프로그램**을 만들고 싶습니다. 지식이를 위해 게임 지형을 만드는 프로그램을 작성해 주세요.

- 가로(n), 세로(m)의 크기가 주어집니다.
- 지형의 테두리는 벽으로 이루어져 있습니다.
- 캐릭터가 있는 좌표가 배열 형태로 주어집니다.
- 장애물이 있는 좌표가 2차원 배열 형태로 주어집니다.

지도는 n x m 크기의 배열이며 배열 안의 값은
  -움직일 수 있는 공간(0)
  -캐릭터(1)
  -벽(2)
3개로 구분되어 있습니다.

입출력예시

입력
가로 = 4
세로 = 5
캐릭터위치 = [0,0]
장애물 = [[0,1],[1,1],[2,3],[1,3]]

```
make_map(가로, 세로, 캐릭터위치, 장애물)
```

출력
```
[2, 2, 2, 2, 2, 2]
[2, 1, 2, 0, 0, 2]
[2, 0, 2, 0, 2, 2]
[2, 0, 0, 0, 2, 2]
[2, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 2]
[2, 2, 2, 2, 2, 2]
```

JavaScript

문제89 : 지식이의 게임 개발 2

(연계형 문제 - 88번을 먼저 풀고 오셔야 합니다!)
제코베의 도움을 받아 성공적으로 지도를 만들어낸 지식이는 캐릭터의 움직임을 구현했습니다.
하지만 지도 위의 캐릭터 위치를 나타내는데 문제가 발생했습니다.
지식이는 지도 위에서 캐릭터의 위치를 나타내기 위해 다시 한번 제코베에 도움을 요청합니다.

지도 위에서 캐릭터의 위치를 나타내주세요

1. 지도는 88번 문제의 해답을 사용해 주세요
2. 입력값은 지도, 캐릭터의 움직임입니다.
3. 캐릭터의 움직임은 { 상:1, 하:2, 좌:3, 우:4 }로 정수로 이루어진 배열이 들어갑니다.
4. 벽과 장애물은 통과할 수 없습니다.
5. 마지막 캐릭터의 위치를 반영한 지도를 보여주고 위치를 반환하는 함수를 작성해 주세요.

데이터

```
가로 = 4
세로 = 5
캐릭터위치 = [0, 0]
장애물 = [[0,1],[1,1],[2,3],[1,3]]
console.log('캐릭터 이동 전 지도')
지도 = make_map(가로, 세로, 캐릭터위치, 장애물)

움직임 = [2,2,2,4,4,4]
console.log('캐릭터 이동 후 지도')
캐릭터 위치 = move(지도, 움직임)
```

출력

```
캐릭터 이동 전 지도
[2, 2, 2, 2, 2, 2]
[2, 1, 2, 0, 0, 2]
[2, 0, 2, 0, 2, 2]
[2, 0, 0, 0, 2, 2]
[2, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 2]
[2, 2, 2, 2, 2, 2]

캐릭터 이동 후 지도
[2, 2, 2, 2, 2, 2]
[2, 0, 2, 0, 0, 2]
[2, 0, 2, 0, 2, 2]
[2, 0, 0, 0, 2, 2]
[2, 0, 0, 0, 1, 2]
[2, 0, 0, 0, 0, 2]
[2, 2, 2, 2, 2, 2]
캐릭터위치 : [4, 4]
```

JavaScript

문제 90 : 같은 의약 성분을 찾아라!

의약품 성분이 총 8개인 약품들이 있습니다. 예를 들어 다음 데이터는 총 8개의 성분을 갖습니다.

판콜비 = 'ABCDEFGH'
넥타이레놀 = 'EFGHIJKL'

특정 약품 A의 성분이 공개되었을 때, 이와 유사한 성분을 가진 데이터들의 출력을 구하는 문제입니다.

입력 : 'ABCDEFGH' 4
데이터 : 'EFGHIJKL', 'EFGHIJKM', 'EFGHIJKZ' 등 1만 개의 데이터
출력 : 'EFGHIJKL', 'EFGHIJKM', 'EFGHIJKZ' 등 4개의 요소가 같은 약품 전부(4개 이상이 아니며 같은 요소가 4개인 것을 출력해야 합니다.)

* 해당 문제는 시간제한이 있습니다.
* 제약 데이터의 성분은 중복이 될 수 없습니다.
(예를 들어 'AAABBBAB'와 같은 데이터는 없습니다.)

Chapter 2-5

문제 91 – 반평균 등수

문제 92 – 키보드 고장

문제 93 – 페이지 교체 - 선입선출 알고리즘

문제 94 – 페이지 교체 - LRU 알고리즘

문제 95 – 도장찍기

문제 96 – 넓은 텃밭 만들기!

문제 97 – 택배 배달

문제 98 – 청길이의 패션 대회

문제 99 – 토끼들의 행진

문제 100 – 퍼즐게임

100문항 문제

문제 91 : 반평균 등수

한 반에 30명인 학생, 총 7개의 반 점수가 '국어, 영어, 수학, 사회, 과학' 순서로 있는 다중 리스트를 랜덤 한 값으로 만들어주시고 아래 값을 모두 출력하세요.

1. 반 점수 모두가 담긴 전교 점수 다중 리스트를 만들어주세요.
2. 반 평균을 구하세요.
3. 반 1등 점수를 구하세요.
4. 전교 평균을 구하세요.

(출력 형식은 상관없습니다.)

```javascript
//아래 코드는 힌트입니다.

let student_score = [];
let class_score = [];
let total_score = [];

for (let i=0; i<5; i++){
    student_score.push(Math.floor(Math.random()*100)+1);
}

console.log(student_score);
```

JavaScript

문제92 : 키보드 고장

P 회사의 회계를 처리하던 은정은 커피를 마시다가 키보드에 커피를 쏟고 말았습니다.
휴지로 닦고 말려보았지만 숫자 3, 4, 6이 도통 눌리지 않습니다.
10분 뒤, 모든 직원들에게 월급을 입금해 주어야 합니다.
여유 키보드는 없으며, 프로그래밍을 매우 잘하고, 모든 작업을 수작업으로 하고 있습니다.

이에 눌리지 않는 키보드를 누르지 않고 월급 입금을 두 번에 나눠주고 싶습니다.

1. 직원은 2000명이며, 3초 이내 수행을 해야합니다.
2. 입력값의 형식은 csv파일형식이며 이과장 '3,000,000', 'S은행', '100-0000-0000-000' 형식으로 주어집니다.
3. 출력값의 형식은 csv파일형식이며 이과장 '1,500,000', '1,500,000', 'S은행', '100-0000-0000-000' 입니다. 또는 '1,000,000', '2,000,000', 'S은행', '100-0000-0000-000' 도 괜찮습니다.

```
이대표,'333,356,766','S은행','100-0000-0000-001'
최차장,'5,000,000','S은행','100-0000-0000-002'
이과장,'3,200,000','S은행','100-0000-0000-003'
홍팀장,'3,300,000','S은행','100-0000-0000-004'
이대리,'5,300,000','S은행','100-0000-0000-005'
```

Plain Text

문제93 : 페이지 교체 - 선입선출 알고리즘

페이지 교체 알고리즘은 메모리를 관리하는 운영체제에서, 페이지 부재가 발생하여 새로운 페이지를 할당하기 위해 현재 **할당된 페이지 중 어느 것을 교체할지를 결정하는 방법**입니다.
이 알고리즘이 사용되는 시기는 페이지 부재(Page Fault)가 발생해 새로운 페이지를 적재해야 하지만 페이지를 적재할 공간이 없어 이미 적재되어 있는 페이지 중 교체할 페이지를 정할 때 사용됩니다.
빈 페이지가 없는 상황에서 메모리에 적재된 페이지와 적재할 페이지를 교체함으로 페이지 부재 문제를 해결할 수 있습니다.
(wikipedia)

이 중 선입선출(FIFO) 알고리즘은 가장 먼저 들어와서 가장 오래있었던 페이지를 우선으로 교체시키는 방법을 의미합니다. 아래의 그림을 참고해주세요.

시간	1	2	3	4	5	6	7	8
들어오는 페이지	B	C	B	A	E	B	C	E
메모리	B	B	B	B	E	E	E	E
		C	C	C	C	B	B	B
				A	A	A	C	C
페이지교체 발생여부	F	F		F	F	F	F	

메모리의 크기가 i로 주어지고 들어올 페이지들이 n으로 주어졌을 때, 전체 실행시간을 구해주세요.

만약 스택 안에 같은 스케줄이 있다면 **hit** 이라고 하며 실행시간은 **1초** 입니다. 스택 안에 스케줄이 없다면 **miss** 라고 하며 실행시간은 **6초** 입니다.

페이지

Aa 페이지	# 페이지 프레임	# 실행시간
BCBAEBCE	3	38
ABCABCABC	3	24
ABCDABEABCDE	4	62
ABCEDF	0	36
ABCDABEA	3	43

COUNT 5

- 예제 1번을 보면 페이지 프레임의 개수는 3개이고 스케줄은 'BCBAEBCE' 입니다. 6번의 miss를 기록하므로 **6번** * **6초** = **36초**가 되고 2번의 hit을 기록하므로 **2번** * **1초** = **2초**입니다. 2개를 합한 값이 실행시간이므로, 38초가 됩니다.

문제94 : 페이지 교체 - LRU 알고리즘

LRU 알고리즘이란 페이지 교체 알고리즘으로써, Least Resently Used의 약자입니다. 즉 페이지 부재가 발생했을 경우 가장 오랫동안 사용되지 않은 페이지를 제거하는 알고리즘입니다.
이 알고리즘의 기본 가설은 가장 오랫동안 이용되지 않은 페이지는 앞으로도 사용할 확률이 적다는 가정하에 페이지 교체가 진행됩니다.
다음 그림을 참고해주세요.

시간	1	2	3	4	5	6	7	8
들어오는 페이지	B	C	B	A	E	B	C	E
메모리	B	B	B	B	B	B	B	B
		C	C	C	E	E	E	E
				A	A	A	C	C
페이지교체 발생여부	F	F		F	F		F	

메모리의 크기가 i로 주어지고 들어올 페이지들이 n으로 주어졌을 때, 전체 실행시간을 구해주세요.

만약 스택 안에 같은 스케줄이 있다면 **hit** 이라고 하며 실행시간은 **1초** 입니다. 스택 안에 스케줄이 없다면 **miss** 라고 하며 실행시간은 **6초** 입니다.

페이지

Aa 페이지	# 페이지 프레임	# 실행시간
BCBAEBCE	3	33
ABCABCABC	3	24
ABCEDF	0	36

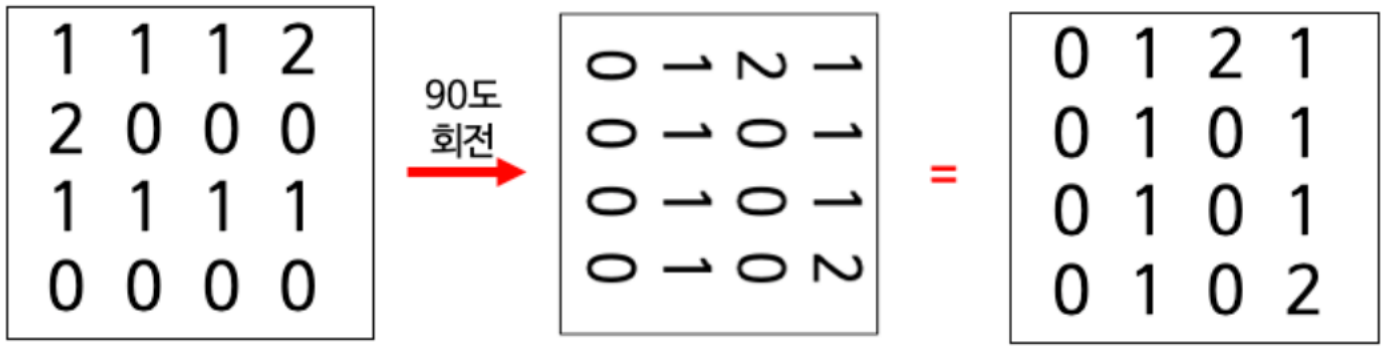
COUNT 3

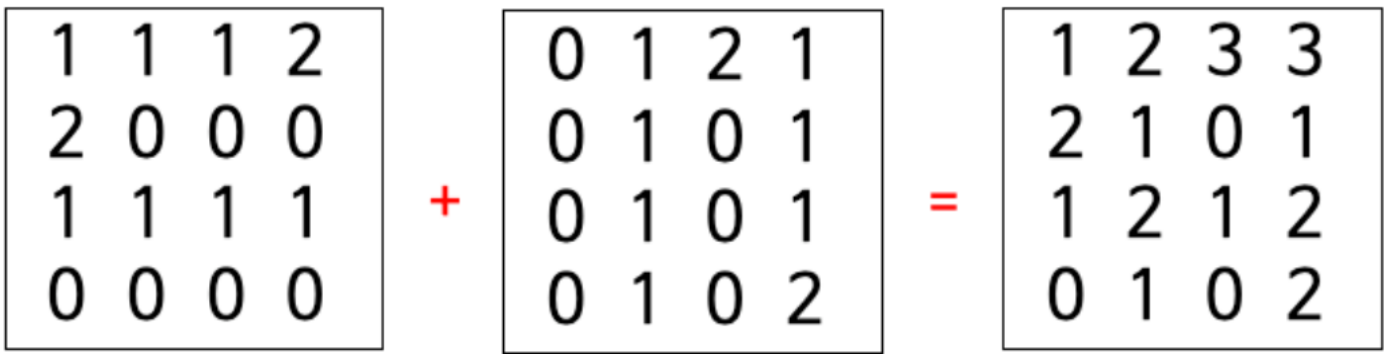
- 예제 1번을 보면 페이지 프레임의 개수는 3개이고 스케줄은 'BCBAEBCE' 입니다. 5번의 miss를 기록하므로 **5번** * **6초** = **30초**가 되고 3번의 hit을 기록하므로 **3번** * **1초** = **3초**입니다. 2개를 합한 값이 실행시간이므로, 33초가 됩니다.

문제95 : 도장찍기

빈 종이에 stamp 모양으로 생긴 도장을 90*k 도로 회전하며 찍습니다.
도장은 N*N 크기이며 각 도장이 찍히는 부분은 1 이상의 자연수와 도장이 찍히지 않는 0으로 이루어져 있습니다.

도장의 크기 N과,
종이에 찍힌 도장 횟수를 표현한 stmp 배열과,
얼만큼 회전할 것인지를 알려주는 회전수 k를 입력받았을 때
각 위치에서 몇 번 도장이 찍혔는지 그 결과값을 출력하세요





입력

```
도장 = [
[1,1,1,2],
[2,0,0,0],
[1,1,1,1],
[0,0,0,0]]

회전 = 1
```

출력

```
[[1, 2, 3, 3], [2, 1, 0, 1], [1, 2, 1, 2], [0, 1, 0, 2]]
```

JavaScript

문제96 : 넓은 텃밭 만들기!

수연이는 밭농사를 시작하기로 마음을 먹었다. 집 앞 텃밭을 만들기로 하고 돌들을 제거하고 있는데 매우 큰 바위는 옮기지 못해 고심하고 있다.

이에 수연이는 다음과 같은 규칙을 정한다.

1. 바위를(바위는 '1'로 표기한다.) 피해 텃밭을 만들되 정사각형 모양으로 텃밭을 만든다.
2. 텃밭은 가장 넓은 텃밭 1개만 만들고 그 크기를 반환한다.
3. 만든 텃밭은 모두 '#'으로 처리한다.

<입출력 예시>

입력

```
0 0 0 0 0
0 1 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
```

출력

```
3 X 3

0 0 # # #
0 1 # # #
0 1 # # #
0 0 1 0 0
0 0 0 1 0
```

입력

```
0 0 0 1 0
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 0 1 0
```

출력

```
2 X 2

# # 0 1 0
# # 0 0 0
1 0 1 0 0
0 0 1 0 0
1 0 0 1 0
```

```javascript
/* ***********문제********** */
const 텃밭 = []; //입력받은 텃밭 리스트
let 가꾼텃밭 = []; //텃밭을 가꾼 후 저장된 리스트

// 코드를 작성해주세요

console.log(가꾼텃밭);
```

JavaScript

문제97 : 택배 배달

n 명의 택배 배달원은 쌓인 택배를 배달해야 합니다.
각 택배는 접수된 순서로 배달이 되며 택배마다 거리가 주어집니다.
거리1당 1의 시간이 걸린다고 가정하였을 때 모든 택배가 배달 완료될 시간을 구하세요.

1. 모든 택배의 배송 시간 1 이상이며 배달지에 도착하고 돌아오는 왕복 시간입니다.
2. 택배는 물류창고에서 출발합니다.
3. 배달을 완료하면 다시 물류창고로 돌아가 택배를 받습니다.
4. 물류창고로 돌아가 택배를 받으면 배달을 시작합니다.
5. 택배를 상차할 때 시간은 걸리지 않습니다.

입력은 배달원의 수와 택배를 배달하는 배달 시간이 주어집니다.

ex) 배달원이 3명이고 각 거리가 [1,2,1,3,3,3]인 순서로 들어오는 경우

```javascript
function solution(n,l){
  //코드 작성
}

const 배달원 = 3;
const 배달시간 = [1,2,1,3,3,3];


console.log(solution(배달원, 배달시간));
// 출력값 = 5
```

배달원1	배달원2	배달원3	시간
1	2	1	택배 상차
0	1	0	1초 경과
3	1	3	택배 상차
2	0	2	1초 경과
2	3	2	택배 상차
1	2	1	1초 경과
0	1	0	1초 경과
0	0	0	1초 경과, 배달 완료
총 경과시간 = 5초			

문제98 : 청길이의 패션 대회

패션의 선도주자 청길이는 패션의 발전을 위해 패션쇼를 방문해 유니크한 아이템을 조사하기로 하였습니다.
청길이는 입장하는 사람들의 패션에서 처음 보는 아이템 만을 기록합니다.
이때 청길이의 기록에서 아래 규칙에 맞게 배열로 출력해 주세요.

1. 청길이는 각 옷의 종류를 정수로 기록해 놓습니다.
   ex) 입력은 "1번: 3,1 2번: 4 3번: 2,1,3 4번: 2,1,3,4" 형태의 문자열입니다.
2. 기록은 청길이가 번호 순서로 유니크한 옷의 번호를 적습니다.
3. 유니크한 옷은 기록된 순서로 추출되고 출력됩니다.
   ex) 출력은 [3,1,4,2]입니다.

입출력예시

입력
```
"1번: 4,2,3 2번: 3 3번: 2,3,4,1 4번: 2,3"
```

출력
```
[4, 2, 3, 1]
```

입력
```
"1번: 3,1 2번: 4 3번: 2,1,3 4번: 2,1,3,4"
```

출력
```
[3, 1, 4, 2]
```

JavaScript

문제99 : 토끼들의 행진

토끼들이 징검다리를 건너려고 합니다. 하지만 돌이 부실해서 몇 번 건너지 못할 것 같습니다.
대기 중인 토끼들의 통과 여부를 배열에 담아 출력해 주세요.

1. 각 돌들이 얼마나 버틸 수 있는지 배열로 주어집니다.

2. 각 토끼가 착지할 때마다 돌의 내구도는 1씩 줄어듭니다.
   ex) [1,2,1,4] 각 돌마다 1마리 2마리 1마리 4마리의 착지를 버틸 수 있습니다.

3. 토끼들은 점프력이 각자 다릅니다.
   ex) [2,1] 첫 번째 토끼는 2칸씩, 두 번째 토끼는 1칸씩 점프합니다.

4. 각 토끼들은 순서대로 다리를 건넙니다.

```
입력
돌의내구도 = [1, 2, 1, 4]
토끼의점프력 = [2, 1]

출력
['pass', 'pass']

입력
돌의내구도 = [1, 2, 1, 4, 5, 2]
토끼의점프력 = [2, 1, 3, 1]

출력
['pass', 'pass', 'fail', 'fail']
```

JavaScript

문제100 : 퍼즐게임

N x M으로 이루어진 아래와 같은 공간에 퍼즐이 쌓여져 있습니다.

퍼즐을 맞추기 위해서는 반드시 맨 오른쪽 줄로 이동시켜 줘야 합니다.
만약 종류가 같은 퍼즐이 연속될 시에 점수가 추가되며 그 퍼즐은 사라집니다.

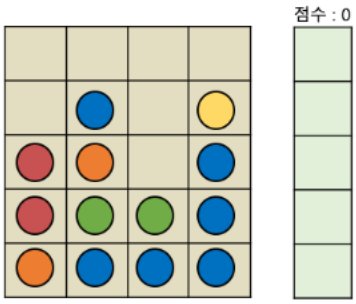
점수는 다음과 같습니다.
- 파란색 공 : 1점
- 빨간색 공 : 2점
- 노란색 공 : 3점
- 초록색 공 : 4점
- 주황색 공 : 5점

점수는 공의 개수만큼 추가됩니다
예를 들어 빨간색 공이 2개 연속되어 없어졌을 경우 2*2 = 4점입니다.

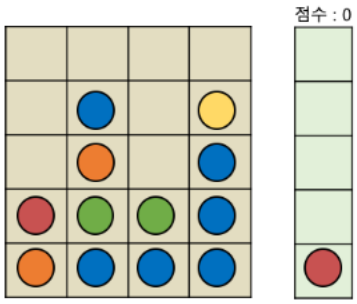
게임 플레이어는 게임이 시작되면 어떤 퍼즐을 이동할 것인지 모두 작성합니다.
만약 비어있는 곳을 선택하게 된다면 점수가 1점 감소하며 그대로 진행합니다.
위 규칙에 맞는 점수를 리턴하는 함수를 작성하세요.

예를 들어 입력이 "1 1 1 1 3 3 3"일 경우,

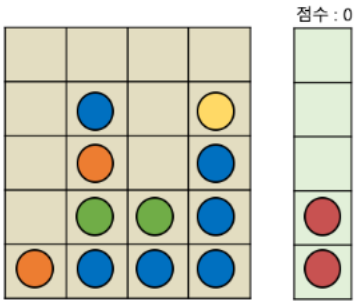
입력이 "1 1 1 1 3 3 3" 일 때



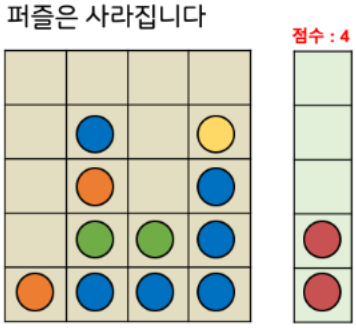
1) 1번째 줄의 퍼즐을 옮깁니다.



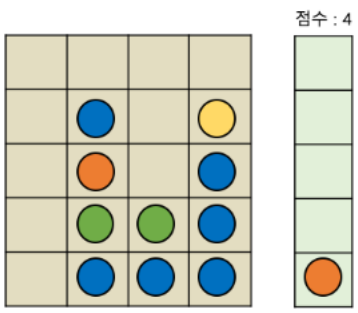
2) 1번째 줄의 퍼즐을 옮깁니다.



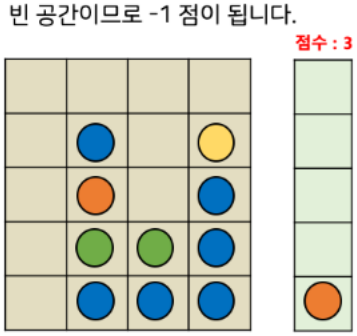
연속되었으므로 4점이 추가되고,
퍼즐은 사라집니다



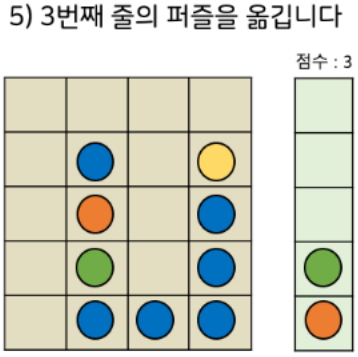
3) 다시 1번째 줄의 퍼즐을 옮깁니다.



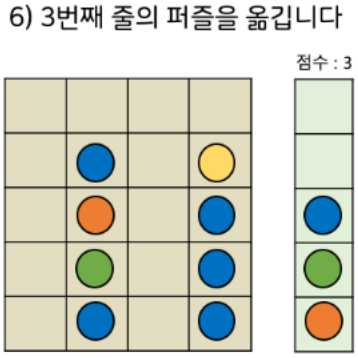
4) 다시 1번째 줄의 퍼즐을 옮기지만
빈 공간이므로 -1 점이 됩니다.



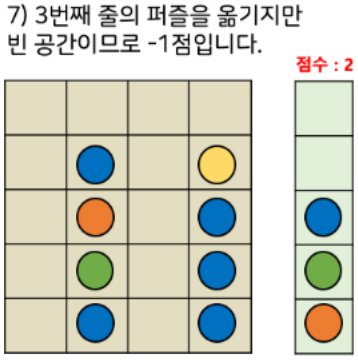
5) 3번째 줄의 퍼즐을 옮깁니다



6) 3번째 줄의 퍼즐을 옮깁니다



7) 3번째 줄의 퍼즐을 옮기지만
빈 공간이므로 -1점입니다.



총 점수는 2점으로 2를 출력해야 합니다.

입력
퍼즐판 = [[0,0,0,0],[0,1,0,3],[2,5,0,1],[2,4,4,1],[5,1,1,1]]
조작 = [1,1,1,1,3,3,3]

출력
2

JavaScript

자바스크립트
JAVA SCRIPT
100제

이런 분들을 위한 책입니다!

이 책은 다양한 유형을 살펴보며 자바스크립트 활용을 통해 문제를 풀 수 있게 하고, 문제 해결능력을 키우는데 목적을 두었습니다.

자바스크립트에 입문한 사람, 문법은 알고 있지만 문제해결을 어떤 식으로 해야하는지 모르는 사람들을 위한 책입니다. 이론과 개념 위주의 책을 통해 기본을 쌓았다면, 스스로 부족한 부분을 점검하는 과정이 필요합니다.
JavaScript 100제를 통해 다시 기초를 다지고 문제 해결능력을 기를 수 있습니다.

CODE
FESTIVAL
JEJU CODING BASE CAMP