

알고리즘 스터디 2주차

김건우 2022. 09. 27.

백준 2675 문자열 반복

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	143935	72659	62234	50.642%

문제

문자열 S 를 입력받은 후에, 각 문자를 R 번 반복해 새 문자열 P 를 만든 후 출력하는 프로그램을 작성하시오. 즉, 첫 번째 문자를 R 번 반복하고, 두 번째 문자를 R 번 반복하는 식으로 P 를 만들면 된다. S 에는 QR Code "alphanumeric" 문자만 들어있다.

QR Code "alphanumeric" 문자는 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ\$%*+-./: 이다.

입력

첫째 줄에 테스트 케이스의 개수 T ($1 \leq T \leq 1,000$)가 주어진다. 각 테스트 케이스는 반복 횟수 R ($1 \leq R \leq 8$), 문자열 S 가 공백으로 구분되어 주어진다. S 의 길이는 적어도 1이며, 20글자를 넘지 않는다.

출력

각 테스트 케이스에 대해 P 를 출력한다.

예제 입력 1 [복사](#)

```
2
3 ABC
5 /HTP
```

예제 출력 1 [복사](#)

```
AAABBBCCC
/////HHHHHTTTTTPPPPP
```

일반적인 풀이(윤아)

```
n = int(input())
for i in range(n):
    a, b = input().split()
    str = list(b)
    for j in range(len(b)):
        for k in range(int(a)):
            print(b[j], end='')
```

파이썬스러운 풀이(문자열)

```
T = int(input())  
for _ in range(T):  
    n, S = input().split()  
    print(''.join(map(lambda x: x * int(n), S)))
```

백준 10817 세수

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	117381	55021	47981	48.290%

문제

세 정수 A, B, C가 주어진다. 이때, 두 번째로 큰 정수를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 세 정수 A, B, C가 공백으로 구분되어 주어진다. ($1 \leq A, B, C \leq 100$)

출력

두 번째로 큰 정수를 출력한다.

예제 입력 1 [복사](#)

20 30 10

예제 출력 1 [복사](#)

20

일반적인 풀이

```
a, b, c = map(int, input().split(' '))
```

```
if(a >= b):  
    if(a >= c):  
        if(b >= c):  
            print(b)  
        else:  
            print(c)  
    else:  
        print(a)  
else:  
    if(b >= c):  
        if(a >= c):  
            print(a)  
        else:  
            print(c)  
    else:  
        print(b)
```

정렬을 이용한 풀이

```
num_list = list(map(int, input().split()))  
num_list.sort()  
print(num_list[1])
```

백준 1789 수들의 합

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	34912	14218	12167	42.327%

문제

서로 다른 N 개의 자연수의 합이 S 라고 한다. S 를 알 때, 자연수 N 의 최댓값은 얼마일까?

입력

첫째 줄에 자연수 S ($1 \leq S \leq 4,294,967,295$)가 주어진다.

출력

첫째 줄에 자연수 N 의 최댓값을 출력한다.

예제 입력 1 [복사](#)

200

예제 출력 1 [복사](#)

19

일반적인 풀이(다현)

```
a = int(input())
```

```
sum = 0
```

```
cnt=0
```

```
n=1
```

```
while(1):
```

```
    sum = sum+n
```

```
    n += 1
```

```
    cnt += 1
```

```
    if(sum>a):
```

```
        print(cnt-1)
```

```
        break
```

수학적인 풀이

```
import math

S = int(input())
print(int((-1 + math.sqrt(1+8*S)) / 2))
```

백준 10102 개표

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	6880	4592	4324	68.041%

문제

A와 B가 한 오디션 프로의 결승전에 진출했다. 결승전의 승자는 심사위원의 투표로 결정된다.

심사위원의 투표 결과가 주어졌을 때, 어떤 사람이 우승하는지 구하는 프로그램을 작성하시오.

입력

입력은 총 두 줄로 이루어져 있다. 첫째 줄에는 심사위원의 수 V ($1 \leq V \leq 15$)가 주어지고, 둘째 줄에는 각 심사위원이 누구에게 투표했는지가 주어진다. A와 B는 각각 그 참가자를 나타낸다.

출력

- A가 받은 표가 B보다 많은 경우에는 A
- B가 받은 표가 A보다 많은 경우에는 B
- 같은 경우에는 Tie

를 출력한다.

예제 입력 1 [복사](#)

```
6
ABBABB
```

예제 출력 1 [복사](#)

```
B
```

일반적인 풀이(반복문)

```
V = int(input())  
vote = input()
```

```
A = 0  
B = 0
```

```
for v in vote:  
    if v == 'A':  
        A += 1  
    else:  
        B += 1
```

```
if A > B:  
    print('A')  
elif A == B:  
    print('Tie')  
else:  
    print('B')
```

문자열 메소드 활용 풀이

```
V = int(input())  
vote = input()
```

```
A = vote.count('A')  
B = vote.count('B')
```

```
if A > B:  
    print('A')  
elif A == B:  
    print('Tie')  
else:  
    print('B')
```

백준 10988 팰린드롬인지 확인하기

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	17809	11812	10201	67.736%

문제

알파벳 소문자로만 이루어진 단어가 주어진다. 이때, 이 단어가 팰린드롬인지 아닌지 확인하는 프로그램을 작성하시오.

팰린드롬이란 앞으로 읽을 때와 거꾸로 읽을 때 똑같은 단어를 말한다.

level, noon은 팰린드롬이고, baekjoon, online, judge는 팰린드롬이 아니다.

입력

첫째 줄에 단어가 주어진다. 단어의 길이는 1보다 크거나 같고, 100보다 작거나 같으며, 알파벳 소문자로만 이루어져 있다.

출력

첫째 줄에 팰린드롬이면 1, 아니면 0을 출력한다.

예제 입력 1 [복사](#)

```
level
```

예제 출력 1 [복사](#)

```
1
```

일반적인 풀이

```
w = input()
x = 1
for i in range(len(w)//2):
    if w[i] != w[-i-1]:
        print(0)
        x = 0
        break
if x == 1:
    print(x)
```

날먹

```
w = input()
if w == w[::-1]:
    print(1)
else:
    print(0)
```


백준 11557

Yangjojang of The year

입력

입력의 첫 줄에는 테스트 케이스의 숫자 T가 주어진다.

매 입력의 첫 줄에는 학교의 숫자 정수 N($1 \leq N \leq 100$)이 주어진다.

이어서 N줄에 걸쳐 학교 이름 S($1 \leq |S| \leq 20$, S는 공백없는 대소문자 알파벳 문자열)와 해당 학교가 지난 한 해동안 소비한 술의 양 L($0 \leq L \leq 10,000,000$)이 공백으로 구분되어 정수로 주어진다.

같은 테스트 케이스 안에서 소비한 술의 양이 같은 학교는 없다고 가정한다.

출력

각 테스트 케이스마다 한 줄에 걸쳐 술 소비가 가장 많은 학교의 이름을 출력한다.

예제 입력 1 [복사](#)

```
2
3
Yonsei 10
Korea 10000000
Ewha 20
2
Yonsei 1
Korea 10000000
```

예제 출력 1 [복사](#)

```
Korea
Korea
```

정렬

```
T = int(input())
for _ in range(T):
    N = int(input())
    alcohol = []

    for _ in range(N):
        S, L = input().split()
        alcohol.append([S, int(L)])

    alcohol = sorted(alcohol, key = lambda x: x[1])
    print(alcohol[-1][0])
```

정렬(더 짧게)

```
for _ in range(int(input())):  
    N = int(input())  
    alcohol = [list(input().split()) for x in range(N)]  
    alcohol = sorted(alcohol, key = lambda x: x[1])  
    print(alcohol[-1][0])
```

정렬(더 더 짧게)

```
for _ in range(int(input())):  
    N = int(input())  
    alcohol = sorted([list(input().split()) for x in range(N)], key=lambda x: x[1])  
    print(alcohol[-1][0])
```

문제풀이를 위한 복집도

복잡도(Complexity)

- 알고리즘의 성능을 나타내는 척도
 - 시간 복잡도: 특정한 크기의 입력에 대하여 알고리즘의 수행 시간
 - 공간 복잡도: 특정한 크기의 입력에 대하여 알고리즘의 메모리 사용량
- 동일한 기능을 수행하는 알고리즘이라면, 일반적으로 복잡도가 낮을수록 좋음

복잡도(Complexity)

- 알고리즘의 성능을 나타내는 척도
 - 시간 복잡도: 특정한 크기의 입력에 대하여 알고리즘의 수행 시간
 - 공간 복잡도: 특정한 크기의 입력에 대하여 알고리즘의 메모리 사용량
- 동일한 기능을 수행하는 알고리즘이라면, 일반적으로 복잡도가 낮을수록 좋음

빅오 표기법(Big-O Notation)

- 가장 영향력이 큰 항만을 고려하는 표기법
 - 함수의 상한만을 나타냄
- 예를 들어 연산 횟수가 $3N^3 + 5N^2 + 1000$ 일 때,
 - $O(N^3)$ 라고 표현함

빅오 표기법(Big-O Notation)

- 위로 갈 수록 좋음(빠름)

순위	명칭
$O(1)$	상수 시간
$O(\log N)$	로그 시간
$O(N)$	선형 시간
$O(N \log N)$	로그 선형 시간
$O(N^2)$	이차 시간
$O(N^3)$	삼차 시간
$O(2^N)$	지수 시간

파이썬에서 Tip

- 파이썬에서 $O(20,000,000)$ 가 대략 1초

알고리즘 문제 해결 과정

1. 지문 읽기 및 컴퓨터적 사고
2. 요구사항(복잡도) 분석
3. 아이디어 찾기
4. 코드 설계

그리디 알고리즘

그리디 알고리즘

- 현재 상황에서 지금 당장 좋은 것만 고르는 방법
- 일반적인 그리디 알고리즘은 문제를 풀기 위한 최소한의 아이디어를 떠올릴 수 있는 능력을 요구
- 그리디 해법은 그 정당성 분석이 중요
 - 단순히 가장 좋아 보이는 것을 반복적으로 선택해도 최적의 해를 구할 수 있는지 검토

그리디 알고리즘

- 일반적인 상황에서 그리디 알고리즘은 최적의 해를 보장할 수 없을 때가 많음

백준 5585 거스름돈

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	32038	20441	17467	63.620%

문제

타로는 자주 JOI잡화점에서 물건을 산다. JOI잡화점에는 잔돈으로 500엔, 100엔, 50엔, 10엔, 5엔, 1엔이 충분히 있고, 언제나 거스름돈 개수가 가장 적게 잔돈을 준다. 타로가 JOI잡화점에서 물건을 사고 카운터에서 1000엔 지폐를 한장 냈을 때, 받을 잔돈에 포함된 잔돈의 개수를 구하는 프로그램을 작성하시오.

입력

입력은 한줄로 이루어져있고, 타로가 지불할 돈(1 이상 1000미만의 정수) 1개가 쓰여져있다.

출력

제출할 출력 파일은 1행으로만 되어 있다. 잔돈에 포함된 매수를 출력하시오.

예제 입력 1 [복사](#)

380

예제 출력 1 [복사](#)

4

그리디 알고리즘

```
a = 1000 - int(input())
b = [500, 100, 50, 10, 5, 1]
count = 0
for i in b:
    count += a // i
    a %= i
print(count)
```


거스름돈: 정당성 분석

- 가장 큰 화폐 단위부터 돈을 거슬러 주는 것이 최적의 해를 보장하는 이유?
 - 가지고 있는 동전이 큰 단위가 항상 작은 단위의 배수이므로 작은 단위의 동전들을 종합해 다른 해가 나올 수 없기 때문
- 만약에 800원을 거슬러 주어야 하는데 화폐단위가 500원, 400원, 100원 이라면?
- 그리디 알고리즘 문제에서는 이처럼 문제 풀이를 위한 최소한의 아이디어를 떠올리고 이것이 정당한지 검토할 수 있어야 함