

C++

코딩테스트

누적합 & 구현 문제들

01

하나,

구현 문제

패션왕 신해빈
좋은 단어



02

두울,

누적합 문제

수열
주몽

01

하나,

구현 문제

곱셈
좋은 단어

02

두울,

누적합 문제

수열
주몽

구현 문제

구현: 머릿속에 있는 알고리즘을 소스코드로 바꾸는 과정

- 풀이를 떠올리는 것은 쉽지만 소스코드로 옮기기 어려운 문제
 - 코딩 테스트에서는 구현이 중심이 되는 문제가 자주 출제된다.
 - ex) 완전 탐색, 시뮬레이션 유형

구현 문제의 종류

- 알고리즘은 간단한데 코드가 치나칠 만큼 길어지는 문제
- 특정 소수점 자리까지 출력해야 하는 문제
- 문자열이 입력으로 주어졌을 때 한 문자 단위로 끊어서 리스트에 파싱 해야 하는 문제
- 대체로 사소한 조건 설정이 많은 문제일수록 코드로 구현하기 까다롭다.

구현 문제를 풀기 위한 선행 조건

- 프로그래밍 문법의 정확한 숙지
- 라이브러리 사용 경험

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	128 MB	58568	15668	11514	25.989%

문제

자연수 A를 B번 곱한 수를 알고 싶다. 단 구하려는 수가 매우 커질 수 있으므로 이를 C로 나눈 나머지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 A, B, C가 빈 칸을 사이에 두고 순서대로 주어진다. A, B, C는 모두 2,147,483,647 이하의 자연수이다.

출력

첫째 줄에 A를 B번 곱한 수를 C로 나눈 나머지를 출력한다.

예제 입력 1 [복사](#)

```
10 11 12
```

예제 출력 1 [복사](#)

```
4
```

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	128 MB	58568	15668	11514	25.989%

문제

자연수 A를 B번 곱한 수를 알고 싶다. 단 구하려는 수가 매우 커질 수 있으므로 이를 C로 나눈 나머지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 A, B, C가 빈 칸을 사이에 두고 순서대로 주어진다. A, B, C는 모두 2,147,483,647 이하의 자연수이다.

출력

첫째 줄에 A를 B번 곱한 수를 C로 나눈 나머지를 출력한다.

예제 입력 1 [복사](#)

```
10 11 12
```

예제 출력 1 [복사](#)

```
4
```

1) A, B, C의 크기
2) 시간 제한

1629번 : 곱셈

곱셈 성공
1 실버 I

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	128 MB	58568	15668	11514	25.989%

문제

자연수 A를 B번 곱한 수를 알고 싶다. 단 구하려는 수가 매우 커질 수 있으므로 이를 C로 나눈 나머지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 A, B, C가 빈 칸을 사이에 두고 순서대로 주어진다. A, B, C는 모두 2,147,483,647 이하의 자연수이다.

출력

첫째 줄에 A를 B번 곱한 수를 C로 나눈 나머지를 출력한다.

예제 입력 1 복사

```
10 11 12
```

예제 출력 1 복사

```
4
```

1) A, B, C의 크기

- int형이 아닌 long long을 사용함
- modular 연산을 통해 overflow를 방지

$$(a * b) \% C = (a \% C * b \% C) \% C$$

2) 시간 제한

- O(n)의 알고리즘을 사용하면 최악의 경우 21억번의 연산, 즉 21초가 소요됨.
- O(logN)의 알고리즘을 사용하면 $\log(2^{31})=31$, 즉 최악의 경우에도 31번의 연산을 필요로함
- O(logN)의 알고리즘인 분할 정복을 사용

1629번 : 곱셈

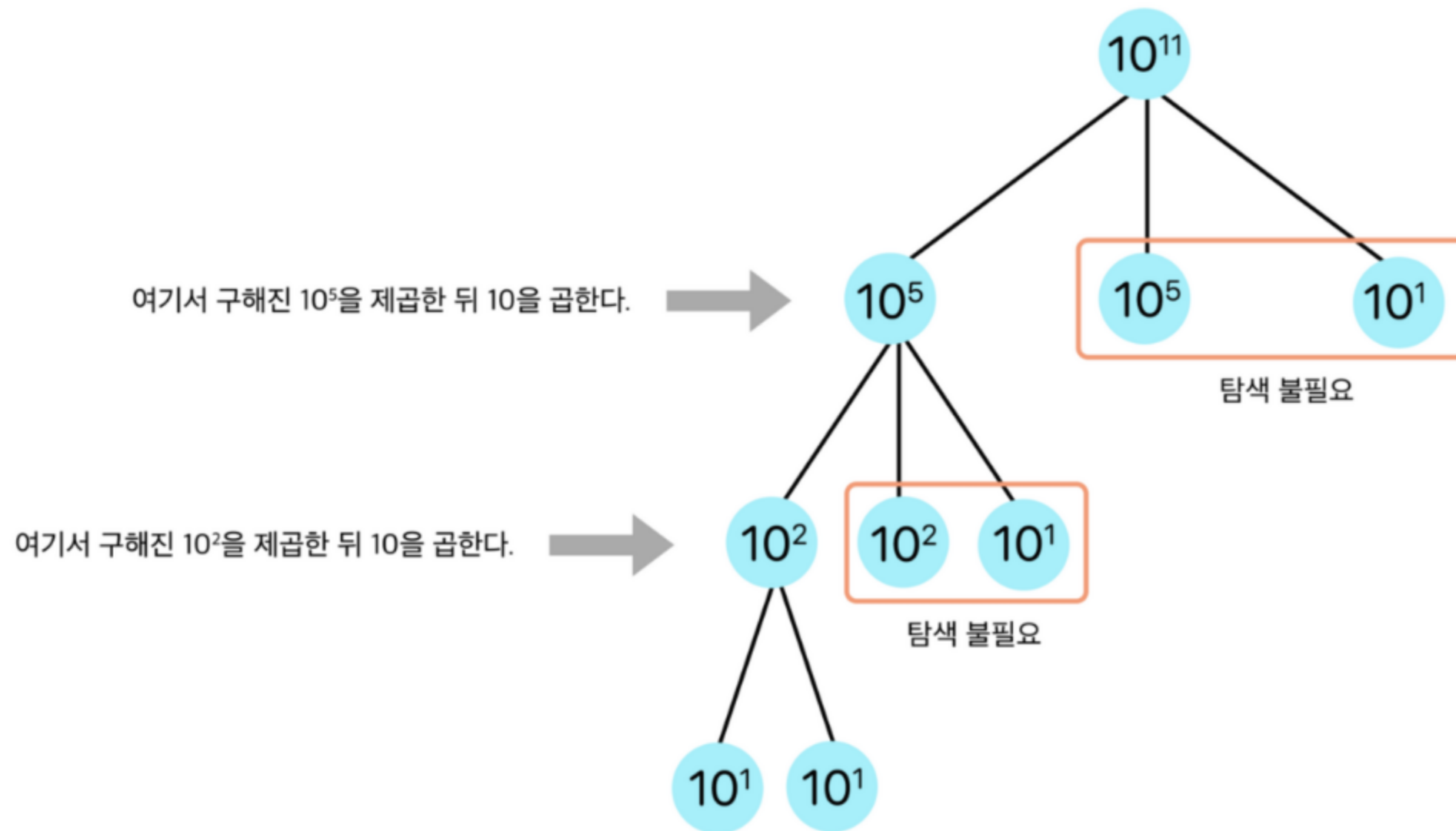
분할정복

짝수인 경우

$$\begin{aligned} a^8 &= a^4 * a^4 \\ &= (a^2 * a^2) * (a^2 * a^2) \\ &= ((a^1 * a^1) * (a^1 * a^1)) * ((a^1 * a^1) * (a^1 * a^1)) \end{aligned}$$

홀수인 경우

$$\begin{aligned} a^9 &= a^4 * a^4 * a \\ &= (a^2 * a^2) * (a^2 * a^2) * a \\ &= ((a^1 * a^1) * (a^1 * a^1)) * ((a^1 * a^1) * (a^1 * a^1)) * a \end{aligned}$$



1629번 : 곱셈

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 long long n;
6 int a,b,c;
7
8 long long Do(int t1, int t2) //재귀함수
9 {
10     if(t2==1) return t1; //더 이상 분할할 수 없는 경우
11     else
12     {
13         long long temp=Do(t1, t2/2);
14
15         if(t2%2) return ((temp*temp)%c*t1)%c; //1) Modular연산을 통해 overflow 방지
16                                                //2) 홀수인 경우 t1을 한 번 더 곱해줌
17         else return (temp*temp)%c;
18     }
19
20 }
21 int main(void)
22 {
23     cin>>a>>b>>c;
24
25     cout<<Do(a%c,b);
26 }
27
```

3986번: 좋은 단어

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	10295	5209	4319	51.669%

문제

이번 계절학기에 심리학 개론을 수강 중인 평석이는 오늘 자정까지 보고서를 제출해야 한다. 보고서 작성이 너무 지루했던 평석이는 노트북에 엮드려서 꾸벅꾸벅 졸다가 제출 마감 1시간 전에 깨고 말았다. 안타깝게도 자는 동안 키보드가 잘못 눌러서 보고서의 모든 글자가 A와 B로 바뀌어 버렸다! 그래서 평석이는 보고서 작성을 때려치우고 보고서에서 '좋은 단어'나 세보기로 마음 먹었다.

평석이는 단어 위로 아치형 곡선을 그어 같은 글자끼리(A는 A끼리, B는 B끼리) 쌍을 짓기로 하였다. 만약 선끼리 교차하지 않으면서 각 글자를 정확히 한 개의 다른 위치에 있는 같은 글자와 짝 지을수 있다면, 그 단어는 '좋은 단어'이다. 평석이가 '좋은 단어' 개수를 세는 것을 도와주자.

입력

첫째 줄에 단어의 수 N 이 주어진다. ($1 \leq N \leq 100$)

다음 N 개 줄에는 A와 B로만 이루어진 단어가 한 줄에 하나씩 주어진다. 단어의 길이는 2와 100,000사이이며, 모든 단어 길이의 합은 1,000,000을 넘지 않는다.

3986번: 좋은 단어

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	10295	5209	4319	51.669%

문제

이번 계절학기에 심리학 개론을 수강 중인 평석이는 오늘 자정까지 보고서를 제출해야 한다. 보고서 작성이 너무 지루했던 평석이는 노트북에 엮드려서 꾸벅꾸벅 졸다가 제출 마감 1시간 전에 깨고 말았다. 안타깝게도 자는 동안 키보드가 잘못 눌러서 보고서의 모든 글자가 A와 B로 바뀌어 버렸다! 그래서 평석이는 보고서 작성을 때려치우고 보고서에서 '좋은 단어'나 세보기로 마음 먹었다.

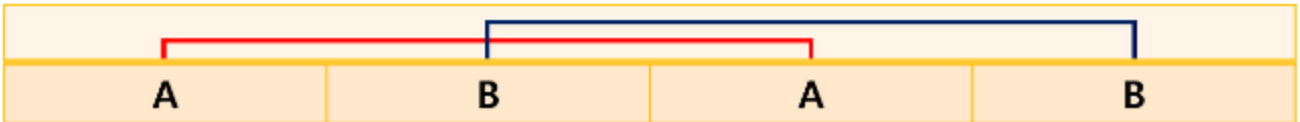
평석이는 단어 위로 아치형 곡선을 그어 같은 글자끼리(A는 A끼리, B는 B끼리) 쌍을 짓기로 하였다. 만약 선끼리 교차하지 않으면서 각 글자를 정확히 한 개의 다른 위치에 있는 같은 글자와 짝 지을수 있다면, 그 단어는 '좋은 단어'이다. 평석이가 '좋은 단어' 개수를 세는 것을 도와주자.

입력

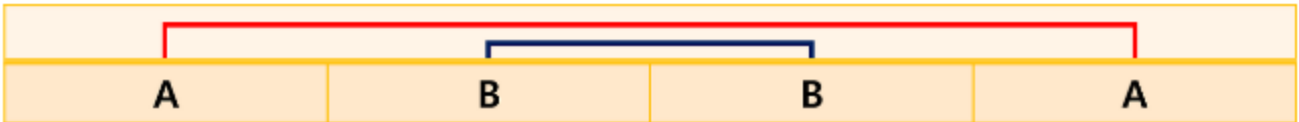
첫째 줄에 단어의 수 N 이 주어진다. ($1 \leq N \leq 100$)

다음 N 개 줄에는 A와 B로만 이루어진 단어가 한 줄에 하나씩 주어진다. 단어의 길이는 2와 100,000사이이며, 모든 단어 길이의 합은 1,000,000을 넘지 않는다.

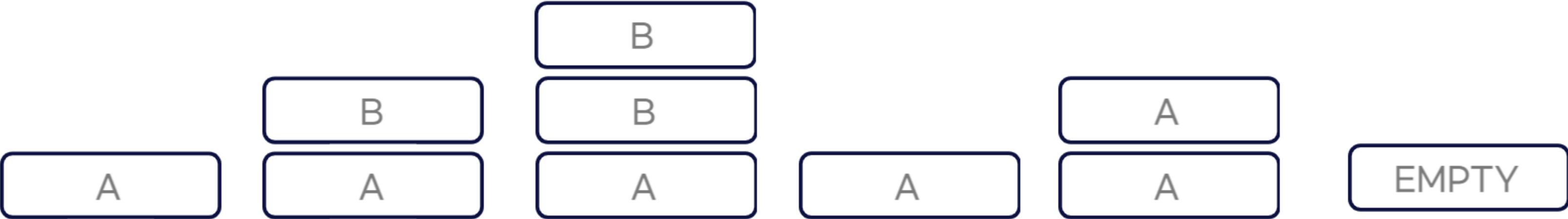
좋은 단어 X



좋은 단어 O



3986번: 좋은 단어



ABBA를 삽입한 경우

3986번: 좋은 단어

```
int main(void)
{
    cin>>cnt;

    for(int i=0; i<cnt;i++)
    {
        stack<int> st;
        cin>>now;

        for(int j=0; j<now.size();j++)
        {
            if(st.size()!=0&&st.top()==now[j]) st.pop(); // stack의 top과 동일하면 pop
            else st.push(now[j]);
        }

        if(st.size()==0) sum++; //모든 값이 pop됐으면 좋은 단어라고 판단
    }
}
```

01

하나,

구현 문제

곱셈
좋은 단어

02

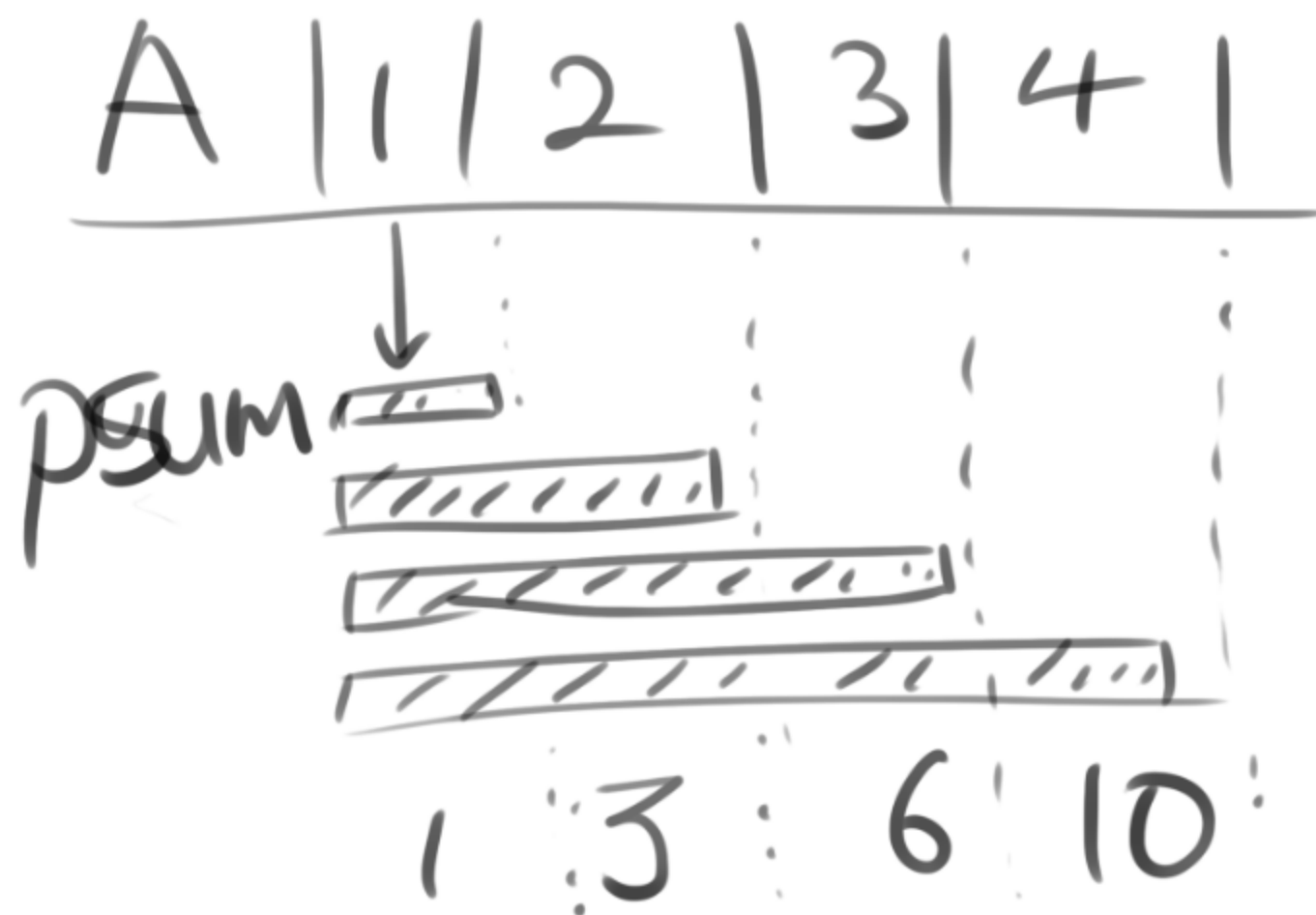
두울,

누적합 문제

수열
주몽

누적합

말 그대로 구간의 누적합을 구하는 문제



배열에 값을 저장하고 지정된 인덱스부터 하나씩 더해 가는 방식은 최악의 경우 $O(n^2)$ 의 시간복잡도를 갖기 때문에 입력의 범위가 클 때 사용할 수 없음.

-> 누적합을 사용하면 $O(N)$ 으로 해결

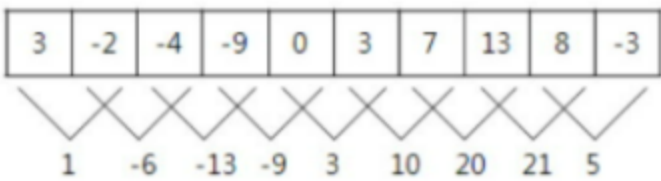
문제

매일 아침 9시에 학교에서 측정한 온도가 어떤 정수의 수열로 주어졌을 때, 연속적인 며칠 동안의 온도의 합이 가장 큰 값을 알아보고자 한다.

예를 들어, 아래와 같이 10일 간의 온도가 주어졌을 때,

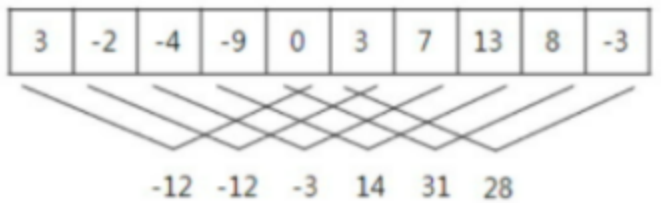
3 -2 -4 -9 0 3 7 13 8 -3

모든 연속적인 이들간의 온도의 합은 아래와 같다.



이때, 온도의 합이 가장 큰 값은 21이다.

또 다른 예로 위와 같은 온도가 주어졌을 때, 모든 연속적인 5일 간의 온도의 합은 아래와 같으며,



이때, 온도의 합이 가장 큰 값은 31이다.

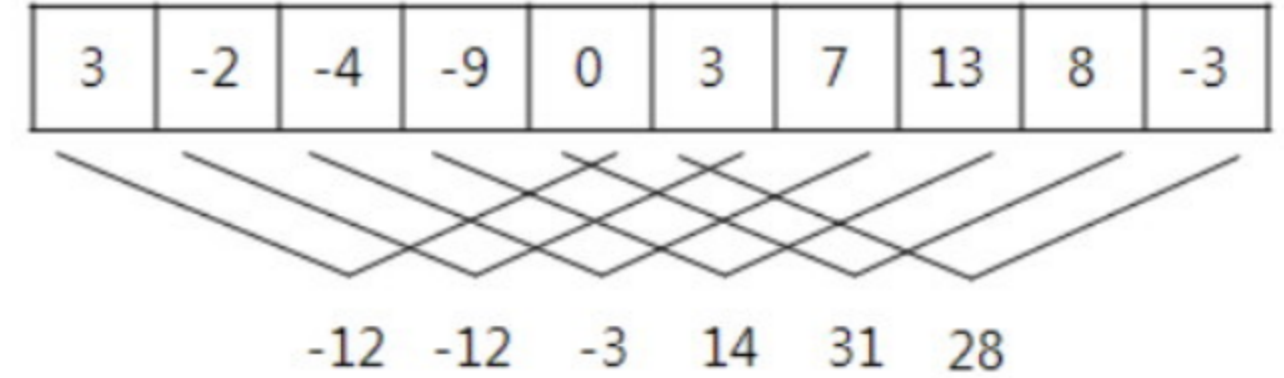
매일 측정한 온도가 정수의 수열로 주어졌을 때, 연속적인 며칠 동안의 온도의 합이 가장 큰 값을 계산하는 프로그램을 작성하시오.

입력

첫째 줄에는 두 개의 정수 N과 K가 한 개의 공백을 사이에 두고 순서대로 주어진다. 첫 번째 정수 N은 온도를 측정한 전체 날짜의 수이다. N은 2 이상 100,000 이하이다. 두 번째 정수 K는 합을 구하기 위한 연속적인 날짜의 수이다. K는 1과 N 사이의 정수이다. 둘째 줄에는 매일 측정한 온도를 나타내는 N개의 정수가 빈칸을 사이에 두고 주어진다. 이 수들은 모두 -100 이상 100 이하이다.

2559번: 수열

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int cnt=0, days=0;
4 int maxValue=INT_MIN;
5 int arr[100001];
6
7 int main(void){
8     cin>>cnt>>days;
9     arr[0]=0;
10
11     for(int i=1; i<=cnt;i++)
12     {
13         cin>>arr[i];
14         arr[i]+=arr[i-1];
15     }
16
17     for(int i=days; i<=cnt;i++)
18     {
19         if(maxValue<(arr[i]-arr[i-days]))
20         {
21             maxValue=arr[i]-arr[i-days];
22         }
23     }
24
25     cout<<maxValue;
26 }
```



문제

11659번: 구간 합 구하기

$N \times N$ 개의 수가 $N \times N$ 크기의 표에 채워져 있다. $(x1, y1)$ 부터 $(x2, y2)$ 까지 합을 구하는 프로그램을 작성하시오. (x, y) 는 x 행 y 열을 의미한다.

예를 들어, $N = 4$ 이고, 표가 아래와 같이 채워져 있는 경우를 살펴보자.

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

table[][]

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

dp[][]

1	3	6	10
3	8	15	24
6	15	27	42
10	24	42	64

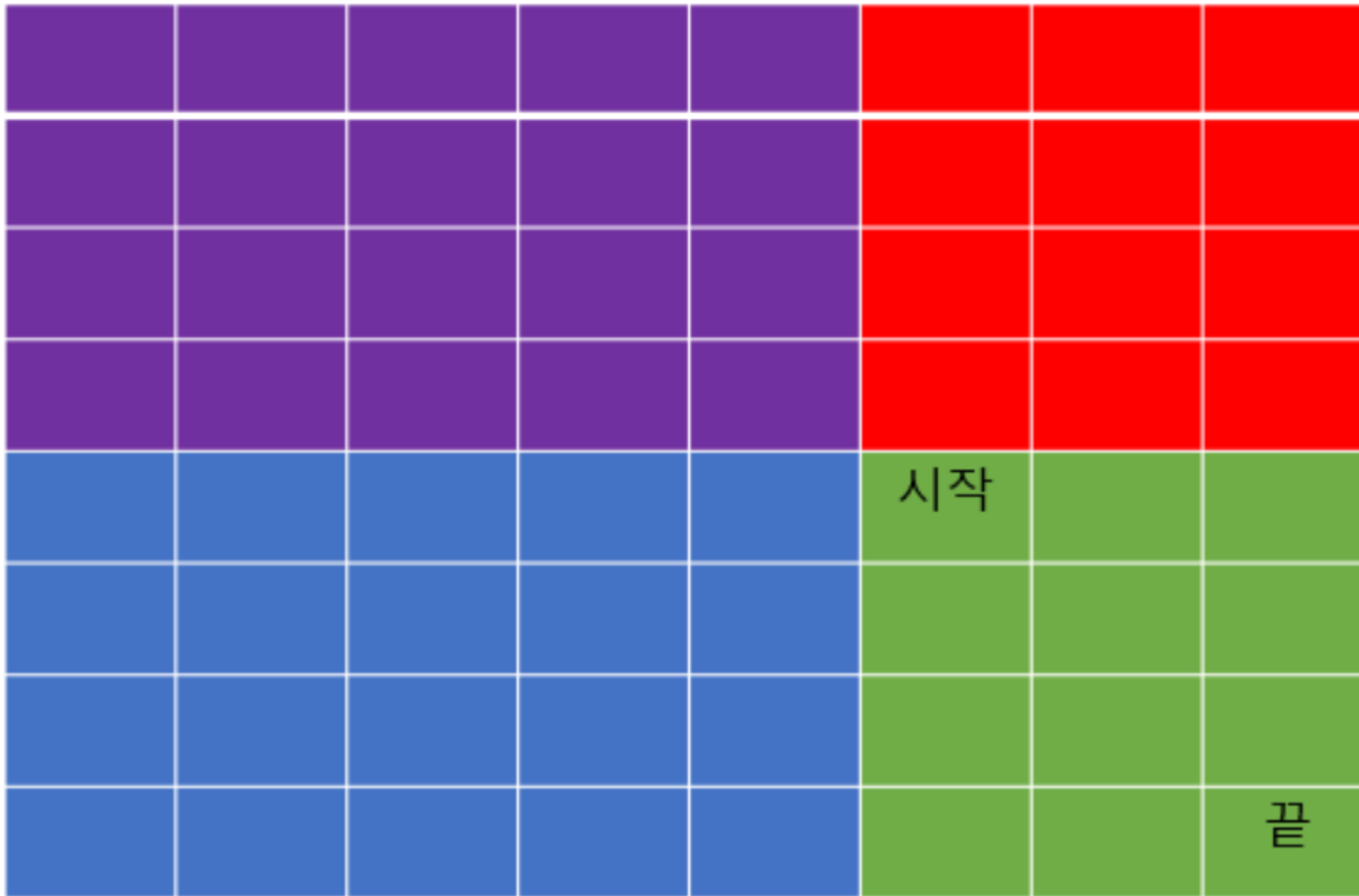
여기서 $(2, 2)$ 부터 $(3, 4)$ 까지 합을 구하면 $3+4+5+4+5+6 = 27$ 이고, $(4, 4)$ 부터 $(4, 4)$ 까지 합을 구하면 7이다.

표에 채워져 있는 수와 합을 구하는 연산이 주어졌을 때, 이를 처리하는 프로그램을 작성하시오.

입력

첫째 줄에 표의 크기 N 과 합을 구해야 하는 횟수 M 이 주어진다. ($1 \leq N \leq 1024$, $1 \leq M \leq 100,000$) 둘째 줄부터 N 개의 줄에는 표에 채워져 있는 수가 1행부터 차례대로 주어진다. 다음 M 개의 줄에는 네 개의 정수 $x1, y1, x2, y2$ 가 주어지며, $(x1, y1)$ 부터 $(x2, y2)$ 의 합을 구해 출력해야 한다. 표에 채워져 있는 수는 1,000보다 작거나 같은 자연수이다. ($x1 \leq x2, y1 \leq y2$)

11659번: 구간 합 구하기



$dp[x2][y2]$ = 전체 영역의 합

$dp[x1 - 1][y2]$ = 빨간색 영역과 보라색영역을 합친 값

$dp[x2][y1 - 1]$ = 파란색 영역과 보라색영역을 합친 값

$dp[x1 - 1][y1 - 1]$ = 보라색 영역의 합의 값

```

int main(void)
{

    scanf("%d %d", &arrSize, &loopCount);

    for (int i = 0; i < arrSize; i++)
    {
        for (int j = 0; j < arrSize; j++)
        {
            int num = 0;
            scanf("%d", &num);
            sum[i+1][j+1] = sum[i][j+1] + sum[i+1][j] - sum[i][j] + num;
        }
    }

    for (int i = 0; i < loopCount; i++)
    {
        sX = 0, sY = 0, eX = 0, eY = 0;

        scanf("%d %d %d %d", &sY, &sX, &eY, &eX);

        printf("%d\n", sum[eY][eX] - sum[sY - 1][eX] - sum[eY][sX - 1] + sum[sY - 1][sX - 1]);

    }

    return 0;
}

```

감사합니다!