

제 16회 대전시 정보올림피아드 본선대회 문제

(고등학생부) 수험번호() 이름()

[문제 1] 제품생산

한 공장에서 동일한 제품 j 개를 만들려고 한다. 이 제품을 만들려면 반드시 A형 기계에서 가공된 후에 B형 기계에서 마무리 가공을 해야 한다. A형 기계가 a 개 있는데 성능차이로 가공시간이 $c_1, c_2, \dots, c_a$ 분 걸리며, B형 기계가 b 개 있지만 성능차이로 가공시간이 $d_1, d_2, \dots, d_b$ 분 걸린다

이 때 A형 기계에서 모든 제품 가공이 끝나는 시각 t_a 와 B형 기계에서 모든 제품 가공이 끝나는 시각 t_b 를 구하는 프로그램을 작성하라. t_a 와 t_b 의 초기 값은 0이며, 가장 작은 값을 구하여야 한다.

j 의 범위는 $[1, 100]$, a 와 b 의 범위는 $[1, 10]$ 이 가능한 값이고, c_a 와 d_b 의 범위는 $[1, 5]$ 가 가능한 값이다. 예로 사용될 in.txt 와 out.txt 는 다음과 같다.

in.txt

 j a $c_1 \ c_2 \ \dots \ c_a$ b $d_1 \ d_2 \ \dots \ d_b$

예

5

2

1 1

3

1 3 4

out.txt

 t_a t_b

예

3

5

[문제 2] A/S 반복 횟수 계산

어느 고객지원센터의 특정 상담 유형의 건수가 요일별로 3개월치가 주어졌다고 한다. 이 것을 주별로 합한 뒤 그 발생횟수의 패턴이 3회 이상 반복적으로 나오는 2개 이상의 수들로 구성된 숫자 묶음과 그 반복 횟수를 계산하는 (숫자 묶음: 반복회수) 프로그램을 작성하여라.

in.txt

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

out.txt

2 3 4 2 2 3 4 2 2 3 4 2

```
(2 3: 3)
(3 4: 3)
(4 2: 3)
(2 3 4: 3)
(3 4 2: 3)
(2 3 4 2: 3)
```

[문제 3] 삼각형 개수 구하기

아래 그림과 같은 모양의 정사각형들을 몇 행 몇 열로 붙인 모형에서 그릴 수 있는 삼각형의 개수를 모두 찾는 프로그램을 다음 [처리 조건]에 따라 작성하시오.

<처리조건>

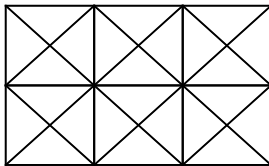
- ① 아래 그림과 같은 정사각형의 행의 개수와 열의 개수를 입력한다. 행과 열의 개수는 1~20 사이의 정수로 제한한다. 만일 다른 값들이 입력되면 에러 처리하고 재 입력을 수행한다.

예) 2행 3열로 붙인 모형을 입력할 경우 다음과 같이 입력한다.

row : 2

column : 3

(※ 참고, 위 입력 값은 다음과 같은 모형을 나타낸다.)



- ② 입력받은 모형에서 생성될 수 있는 삼각형의 개수를 모두 찾아 다음 [프로그램 수행 결과]와 같이 처리한다.
- ③ 삼각형의 개수를 출력한 다음에 프로그램 종료 여부를 확인한다.

exit program (y/n) ?

'Y' 또는 'y' 문자를 입력하면 'good-bye!' 메시지 출력과 함께 프로그램을 종료하고, 'N' 또는 'n' 문자를 입력하면 프로그램을 재 수행한다.

[프로그램 수행 결과]

```
row : 2
column : 30 /*1~20 범위를 넘었음*/
input error !
input again : 1 /*column의 수정 값을 재입력*/

number of triangle : 18

exit program (y/n) ? y

good-bye!
```

[문제 4] 길 찾기

철수는 걸어서 언덕 오르는 것을 너무나도 싫어해서 가는 곳마다 자전거를 탄다. 뿐만 아니라 한번 갈 곳을 선택하면 가능하면 가장 짧고 쉬운 길을 선택해 가기를 원한다. 그런데 다행스럽게도 철수가 사는 '격자동'은 자전거 도로가 아주 잘되어 있어서 가로 세로로 바둑판과 같이 격자로 펼쳐져 있다. 가로 길은 동서로 나아 있으며 세로 길은 남북으로 나아 있고 가로 세로가 만나는 격자 지점에서 바로 옆의 동서 혹은 남북의 가로 세로 지점까지의 거리는 똑 같이 되어 있다. 그런데 문제는 '격자동'이 매우 언덕이 많으며, 자전거 일방 통행 도로 또한 많다는 것이다.

철수가 자전거로 출발하는 지점과 도달하고자 하는 지점 사이의 적합한 길을 찾는데 있어서 철수는 다음과 같은 규칙을 정해놓고 있다.

1. 인접한 가로 세로 격자 지점 사이의 고도 차이가 10미터 보다 크면 올라가지 않는다.
2. 자전거 일방 통행 길을 절대로 거슬러 가지 않는다.
3. 출발 지점과 도달 지점까지 언제나 가장 짧은 거리를 여행한다.
4. 같은 거리일 경우 격자간 고도의 차를 +, - 증감에 의하여 그 합이 가장 적은 경로를 택한다(모두가 같은 경우는 어느 경로를 선택하여도 무방하다).

<입력>

입력 파일(bike.txt)은 다음과 같은 형식을 갖는 데이터를 포함한다.

- 첫 번째 라인은 두 개의 정수로 구성되는데 한 칸 이상으로 띄어져 있다. 첫 번째 정수 n 은 가로 길의 수를 나타내고, 두 번째 정수 m 은 세로 길의 수를 나타낸다. $1 \leq n \leq 20$, $1 \leq m \leq 20$.
- 첫 번째 라인 뒤에는 n 개의 라인을 갖게 되는데, 라인은 1부터 n 까지의 가로 길을 나타내고 각 라인은 또한 m 개의 정수를 갖는데 1부터 m 까지의 세로 길을 나타낸다. 이러한 식으로 구성되어 있는 각 정수는 그 격자 지점의 고도(미터 단위)를 나타낸다.
- $n+1$ 개의 라인 뒤에는 자전거 일방 통행 도로를 나타내는데, 두 개의 정수를 짝으로 첫 번째 짝은 시작 가로-세로 격자 지점을 두 번째 짝은 일방 통행 도로가 끝나는 가로-세로 격자 지점을 나타낸다. 즉 형식은 다음과 같다.

가로 세로 가로 세로

예를 들어 다음과 같이 되어 있을 경우

```
5 7 5 8
5 8 5 9
5 9 5 10
```

다섯 번째 가로 길 5와 일곱 번째 세로 길 7, 즉 5-7에서 5-8, 5-8에서 5-9, 그리고 5-9에서 5-10으로 자전거 일방 통행로가 되어 있는 것을 말한다.

만일 시작 격자 지점과 끝 격자 지점이 인접해 있지 않을 경우에는 시작 지점에서 끝 지점까지 중간
의 격자 지점을 관통하여 일방 통행인 것을 나타낸다. 일방 통행 도로에 대한 정보의 입력이 끝나
면 바로 뒷줄에 한 칸씩 띄어서 네 개의 0으로 구분하는 라인을 입력한다.

· 마지막으로 철수가 출발하는 지점에서 도착하고자 하는 지점까지를 나타내는 한 개 이상의 라인을
포함하는데, 형식은 앞에서와 같이 가로-세로 격자 지점으로 나타내는 출발점과 마찬가지로 가로-세
로 격자 지점으로 나타내는 도착점으로 구성된다. 즉 형식은 앞에서와 같다.

가로 세로 가로 세로

입력의 끝은 한 칸씩 띄어서 네 개의 0으로 구분하는 라인으로 마친다.

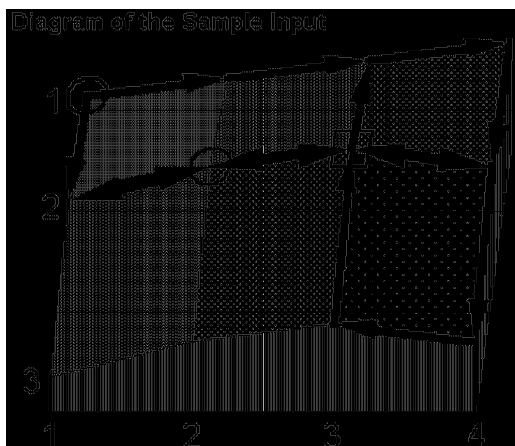
여기서 입력되는 모든 가로와 세로 격자 지점의 값들은 입력의 첫 번째 라인에서 입력한 범위 내에 있도록
한다.

<출 력>

출력 파일은 bike.out으로 한다. 여기서 여러분은 입력 파일에서 나타내었던 출발 지점과 도착 지점까지
철수가 정한 세 가지 규칙을 지키면서 각각 지나는 가로-세로 격자 지점을 순서대로 출력하는 프로그램을
작성하시오. 만일 세 가지 규칙을 만족하는 길의 순서가 두 가지 이상일 경우 그 중 어느 것이든 정답으
로 한다. 또한 만일 세 가지 규칙을 만족하는 길이 전혀 없거나 출발 지점과 도착 지점이 같다면 적합한
메시지를 출력한다. 하나의 출력 라인 뒤에는 한 줄을 비우고 다음 출력을 하는 것으로 한다.

Sample 입력과 출력은 다음에 있다.

[아래의 Sample Input 내용에 따른 지도의 예]



<입.출력 예>

Sample Input

```
3 4
10 15 20 25
19 30 35 30
10 19 26 20
1 1 1 4
2 1 2 4
3 4 3 3
3 3 1 3
1 4 3 4
2 4 2 1
1 1 2 1
0 0 0 0
1 1 2 2
2 3 2 3
2 2 1 1
0 0 0 0
```

Sample Output

1-1 to 1-2 to 1-3 to 1-4 to 2-4 to 2-3 to 2-2

To get from 2-3 to 2-3, stay put!

There is no acceptable route from 2-2 to 1-1

[문제 5] Factorial 구하기

Factorial은 수를 늘림에 따라 기하 급수적으로 그 값이 커지게 된다. 따라서 그러한 값을 적절히 처리하기 위해서는 컴퓨터에서 제공하는 정수나 실수의 한계로는 불가능하게 되어 할 수 없이 다른 방법으로 처리하여야만 한다. 25부터 49까지의 임의의 두 자리 정수 X, Y(X>Y)를 입력하여 X!, Y!, X!-Y!의 값을 구하는 프로그램을 작성하여라.

<조건>

1) $X! = 1 \times 2 \times 3 \times 4 \times \dots \times (X-1) \times X$ 이다.

2) 33!의 값이 8683317618811886495518194401280000000 일 경우

8 6 8 3 3 1 7 6 1 8

8 1 1 8 8 6 4 9 5 5

1 8 1 9 4 4 0 1 2 8

0 0 0 0 0 0 0

과 같이 한 줄에 10 자리 씩 출력한다.

3) X의 값이 0이면 출력을 끝내며 주어진 범위 밖의 수를 입력하였을 경우에는 X의 값을 다시 입력받도록 한다. Y의 값도 X 값 보다 작은 값으로 범위를 벗어나지 않게 입력받도록 한다.

<실행 예>

X = 33

8 6 8 3 3 1 7 6 1 8

8 1 1 8 8 6 4 9 5 5

1 8 1 9 4 4 0 1 2 8

0 0 0 0 0 0 0

Y = 31

8 2 2 2 8 3 8 6 5 4

1 7 7 9 2 2 8 1 7 7

2 5 5 6 2 8 8 0 0 0

0 0 0 0

X! - Y!

8 6 7 5 0 9 4 7 8 0

1 5 7 7 0 8 5 7 2 7

0 0 4 6 8 8 3 8 4 0

0 0 0 0 0 0 0

X = 0