

## 문제 1. 문자 찾기

한 개의 문자열을 입력받고, 특정 문자를 입력받아 해당 특정문자가 입력받은 문자열에 몇 개 존재하는지 알아내는 프로그램을 작성하세요. 대소문자를 구분하지 않습니다. 문자열의 길이는 100을 넘지 않습니다.

### ■ 입력설명

첫 줄에 문자열이 주어지고, 두 번째 줄에 문자가 주어진다. 문자열은 영어 알파벳으로만 구성되어 있습니다.

### ■ 출력설명

첫 줄에 해당 문자의 개수를 출력한다.

### ■ 입력예제 1

Computercooler

c

### ■ 출력예제 1

2

<hint>

String.charAt(i) : i 번째의 글자 한개를 char 형태로 받음

String.toUpperCase / Character.toUpperCase : 문자열을 대문자로 / 글자 하나를 대문자로

String.toCharArray : 문자열을 글자 배열로 리턴

```
Scanner scan = new Scanner(System.in);

public Quiz01() {
    // 문자를 받는다
    String str=scan.next();
    // 비교할 대상의 데이터를 받는다
    char t=scan.next().charAt(0);

    // 카운트 변수
    int answer=0;
    // 단어를 대문자로
    str=str.toUpperCase();
    // 글자 하나를 대문자로
    t=Character.toUpperCase(t);

    System.out.println(answer);
}
```

## 문제 2. 대소문자 변환

대문자와 소문자가 같이 존재하는 문자열을 입력받아 대문자는 소문자로 소문자는 대문자로 변환하여 출력하는 프로그램을 작성하세요.

### ■ 입력설명

첫 줄에 문자열이 입력된다. 문자열의 길이는 100을 넘지 않습니다. 문자열은 영어 알파벳으로만 구성되어 있습니다.

### ■ 출력설명

첫 줄에 대문자는 소문자로, 소문자는 대문자로 변환된 문자열을 출력합니다.

### ■ 입력예제 1

StuDY

### ■ 출력예제 1

sTUDy

<hint>

Character.isLowerCase(x) : 글자 한 개가 소문자인지 대문자인지 여부(소문자면 true 아니면 false)

Character.toUpperCase(x) : 글자 한 개를 대문자에서 소문자로 변환

```
public Quiz02() {  
    String answer="";  
    String str=scan.next();  
  
    for(char x : str.toCharArray()) {  
        if ( )  
            answer+=  
        else  
            answer+=  
        }  
    System.out.println(answer);  
}
```

이 부분은 무시하셔도 됩니다.

## 문제3. 문장 속 단어

한 개의 문장이 주어지면 그 문장 속에서 가장 긴 단어를 출력하는 프로그램을 작성하세요. 문장속의 각 단어는 공백으로 구분됩니다.

### ■ 입력설명

첫 줄에 길이가 100을 넘지 않는 한 개의 문장이 주어집니다. 문장은 영어 알파벳으로만 구성되어 있습니다.

### ■ 출력설명

첫 줄에 가장 긴 단어를 출력한다. 가장 긴 단어가 여러개일 경우 문장속에서 가장 앞쪽에 위치한 단어를 답으로 합니다.

### ■ 입력예제 1

it is time to study

### ■ 출력예제 1

study

<hint>

String.indexOf(x) : 문자열 안에서 특정 문자 x의 인덱스 위치를 알아낸다.

String.substring(x,y) : 특정 문자열 안에서 인덱스 x부터 y까지를 끊는다.

String.split(str) : str을 기준으로 문자열 배열을 생성한다.

String.length() : 문자열의 길이를 출력한다.

```

public Quiz03() {
    String str = scan.nextLine();
    String answer="";
    // Integer의 가장 최저값을 m에 삽입
    int m=Integer.MIN_VALUE;
    String[] s = [redacted]
    for(String x : s){
        int len=[redacted]
        if([redacted]){
            [redacted]
        }
    }
    System.out.println(answer);
}

```

#### 문제 4. 단어 뒤집기

N개의 단어가 주어지면 각 단어를 뒤집어 출력하는 프로그램을 작성하세요.

##### ■ 입력설명

첫 줄에 자연수 N( $3 \leq N \leq 20$ )이 주어집니다.

두 번째 줄부터 N개의 단어가 각 줄에 하나씩 주어집니다. 단어는 영어 알파벳으로만 구성되어 있습니다.

##### ■ 출력설명

N개의 단어를 입력된 순서대로 한 줄에 하나씩 뒤집어서 출력합니다.

##### ■ 입력예제 1

3

good

Time

Big

##### ■ 출력예제 1

doog

emiT

giB

<hint>

String.valueOf(char[]) : 캐릭터 배열을 String으로 반환한다.

```

public Quiz04() {
    int n=scan.nextInt();
    String[] str=new String[n];
    for(int i=0; i<n; i++){
        str[i]=scan.next();
    }
    for(String x : str){
        char[] s=x.toCharArray();
        int lt=0, rt=x.length()-1;
        while([redacted]){
            [redacted]
        }
        String tmp=String.valueOf(s);
        answer.add(tmp);
    }
    for(String x : answer){
        System.out.println(x);
    }
}

```

#### 문제 5. 특정 문자 뒤집기

영어 알파벳과 특수문자로 구성된 문자열이 주어지면 영어 알파벳만 뒤집고, 특수문자는 자기 자리에 그대로 있는 문자열을 만들어 출력하는 프로그램을 작성하세요.

##### ■ 입력설명

첫 줄에 길이가 100을 넘지 않는 문자열이 주어집니다.

##### ■ 출력설명

첫 줄에 알파벳만 뒤집힌 문자열을 출력합니다.

##### ■ 입력예제 1

a#b!GE\*T@S

##### ■ 출력예제 1

S#T!EG\*b@a

<hint>

Character.isAlphabetic(char) : 해당 문자가 알파벳인지 아닌지 검사하는 메서드

```
public Quiz05() {
    String str=scan.next();
    String answer;
    char[] s=str.toCharArray();
    int lt=0, rt=str.length()-1;
    while(lt<rt){
        if(!Character.isAlphabetic(s[lt])) [redacted]
        else if(!Character.isAlphabetic(s[rt])) [redacted]
        else{
            [redacted]
        }
    }
    answer=String.valueOf(s);
    System.out.println(answer);
}
```

#### 문제 6. 중복문자제거

소문자로 된 한개의 문자열이 입력되면 중복된 문자를 제거하고 출력하는 프로그램을 작성하세요. 제거된 문자열의 각 문자는 원래 문자열의 순서를 유지합니다.

##### ■ 입력설명

첫 줄에 문자열이 입력됩니다. 문자열의 길이는 100을 넘지 않는다.

##### ■ 출력설명

첫 줄에 중복문자가 제거된 문자열을 출력합니다.

##### ■ 입력예제 1

ksekkset

##### ■ 출력예제 1

kset

```
public Quiz06() {
    String str=scan.next();
    String answer="";
    for(int i=0; i<str.length(); i++){
        if( [redacted] )
            [redacted]
    }
    System.out.println(answer);
}
```

#### 문제 7. 회문 문자열

앞에서 읽을 때나 뒤에서 읽을 때나 같은 문자열을 회문 문자열이라고 합니다. 문자열이 입력되면 해당 문자열이 회문 문자열이면 "YES", 회문 문자열이 아니면 "NO"를 출력하는 프로그램을 작성하세요. 단 회문을 검사할 때 대소문자를 구분하지 않습니다.

##### ■ 입력설명

첫 줄에 길이 100을 넘지 않는 공백이 없는 문자열이 주어집니다.

##### ■ 출력설명

첫 번째 줄에 회문 문자열인지의 결과를 YES 또는 NO로 출력합니다.

##### ■ 입력예제 1

gooG

##### ■ 출력예제 1

YES

```
public Quiz07() {
    String str=scan.next();
    String answer="YES";
    str=str.toLowerCase();
    int len=str.length();
    for(int i=0; i<len/2; i++){
        if( [redacted] )
            answer="NO";
    }
    System.out.println(answer);
}
```

## 문제 8. 유효한 팰린드롬

앞에서 읽을 때나 뒤에서 읽을 때나 같은 문자열을 팰린드롬이라고 합니다. 문자열이 입력되면 해당 문자열이 팰린드롬이면 "YES", 아니면 "NO"를 출력하는 프로그램을 작성하세요. 단 회문을 검사할 때 알파벳만 가지고 회문을 검사하며, 대소문자를 구분하지 않습니다. 알파벳 이외의 문자들의 무시합니다.

### ■ 입력설명

첫 줄에 길이 100을 넘지 않는 공백이 없는 문자열이 주어집니다.

### ■ 출력설명

첫 번째 줄에 팰린드롬인지의 결과를 YES 또는 NO로 출력합니다.

### ■ 입력예제 1

found7, time: study; Yduts; emit, 7Dnuof

### ■ 출력예제 1

YES

<hint>

String.replaceAll([정규표현식],str) : 정규표현식에 해당하는 모든 문자를 str로 바꾼다.

new StringBuilder(string) : String의 완전 객체 형태, 단 이것은 String처럼 쓸 수 없지만 String을 인자로 받아 사용이 가능하다.

StringBuilder.reverse() : 문자열을 뒤집는다.

StringBuilder.toString() : StringBuilder 객체를 문자열로 반환한다.

```
public Quiz08() {
    String str=scan.nextLine();
    String answer="NO";
    // 대문자로 변형 시킨 이후에 대문자를 제외한
    // 모든 특수문자를 삭제한다.
    str=str.
        .replaceAll("[^A-Z]", "");
    // StringBuilder를 사용해서 문자를 뒤집은 다음
    // String 값을 얻는다.
    String tmp=new StringBuilder(str)
        .reverse()
        .toString();
    if(str.equals(tmp)) answer="YES";
    System.out.println(answer);
}
```

## 문제 9. 숫자만 추출

문자와 숫자가 섞여있는 문자열이 주어지면 그 중 숫자만 추출하여 그 순서대로 자연수를 만듭니다. 만약 "tge0a1h205er"에서 숫자만 추출하면 0, 1, 2, 0, 5이고 이것을 자연수를 만들면 1205이 됩니다. 추출하여 만들어지는 자연수는 100,000,000을 넘지 않습니다.

### ■ 입력설명

첫 줄에 숫자가 섞인 문자열이 주어집니다. 문자열의 길이는 100을 넘지 않습니다.

### ■ 출력설명

첫 줄에 자연수를 출력합니다.

### ■ 입력예제 1

g0en2T0s8eSoft

### ■ 출력예제 1

208

<hint>

Character.isDigit(x) : 캐릭터 x가 숫자인지 아닌지 판별하는 메서드

Character.getNumericValue(x) : x가 숫자일 경우 숫자로 변환시키는 메서드

```
public Quiz09() {
    String str=scan.next();
    String answer="";
    for(char x : [redacted]) {
        if([redacted])
            [redacted]
    }
    System.out.println(answer);
}
```

#### 문제 10. 가장 짧은 문자거리

한 개의 문자열 s와 문자 t가 주어지면 문자열 s의 각 문자가 문자 t와 떨어진 최소거리를 출력하는 프로그램을 작성하세요.

##### ■ 입력설명

첫 번째 줄에 문자열 s와 문자 t가 주어진다. 문자열과 문자는 소문자로만 주어집니다. 문자열의 길이는 100을 넘지 않습니다.

##### ■ 출력설명

첫 번째 줄에 각 문자열 s의 각 문자가 문자 t와 떨어진 거리를 순서대로 출력한다.

##### ■ 입력예제 1

teacher mode e

##### ■ 출력예제 1

1 0 1 2 1 0 1 2 2 1 0

<hint>

Math.min(a, b) : a와 b중에 최소값을 출력

```
public Quiz10() {
    String s=scan.next();
    char t=scan.next().charAt(0);
    int[] answer=new int[s.length()];
    // 실제 캐릭터의 거리 관련 변수
    int p=1000;
    // 앞에서 진행
    for(int i=[redacted]) {
        if([redacted]) {
            [redacted]
        }
        else{
            [redacted]
        }
    }
    p=1000;
    // 뒤에서 진행
    for(int i=[redacted]) {
        if([redacted]) [redacted];
        else{
            [redacted]
        }
    }
    for(int x : answer){
        System.out.print(x+" ");
    }
}
```

#### 문제 11. 문자열 압축

알파벳 대문자로 이루어진 문자열을 입력받아 같은 문자가 연속으로 반복되는 경우 반복되는 문자 바로 오른쪽에 반복 횟수를 표기하는 방법으로 문자열을 압축하는 프로그램을 작성하시오. 단 반복횟수가 1인 경우 생략합니다.

##### ■ 입력설명

첫 줄에 문자열이 주어진다. 문자열의 길이는 100을 넘지 않는다.

##### ■ 출력설명

첫 줄에 압축된 문자열을 출력한다.

##### ■ 입력예제 1

KKHSSSSSSSE

##### ■ 출력예제 1

K2HS7E

```

public Quiz11() {
    String s = scan.next();
    String answer="";
    s=s+" ";
    int cnt=1;
    for(int i= ) {
        if( ) {
            else{
        }
    }
    System.out.println(answer);
}

```

```

public Quiz01() {
    int n=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    ArrayList<Integer> answer =
        new ArrayList<Integer>();
    answer.add(arr[0]);
    for(int i= ) {
        if( ) {
    }
    for(int x : answer){
        System.out.print(x+" ");
    }
}

```

## 문제 12. 큰 수 출력하기

N( $1 \leq N \leq 100$ )개의 정수를 입력받아, 자신의 바로 앞 수보다 큰 수만 출력하는 프로그램을 작성하세요.(첫 번째 수는 무조건 출력한다)

### ■ 입력설명

첫 줄에 자연수 N이 주어지고, 그 다음 줄에 N개의 정수가 입력된다.

### ■ 출력설명

자신의 바로 앞 수보다 큰 수만 한 줄로 출력한다.

### ■ 입력예제 1

6

7 3 9 5 6 12

### ■ 출력예제 1

7 9 6 12

## 문제 13. 보이는 학생

선생님이 N명의 학생을 일렬로 세웠습니다. 일렬로 서 있는 학생의 키가 앞에서부터 순서대로 주어질 때, 맨 앞에 서 있는 선생님이 볼 수 있는 학생의 수를 구하는 프로그램을 작성하세요. (어떤 학생이 자기 앞에 서 있는 학생들보다 크면 그 학생은 보이고, 작거나 같으면 보이지 않습니다.)

### ■ 입력설명

첫 줄에 정수 N( $5 \leq N \leq 100,000$ )이 입력된다. 그 다음줄에 N명의 학생의 키가 앞에서부터 순서대로 주어진다.

### ■ 출력설명

선생님이 볼 수 있는 최대학생수를 출력한다.

### ■ 입력예제 1

8

130 135 148 140 145 150 150 153

### ■ 출력예제 1

5

```

public Quiz02() {
    int n=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }
    int answer=1, max=arr[0];
    for(int i=1; i<n; i++){
        if(arr[i]>max){
            max=arr[i];
        }
    }
    System.out.println(answer);
}

```

문제 14. 가위 바위 보

A, B 두 사람이 가위바위보 게임을 합니다. 총 N번의 게임을 하여 A가 이기면 A를 출력하고, B가 이기면 B를 출력합니다. 비길 경우에는 D를 출력합니다. 가위, 바위, 보의 정보는 1:가위, 2:바위, 3:보로 정하겠습니다.

예를 들어 N=5이면

회수	1	2	3	4	5
A의 정보	2	3	3	1	3
B의 정보	1	1	2	2	3
승자	A	B	A	B	D

두 사람의 각 회의 가위, 바위, 보 정보가 주어지면 각 회를 누가 이겼는지 출력하는 프로그램을 작성하세요.

입력설명

첫 번째 줄에 게임 횟수인 자연수  $N(1 \leq N \leq 100)$ 이 주어집니다. 두 번째 줄에는 A가 낸 가위, 바위, 보 정보가 N개 주어집니다. 세 번째 줄에는 B가 낸 가위, 바위, 보 정보가 N개 주어집니다.

출력설명

각 줄에 각 회의 승자를 출력합니다. 비겼을 경우는 D를 출력합니다.

입력예제 1

5

2 3 3 1 3

1 1 2 2 3

출력예제 1

A

B

A

B

D

```

public Quiz03() {
    int n=scan.nextInt();
    int[] a=new int[n];
    int[] b=new int[n];
    for(int i=0; i<n; i++){
        a[i]=scan.nextInt();
    }
    for(int i=0; i<n; i++){
        b[i]=scan.nextInt();
    }

    String answer="";
    for(int i=0; i<n; i++){
        // 비겼을 경우 D 출력
        // A가 이겼을 경우 A 출력
        // B가 이겼을 경우 B 출력
    }

    for(char x : answer.toCharArray())
        System.out.println(x);
}

```

문제 15. 피보나치 수열

1) 피보나치 수열을 출력한다. 피보나치 수열이란 앞의 2개의 수를 합하여 다음 숫자가 되는 수열이다.

2) 입력은 피보나치 수열의 총 항의 수 이다. 만약 7이 입력되면 1 1 2 3 5 8 13을 출력하면

된다.

입력설명

첫 줄에 총 항수  $N(3 \leq N \leq 45)$ 이 입력된다.



■ 출력설명

첫 줄에 피보나치 수열을 출력합니다.

■ 입력예제 1

10

■ 출력예제 1

1 1 2 3 5 8 13 21 34 55

```
public Quiz04() {
    int n=scan.nextInt();
    int[] answer=new int[n];
    answer[0]=1;
    answer[1]=1;
    for(int x : answer)
        System.out.print(x+" ");
}
```

문제 16. 소수(에라토스테네스 체)

자연수 N이 입력되면 1부터 N까지의 소수의 개수를 출력하는 프로그램을 작성하세요. 만약 20이 입력되면 1부터 20까지의 소수는 2, 3, 5, 7, 11, 13, 17, 19로 총 8개입니다. 제한시간은 1초입니다.

■ 입력설명

첫 줄에 자연수의 개수 N( $2 \leq N \leq 200,000$ )이 주어집니다.

■ 출력설명

첫 줄에 소수의 개수를 출력합니다.

■ 입력예제 1

20

■ 출력예제 1

8

```
public Quiz05() {
    int n=scan.nextInt();
    int cnt=0;
    int[] ch = new int[n+1];
    for(int i=2; i<=n; i++){
        if(ch[i]==0){
            cnt++;
            for(int j=i; j<=n; j+=i)
                ch[j]=1;
        }
    }
    System.out.println(cnt);
}
```

문제 17. 뒤집은 소수

N개의 자연수가 입력되면 각 자연수를 뒤집은 후 그 뒤집은 수가 소수이면 그 소수를 출력하는 프로그램을 작성하세요. 예를 들어 32를 뒤집으면 23이고, 23은 소수이다. 그러면 23을 출력한다. 단 910를 뒤집으면 019로 숫자화 해야 한다. 첫 자리부터의 연속된 0은 무시한다.

■ 입력설명

첫 줄에 자연수의 개수 N( $3 \leq N \leq 100$ )이 주어지고, 그 다음 줄에 N개의 자연수가 주어진다. 각 자연수의 크기는 100,000를 넘지 않는다.

■ 출력설명

첫 줄에 뒤집은 소수를 출력합니다. 출력순서는 입력된 순서대로 출력합니다.

■ 입력예제 1

9

32 55 62 20 250 370 200 30 100

■ 출력예제 1

23 2 73 2 3

```

public Quiz06() {
    int n=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    ArrayList<Integer> answer =
        new ArrayList<>();
    for(int i=0; i<n; i++){
        int tmp=arr[i];
        int res=0;
        // 숫자를 뒤집는 로직
        while( ) {
            [ ]
        }
        // 소수인지 판별하는 로직
        if(isPrime(res)) answer.add(res);
    }
    for(int x : answer){
        System.out.print(x+" ");
    }
}

public boolean isPrime(int num){
    [ ]
}

```

#### 문제 18. 점수계산

OX 문제는 맞거나 틀린 두 경우의 답을 가지는 문제를 말한다. 여러 개의 OX 문제로 만들어진 시험에서 연속적으로 답을 맞히는 경우에는 가산점을 주기 위해서 다음과 같이 점수 계산을 하기로 하였다. 1번 문제가 맞는 경우에는 1점으로 계산한다. 앞의 문제에 대해서는 답을 틀리다가 답이 맞는 처음 문제는 1점으로 계산한다. 또한, 연속으로 문제의 답이 맞는 경우에서 두 번째 문제는 2점, 세 번째 문제는 3점, ..., K번째 문제는 K점으로 계산한다. 틀린 문제는 0점으로 계산한다. 예를 들어, 아래와 같이 10 개의 OX 문제에서 답이 맞은 문제의 경우에는 1로 표시하고, 틀린 경우에는 0으로 표시하였을 때, 점수 계산은 아래 표와 같이 계산되어, 총 점수는 1+1+2+3+1+2=10 점이다.

1 0 1 1 1 0 0 1 1 0

채점	1	0	1	1	1	0	0	1	1	0
점수	1	0	1	2	3	0	0	1	2	0

시험문제의 채점 결과가 주어졌을 때, 총 점수를 계산하는 프로그램을 작성하시오.

#### 입력설명

첫째 줄에 문제의 개수  $N$  ( $1 \leq N \leq 100$ )이 주어진다. 둘째 줄에는  $N$ 개 문제의 채점 결과를 나타내는 0 혹은 1이 빈 칸을 사이에 두고 주어진다. 0은 문제의 답이 틀린 경우이고, 1은 문제의 답이 맞는 경우이다.

#### 출력설명

첫째 줄에 입력에서 주어진 채점 결과에 대하여 가산점을 고려한 총 점수를 출력한다.

#### 입력예제 1

10

1 0 1 1 1 0 0 1 1 0

#### 출력예제 1

10

```

public Quiz07() {
    int n=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    int answer=0, cnt=0;
    for(int i= ) {
        [ ]
    }
    System.out.println(answer);
}

```

문제 19. 등수구하기

N명의 학생의 국어점수가 입력되면 각 학생의 등수를 입력된 순서대로 출력하는 프로그램을 작성하세요. 같은 점수가 입력될 경우 높은 등수로 동일 처리한다. 즉 가장 높은 점수가 92점인데 92점이 3명 존재하면 1등이 3명이고 그 다음 학생은 4등이 된다.

■ 입력설명

첫 줄에 N( $3 \leq N \leq 100$ )이 입력되고, 두 번째 줄에 국어점수를 의미하는 N개의 정수가 입력된다.

■ 출력설명

입력된 순서대로 등수를 출력한다.

■ 입력예제 1

5  
87 89 92 100 76

■ 출력예제 1

4 3 2 1 5

```
public Quiz08() {
    int n=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    int[] answer = new int[n];
    for(int i=0; i<n; i++){
        int cnt=1;
        for(int j=i+1; j<n; j++){
            if(arr[i]>arr[j]) cnt++;
        }
        answer[i]=cnt;
    }
    for(int x : answer)
        System.out.print(x+" ");
}
```

문제 20. 격자판 최대합

5\*5 격자판에 아래와 같이 숫자가 적혀있습니다.

<del>10</del>	18	10	12	<del>15</del>
<del>12</del>	<del>30</del>	30	<del>28</del>	<del>11</del>
11	25	<del>50</del>	53	15
19	<del>27</del>	29	<del>37</del>	27
<del>19</del>	18	30	13	<del>19</del>

N\*N의 격자판이 주어지면 각 행의 합, 각 열의 합, 두 대각선의 합 중 가장 큰 합을 출력합니다.

■ 입력설명

첫 줄에 자연수 N이 주어진다.( $2 \leq N \leq 50$ ) 두 번째 줄부터 N줄에 걸쳐 각 줄에 N개의 자연수가 주어진다. 각 자연수는 100을 넘지 않는다.

■ 출력설명

최대합을 출력합니다.

■ 입력예제 1

5  
10 13 10 12 15  
12 39 30 23 11  
11 25 50 53 15  
19 27 29 37 27  
19 13 30 13 19

■ 출력예제 1

155

<hint>

Math.max(a, b) : a와 b중에 큰 수를 출력한다.

```

public Quiz09() {
    int n=scan.nextInt();
    int[][] arr=new int[n][n];
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            arr[i][j]=scan.nextInt();
        }
    }

    int answer=Integer.MIN_VALUE;
    int sum1=0, sum2=0;
    // 가로 세로 합계 비교
    for(int i=0; i<n; i++){
        sum1=sum2=0;
        for(int j=0; j<n; j++){
            // 
        }
        answer=Math.max(answer, sum1);
        answer=Math.max(answer, sum2);
    }
    // 대각선 합계 비교
    sum1=sum2=0;
    for(int i=0; i<n; i++){
        // 
    }
    answer=Math.max(answer, sum1);
    answer=Math.max(answer, sum2);

    System.out.println(answer);
}

```

#### □ 입력설명

첫 줄에 자연수  $N$ 이 주어진다. ( $2 \leq N \leq 50$ ) 두 번째 줄부터  $N$ 줄에 걸쳐 각 줄에  $N$ 개의 자연수가 주어진다. 각 자연수는 100을 넘지 않는다.

#### □ 출력설명

봉우리의 개수를 출력하세요.

#### □ 입력예제 1

5

5 3 7 2 3

3 7 1 6 1

7 2 5 3 4

4 3 6 4 1

8 7 3 5 2

#### □ 출력예제 1

10

#### 문제 21. 봉우리

지도 정보가  $N \times N$  격자판에 주어집니다. 각 격자에는 그 지역의 높이가 쓰여있습니다. 각 격자판의 숫자 중 자신의 상하좌우 숫자보다 큰 숫자는 봉우리 지역입니다. 봉우리 지역이 몇 개 있는 지 알아내는 프로그램을 작성하세요. 격자의 가장자리는 0으로 초기화 되었다고 가정한다. 만약  $N=5$  이고, 격자판의 숫자가 다음과 같다면 봉우리의 개수는 10개입니다.

0	0	0	0	0	0	0
0	5	3	7	2	3	0
0	3	7	1	6	1	0
0	7	2	5	3	4	0
0	4	3	6	4	1	0
0	8	7	3	5	2	0
0	0	0	0	0	0	0

```

Scanner scan = new Scanner(System.in);
// 어떤 지역을 중심으로 상하좌우
int[] dx={-1, 0, 1, 0};
int[] dy={0, 1, 0, -1};

public Quiz10() {
    int n=scan.nextInt();
    int[][] arr=new int[n][n];
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            arr[i][j]=scan.nextInt();
        }
    }

    int answer=0;
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            boolean flag=true;
            for(int k=0; k<4; k++){
                int nx=
                int ny=
                if(
            ){
                flag=false;
                break;
            }
            if(flag) answer++;
        }
    }
    System.out.println(answer);
}

```

은 반이었고, 3번 학생 및 5번 학생과 3학년 때 같은 반이었으며, 2번 학생과는 4학년 때 같은 반이었음을 알 수 있다. 그러므로 이 학급에서 4번 학생과 한번이라도 같은 반이었던 사람은 2번 학생, 3번 학생과 5번 학생으로 모두 3명이다. 이 예에서 4번 학생이 전체 학생 중에서 같은 반이었던 학생 수가 제일 많으므로 임시 반장이 된다. 각 학생들이 1학년부터 5학년까지 속했던 반이 주어질 때, 임시 반장을 정하는 프로그램을 작성하시오.

#### ■ 입력설명

첫째 줄에는 반의 학생 수를 나타내는 정수가 주어진다. 학생 수는 3 이상 1000 이하이다. 둘째 줄부터는 1번 학생부터 차례대로 각 줄마다 1학년부터 5학년까지 몇 반에 속했었는지를 나타내는 5개의 정수가 빈칸 하나를 사이에 두고 주어진다. 주어지는 정수는 모두 1 이상 9 이하의 정수이다.

#### ■ 출력설명

첫 줄에 임시 반장으로 정해진 학생의 번호를 출력한다. 단, 임시 반장이 될 수 있는 학생이 여러 명인 경우에는 그 중 가장 작은 번호만 출력한다.

#### ■ 입력예제 1

```

5
2 3 1 7 3
4 1 9 6 8
5 5 2 4 4
6 5 2 6 7
8 4 2 2 2

```

#### ■ 출력예제 1

```

4

```

#### 문제 22. 임시반장 정하기

김근형 선생님은 올해 6학년 1반 담임을 맡게 되었다. 김근형 선생님은 우선 임시로 반장을 정하고 학생들이 서로 친숙해진 후에 정식으로 선거를 통해 반장을 선출하려고 한다. 그는 자기반 학생 중에서 1학년부터 5학년까지 지내오면서 한번이라도 같은 반이었던 사람이 가장 많은 학생을 임시 반장으로 정하려 한다. 그래서 김근형 선생님은 각 학생들이 1학년부터 5학년까지 몇 반에 속했었는지를 나타내는 표를 만들었다. 예를 들어 학생 수가 5명일 때의 표를 살펴보자.

	1학년	2학년	3학년	4학년	5학년
1번 학생	2	3	1	7	3
2번 학생	4	1	9	6	8
3번 학생	5	5	2	4	4
4번 학생	6	5	2	6	7
5번 학생	8	4	2	2	2

위 경우에 4번 학생을 보면 3번 학생과 2학년 때 같

```

public Quiz11() {
    int n=scan.nextInt();
    int[][] arr=new int[n+1][6];
    for(int i=1; i<=n; i++){
        for(int j=1; j<=5; j++){
            arr[i][j]=scan.nextInt();
        }
    }

    int answer=0, max=0;
    // i 번째 학생
    for(int i=1; i<=n; i++){
        int cnt=0;
        // j 번째 학생
        for(int j=1; j<=5; j++){
            // k는 1학년 부터 5학년 까지
            for(int k=1; k<=5; k++){
                if(arr[i][j]==arr[i][k]){
                    cnt++;
                    break;
                }
            }
        }
        if(cnt>max){
            max=cnt;
        }
    }
    System.out.println(answer);
}

```

2 3 6 7 9

#### ■ 출력예제 1

1 2 3 3 5 6 7 9

```

public Quiz01() {
    int n=scan.nextInt();
    int[] a=new int[n];
    for(int i=0; i<n; i++){
        a[i]=scan.nextInt();
    }
    int m=scan.nextInt();
    int[] b=new int[m];
    for(int i=0; i<m; i++){
        b[i]=scan.nextInt();
    }

    ArrayList<Integer> answer =
        new ArrayList<>();
    int p1=0, p2=0;
    while(p1<a.length & p2<b.length){
        if(a[p1]<b[p2]){
            answer.add(a[p1]);
            p1++;
        } else if(a[p1]>b[p2]){
            answer.add(b[p2]);
            p2++;
        } else {
            answer.add(a[p1]);
            p1++;
            p2++;
        }
    }

    for(int x : answer)
        System.out.print(x+" ");
}

```

#### 문제 23. 두 배열 합치기

오름차순으로 정렬이 된 두 배열이 주어지면 두 배열을 오름차순으로 합쳐 출력하는 프로그램을 작성하세요.

##### ■ 입력설명

첫 번째 줄에 첫 번째 배열의 크기  $N(1 \leq N \leq 100)$ 이 주어집니다. 두 번째 줄에  $N$ 개의 배열 원소가 오름차순으로 주어집니다. 세 번째 줄에 두 번째 배열의 크기  $M(1 \leq M \leq 100)$ 이 주어집니다. 네 번째 줄에  $M$ 개의 배열 원소가 오름차순으로 주어집니다. 각 리스트의 원소는 int형 변수의 크기를 넘지 않습니다.

##### ■ 출력설명

오름차순으로 정렬된 배열을 출력합니다.

##### ■ 입력예제 1

3

1 3 5

5

#### 문제 24. 공통원소 구하기

A, B 두 개의 집합이 주어지면 두 집합의 공통 원소를 추출하여 오름차순으로 출력하는 프로그램을 작성하세요.

##### ■ 입력설명

첫 번째 줄에 집합 A의 크기  $N(1 \leq N \leq 30,000)$ 이 주어집니다. 두 번째 줄에  $N$ 개의 원소가 주어집니다. 원소가 중복되어 주어지지 않습니다. 세 번째 줄에 집합 B의 크기  $M(1 \leq M \leq 30,000)$ 이 주어집니다. 네 번째 줄에  $M$ 개의 원소가 주어집니다. 원소가 중복되어 주어지지 않습니다. 각 집합의 원소는 1,000,000,000이하의 자연수입니다.

##### ■ 출력설명

두 집합의 공통원소를 오름차순 정렬하여 출력합니다.

■ 입력예제 1

5

1 3 9 5 2

5

3 2 5 7 8

■ 출력예제 1

2 3 5

```
public Quiz02() {
    int n=scan.nextInt();
    int[] a=new int[n];
    for(int i=0; i<n; i++){
        a[i]=scan.nextInt();
    }
    int m=scan.nextInt();
    int[] b=new int[m];
    for(int i=0; i<m; i++){
        b[i]=scan.nextInt();
    }

    ArrayList<Integer> answer =
        new ArrayList<Integer>();
    // 각각 배열 a와 b를 오름차순 정렬
    Arrays.sort(a);
    Arrays.sort(b);
    int p1=0, p2=0;
    while( ) {
        if( ) {
            ;
        }
        else if( ) ;
        else ;
    }
    for(int x : answer)
        System.out.print(x+" ");
}
```

문제 25. 최대 매출

현수의 아빠는 제과점을 운영합니다. 현수아빠는 현수에게 N일 동안의 매출기록을 주고 연속된 K일 동안의 최대 매출액이 얼마인지 구하라고 했습니다. 만약 N=10이고 10일 간의 매출기록이 아래와 같습니다. 이

때 K=3이면

12 15 11 20 25 10 20 19 13 15

연속된 3일간의 최대 매출액은 11+20+25=56만원입니다. 여러분이 현수를 도와주세요.

■ 입력설명

첫 줄에  $N(5 \leq N \leq 100,000)$ 과  $K(2 \leq K \leq N)$ 가 주어집니다. 두 번째 줄에 N개의 숫자열이 주어집니다. 각 숫자는 500이하의 음이 아닌 정수입니다.

■ 출력설명

첫 줄에 최대 매출액을 출력합니다.

■ 입력예제 1

10 3

12 15 11 20 25 10 20 19 13 15

■ 출력예제 1

56

```
public Quiz03() {
    int n=scan.nextInt();
    int k=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    int answer, sum=0;
    for(int i= )
        ;
    for(int i= ) {
        ;
    }
    System.out.println(answer);
}
```

문제 26. 연속 부분수열

N개의 수로 이루어진 수열이 주어집니다.

이 수열에서 연속부분수열의 합이 특정숫자 M이 되는 경우가 몇 번 있는지 구하는 프로그램을 작성하세요.



만약  $N=8$ ,  $M=6$ 이고 수열이 다음과 같다면

1 2 1 3 1 1 1 2

합이 6이 되는 연속부분수열은 {2, 1, 3}, {1, 3, 1, 1}, {3, 1, 1, 1}로 총 3가지입니다.

#### ■ 입력설명

첫째 줄에  $N(1 \leq N \leq 100,000)$ ,  $M(1 \leq M \leq 100,000,000)$ 이 주어진다. 수열의 원소값은 1,000을 넘지 않는 자연수이다.

#### ■ 출력설명

첫째 줄에 경우의 수를 출력한다.

#### ■ 입력예제 1

8 6

1 2 1 3 1 1 1 2

#### ■ 출력예제 1

3

```
public Quiz04() {
    int n=scan.nextInt();
    int m=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    int answer=0, sum=0, lt=0;
    for(int rt=0; rt<n; rt++){
        while (sum > m) {
            sum -= arr[lt];
            lt++;
        }
        sum += arr[rt];
        if (sum == m) answer++;
    }
    System.out.println(answer);
}
```

문제 27. 연속된 자연수의 합

$N$ 입력으로 양의 정수  $N$ 이 입력되면 2개 이상의 연속

된 자연수의 합으로 정수  $N$ 을 표현하는 방법의 가짓수를 출력하는 프로그램을 작성하세요.

만약  $N=15$ 이면

$7+8=15$

$4+5+6=15$

$1+2+3+4+5=15$

와 같이 총 3가지의 경우가 존재한다.

#### ■ 입력설명

첫 번째 줄에 양의 정수  $N(7 \leq N < 1000)$ 이 주어집니다.

#### ■ 출력설명

첫 줄에 총 경우수를 출력합니다.

#### ■ 입력예제 1

15

#### ■ 출력예제 1

3

```
public Quiz05() {
    int n=scan.nextInt();
    int answer=0, cnt=1;
    n--;
    while (cnt <= n) {
        // ...
    }
    System.out.print(answer);
}
```

문제 28. 최대 길이 연속부분수열

0과 1로 구성된 길이가  $N$ 인 수열이 주어집니다. 여러분은 이 수열에서 최대  $k$ 번을 0을 1로 변경할 수 있습니다. 여러분이 최대  $k$ 번의 변경을 통해 이 수열에서 1로만 구성된 최대 길이의 연속부분수열을 찾는 프로그램을 작성하세요.



만약 길이가 길이가 14인 다음과 같은 수열이 주어지고 k=2라면

1 1 0 0 1 1 0 1 1 0 1 1 0 1

여러분이 만들 수 있는 1이 연속된 연속부분수열은 1 1 0 0 1 1 1 1 1 1 1 1 0 1 이며 그

길이는 8입니다.

#### ■ 입력설명

첫 번째 줄에 수열의 길이인 자연수  $N(5 \leq N < 100,000)$ 이 주어집니다. 두 번째 줄에 N길이의 0과 1로 구성된 수열이 주어집니다.

#### ■ 출력설명

첫 줄에 최대 길이를 출력하세요.

#### ■ 입력예제 1

14 2

1 1 0 0 1 1 0 1 1 0 1 1 0 1

#### ■ 출력예제 1

8

```
public Quiz06() {
    int n=scan.nextInt();
    int k=scan.nextInt();
    int[] arr=new int[n];
    for(int i=0; i<n; i++){
        arr[i]=scan.nextInt();
    }

    int answer=0, cnt=0, lt=0;
    for(int rt=0; rt<n; rt++){
        if(arr[rt]==0) [redacted];
        while(cnt>k) {
            if(arr[lt]==0) [redacted];
            [redacted];
        }
        answer=[redacted];
    }
    System.out.println(answer);
}
```