

3의 배수는 통과?

1부터 입력한 정수까지 1씩 증가시켜 출력하는 프로그램을 작성하되,
3의 배수인 경우는 출력하지 않도록 만들어보자.

예를 들면,
1 2 4 5 7 8 10 11 13 14 ...
와 같이 출력하는 것이다.

참고
반복문 안에서 continue;가 실행되면 그 아래의 내용을 건너뛰고, 다음 반복을 수행한다.
즉, 다음 반복으로 넘어가는 것이다.

예시

```
for(i=1; i<=n; i++)    //1부터 시작해 n까지 1씩 증가시키면서 반복한다.
{
    if(i%2==0)
        continue;      //짝수라면? 아래 부분은 건너뛰고! 계속~
    printf("%d ", i);    //결국 짝수가 아닐 때에만 출력된다.
}
```

◎ 입력 형식

정수 한 개를 입력받는다.
(1 ~ 100)

◎ 출력 형식

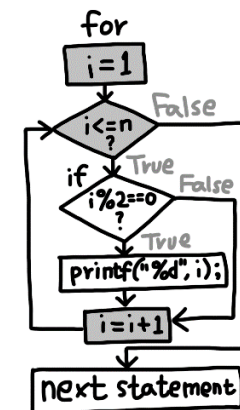
1부터 입력한 정수보다 작거나 같을 때까지 1씩 증가시켜 출력하되
3의 배수는 출력하지 않는다.

입력 예시

10

출력 예시

1 2 4 5 7 8 10



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29

수 나열하기1

어떤 규칙에 따라 수를 순서대로 나열한 것을 수열(series)이라고 한다.

예를 들어

1 4 7 10 13 16 19 22 25 ... 은

1부터 시작해 이전에 만든 수에 3을 더해 다음 수를 만든 수열이다.

이러한 것을 수학에서는 앞뒤 수들의 차이가 같다고 하여
등차(차이가 같다는 한문 말) 수열이라고 한다.

수열을 알게 된 영일이는 갑자기 궁금해졌다.

"그럼.... 123번째 나오는 수는 뭘까?"

영일이는 프로그램을 만들어 더 큰 수도 자동으로 계산하고 싶어졌다.

시작 값(a), 등차(d), 몇 번째인지를 나타내는 정수(n)가 입력될 때
n번째 수를 출력하는 프로그램을 만들어보자.

◎ 입력 형식

시작 값(a), 등차의 값(d), 몇 번째 수 인지를 의미하는 정수(n)가
공백을 두고 입력된다. (모두 0 ~ 100)

◎ 출력 형식

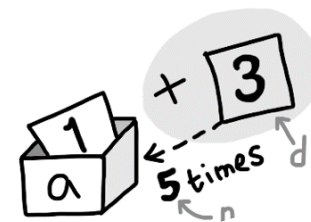
n번째 수를 출력한다.

입력 예시

1 3 5

출력 예시

13



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

수 나열하기2

어떤 규칙에 따라 수를 순서대로 나열한 것을 수열이라고 한다.

예를 들어

2 6 18 54 162 486 ... 은

2부터 시작해 이전에 만든 수에 3을 곱해 다음 수를 만든 수열이다.

이러한 것을 수학에서는 앞뒤 수들의 비율이 같다고 하여
등비(비율이 같다는 한문 말) 수열이라고 한다.

등비 수열을 알게된 영일이는 갑자기 궁금해졌다.

"그럼.... 13번째 나오는 수는 뭐까?"

영일이는 프로그램을 만들어 더 큰 수도 자동으로 계산하고 싶어졌다.

시작 값(a), 등비(r), 몇 번째인지를 나타내는 정수(n)가 입력될 때
n번째 수를 출력하는 프로그램을 만들어보자.

◎ 입력 형식

시작 값(a), 등비의 값(r), 몇 번째 인지를 나타내는 정수(n)가
공백을 두고 입력된다. (모두 0 ~ 10)

◎ 출력 형식

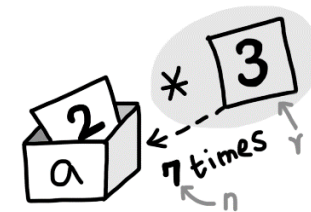
n번째 수를 출력한다.

입력 예시

2 3 7

출력 예시

1458



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

수 나열하기3

어떤 규칙에 따라 수를 순서대로 나열한 것을 수열이라고 한다.

예를 들어

1 -1 3 -5 11 -21 43 ... 은

1부터 시작해 이전에 만든 수에 -2를 곱한 다음 1을 더해 다음 수를 만든 수열이다.

이런 이상한 수열을 알게 된 영일이는 또 궁금해졌다.

"그럼.... 13번째 나오는 수는 뭐까?"

영일이는 물론 수학을 아주 잘하지만 이런 문제는 본 적이 거의 없었다...

그래서 프로그램을 만들어 더 큰 수도 자동으로 계산하고 싶어졌다.

시작 값(a), 곱할 값(m), 더할 값(d), 몇 번째인지를 나타내는 정수(n)가 입력될 때,
n번째 수를 출력하는 프로그램을 만들어보자.

◎ 입력 형식

시작 값(a), 곱할 값(m), 더할 값(d), 몇 번째 인지를 나타내는 정수(n)가
공백을 두고 입력된다. (a, m, d는 -50 ~ +50, n은 10이하의 자연수)

◎ 출력 형식

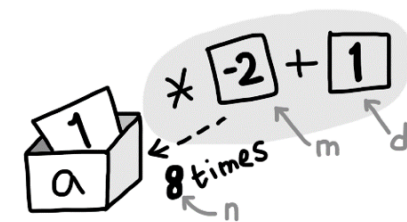
n번째 수를 출력한다.

입력 예시

1 -2 1 8

출력 예시

-85



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

함께 문제 푸는 날1

온라인 채점시스템에는 초등학생, 중고등학생, 대학생, 대학원생, 일반인, 군인, 프로그래머, 탑코더 등 아주 많은 사람들이 들어와 문제를 풀고 있는데,

매시간의 실시간 채점 정보는 메뉴의 Judge Status 를 통해 살펴볼 수 있다.

자! 여기서...잠깐..

같은 날 동시에 가입한 3명의 사람들이 온라인 채점시스템에 들어와 문제를 푸는 날짜가 매우 규칙적이라고 할 때, 다시 모두 함께 문제를 풀게 되는 그날은 언제일까?

예를 들어 3명이 같은 날 가입/등업하고, 각각 3일마다, 7일마다, 9일마다 한 번씩 들어온다면, 처음 가입하고 63일 만에 다시 3명이 함께 문제를 풀게 된다.

갑자기 힌트?

왠지 어려워 보이지 않는가?

수학에서 배운 최소공배수를 생각한 사람들도 있을 것이다. 하지만, 정보에서 배우고 경험하는 정보과학의 세상은 때때로 컴퓨터의 힘을 빌려 간단한 방법으로 해결할 수 있게 한다.

아래의 코드를 읽고 이해한 후 도전해 보자.

day는 날 수, a/b/c는 방문 주기이다.

```
...
day=1;
while(day%a!=0 || day%b!=0 || day%c!=0) day++; //이게 무슨 의미일까?
printf("%d", day);
...
```

물론, 아주 많은 다양한 방법이 있을 수 있다.

정보과학의 문제 해결에 있어서 정답은?

하나가 아니라 주어진 시간/기억공간으로 정확한 결과를 얻을 수 있는 모든 방법이다.

따라서, 모든 문제들에는 정답이 하나뿐만이 아니다.

새로운, 더 빠른, 더 간단한 방법을 다양하게 생각해보고 도전해 보기 바란다.

◎ 입력 형식

같은 날 동시에 가입한 인원 3명이 규칙적으로 방문하는, 방문 주기가 공백을 두고 입력된다. (단, 입력값은 100이하의 자연수이다.)

◎ 출력 형식

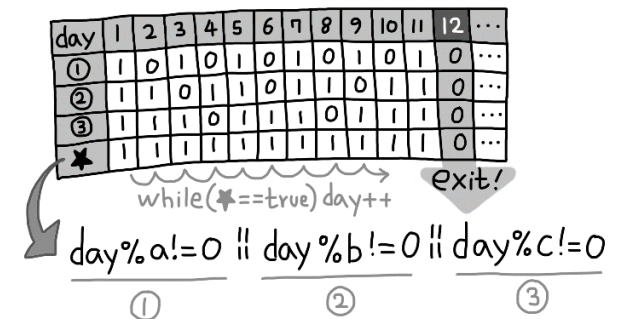
3명이 다시 모두 함께 방문해 문제를 풀어보는 날(동시 가입/등업 후 며칠 후?)을 출력한다.

입력 예시

3 7 9

출력 예시

63



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

이상한 출석 번호 부르기1

정보과학 선생님은 수업을 시작하기 전에 이상한 출석을 부른다.

선생님은 출석부를 보고 번호를 부르는데,
학생들의 얼굴과 이름을 빨리 익히기 위해 번호를 무작위(랜덤)으로 부른다.

그리고 얼굴과 이름이 잘 기억되지 않는 학생들은 번호를 여러 번 불러
이름과 얼굴을 빨리 익히려는 것이다.

출석 번호를 n번 무작위로 불렀을 때, 각 번호(1 ~ 23)가 불린 횟수를 각각 출력해보자.

참고

각 번호가 불린 횟수를 기록하고 출력하기 위해 변수 23개를 선언할 수도 있다.
하지만 C언어에서는 같은 이름에 번호를 붙여 사용하는 배열(array)을 사용할 수 있다.
이는 마치 아파트의 동 호수(예를 들어 a동 101호)와 같이 번호를 붙여 집을 구분하는 것과 비슷하다.

예를 들어 a동 101호를 다르게 표현하면 a[101] 과 같은 표현이 가능한 것이다.
이렇게 번호를 붙여 데이터를 저장할 수 있는 변수인 배열을 사용하기 위해서는
변수처럼, 사용하기 전에 선언해주어야 하는데 아래와 같은 방법으로 가능하다.

예시

```
int a[100];           //a[0] ~ a[99] 까지 정수를 저장할 수 있는 배열을 만들어라.
for(i=0; i<100; i++)
{
    scanf("%d", &a[i]); // 각각의 방에 순서대로 반복하면서 값을 입력해라.
}
```

배열을 사용하기 전에 배열에 들어있는 값을 초기화시키는 것이 좋은데,
다음과 같은 여러 가지 방법이 있다.

```
int a[24]={};          //0번부터 23번까지 모두 0으로 초기화된다.
int a[24]={1,2,3};     //1,2,3 이 순서대로 들어가고 나머지는 모두 0으로 초기화된다.
int a[3]={1,2,3};      //1,2,3 이 순서대로 저장된다.
int a[3]={1,2,3,4};    //방을 3개 만들고 값을 4개를 집어넣어라? 오류 발생!
```

문제 해결을 위한 참고 코드

```
int n, i, t;
int a[24]={};
scanf("%d", &n);           //개수 입력받기
for(i=1; i<=n; i++)       //개수만큼 입력받기
{
    scanf("%d", &t);       //읽어서
    a[t]=a[t]+1;          //들어있던 값에 1만큼 더해 다시 저장. a[t]+=1 과 같다.
}

for(i=1; i<=23; i++)
{
    printf("%d ", a[i]); //1~23 번 배열에 저장되어있는 값 출력하기
}
```

◎ 입력 형식

첫 번째 줄에 출석 번호를 부른 횟수인 정수 n이 입력된다. (1 ~ 10000)
두 번째 줄에는 무작위로 부른 n개의 번호(1 ~ 23)가 공백을 두고 순서대로 입력된다.

◎ 출력 형식

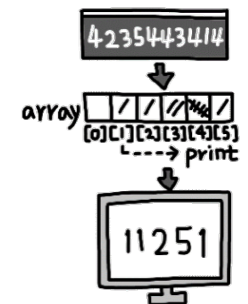
1번부터 번호가 불린 횟수를 순서대로 공백으로 구분하여 한 줄로 출력한다.

입력 예시

10
1 3 2 2 5 6 7 4 5 9

출력 예시

1 2 1 1 2 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

이상한 출석 번호 부르기2

정보과학 선생님은 수업을 시작하기 전에 이상한 출석을 부른다.

학생들의 얼굴과 이름을 빨리 익히기 위해 번호를 무작위(랜덤)으로 부르는데,
영일이는 선생님이 부른 번호들을 기억하고 있다가 거꾸로 불러보는 것을 해보고 싶어졌다.

출석 번호를 n번 무작위로 불렀을 때, 부른 번호를 거꾸로 출력해 보자.

참고
배열에 순서대로 기록해 두고, 기록된 내용을 거꾸로 출력하면 된다.

```
예시
int n, i;
int a[1000]={};
scanf("%d", &n);      //개수 입력 받기
for(i=1; i<=n; i++)    //개수 만큼 입력 받기
    scanf("%d", &a[i]); //읽어서 순서대로 배열에 넣는다.

for(i=n; i>=1; i--)
    printf("%d ", a[i]); //i 번 배열에 저장되어 있는 값 출력하기
```

◎ 입력 형식

번호를 부른 횟수(n, 1 ~ 10000)가 첫 줄에 입력된다.
n개의 랜덤 번호(k, 1 ~ 23)가 두 번째 줄에 공백을 사이에 두고 순서대로 입력된다.

◎ 출력 형식

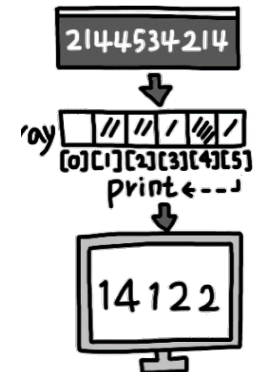
출석을 부른 번호 순서를 바꾸어 공백을 두고 출력한다.

입력 예시

```
10
10 4 2 3 6 6 7 9 8 5
```

출력 예시

```
5 8 9 7 6 6 3 2 4 10
```



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

이상한 출석 번호 부르기3

정보과학 선생님은 오늘도 이상한 출석을 부른다.

영일이는 오늘도 다른 생각을 해보았다.

출석 번호를 다 부르지는 않은 것 같은데... 가장 빠른 번호가 뭐였지?

출석 번호를 n번 무작위로 불렀을 때, 가장 빠른 번호를 출력해 보자.

참고

배열에 순서대로 기록해 두면, 기록된 내용을 모두 검사해 가장 작은 값을 찾아내면 된다.

그런데, 가장 작은 값은 어떻게 어떤 것과 비교하고 찾아야 할까?

◎ 입력 형식

번호를 부른 횟수(n, 1 ~ 10000)가 첫 줄에 입력된다.

n개의 랜덤 번호(k, 1 ~ 23)가 두 번째 줄에 공백을 사이에 두고 순서대로 입력된다.

◎ 출력 형식

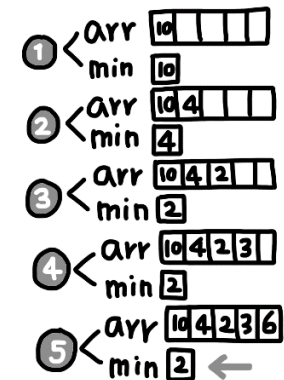
출석을 부른 번호 중에 가장 빠른 번호를 1개만 출력한다.

입력 예시

10
10 4 2 3 6 6 7 9 8 5

출력 예시

2



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29

바둑판에 흰 돌 놓기

기숙사 생활을 하는 학교에서 어떤 금요일(전원 귀가일)에는 모두 집으로 귀가를 한다.

오랜만에 집에 간 영일이는 아버지와 함께 두던 매우 큰 오목에 대해서 생각해 보다가 "바둑판에 돌을 올린 것을 프로그래밍 할 수 있을까?"하고 생각하였다.

바둑판($19 * 19$)에 n 개의 흰 돌을 놓는다고 할 때,
 n 개의 흰 돌이 놓인 위치를 출력하는 프로그램을 작성해보자.

참고

가로번호, 세로번호를 사용할 수 있는 2차원 배열을 사용하면
이러한 형태를 쉽게 기록하고 사용할 수 있다. 물론 더 확장한 n 차원 배열도 만들 수 있다.

예시

```
int n, i, j, x, y;
int a[20][20]={};
scanf("%d", &n);
for(i=1; i<=n; i++)
{
    scanf("%d %d", &x, &y);
    a[x][y]=1;
}
for(i=1; i<=19; i++)    //한 줄(위에서 아래로) 씩
{
    for(j=1; j<=19; j++)    //한 열(왼쪽에서 오른쪽으로) 씩
    {
        printf("%d ", a[i][j]); //값 출력
    }
    printf("\n");           //줄 바꾸기
}
```

◎ 입력 형식

바둑판에 올려 놓을 흰 돌의 개수(n)가 첫 줄에 입력된다.

둘째 줄 부터 $n+1$ 번째 줄까지 흰 돌을 놓을 좌표(x, y)가 n 줄 입력된다.

n 은 10이하의 자연수이고 x, y 좌표는 1 ~ 19 까지이며, 같은 좌표는 입력되지 않는다.

◎ 출력 형식

흰 돌이 올려진 바둑판의 상황을 출력한다.

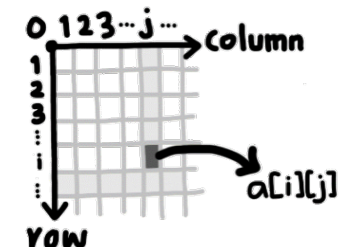
흰 돌이 있는 위치는 1, 없는 곳은 0으로 출력한다.

입력 예시

5
11
22
33
44
55

출력 예시

10000000000000000000
01000000000000000000
00100000000000000000
00010000000000000000
00001000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

바둑알 십자 뒤집기

부모님을 기다리던 영일이는 검정/흰 색 바둑알을 바둑판에 꽂 채워 깔아 놓고 놀다가...

"십(+)자 뒤집기를 해볼까?"하고 생각했다.

바둑판(19 * 19)에 흰 돌(1) 또는 검정 돌(0)이 모두 꽂 채워져 놓여있을 때,
n개의 좌표를 입력받아 십(+)자 뒤집기한 결과를 출력하는 프로그램을 작성해보자.

참고
가로 번호, 세로 번호를 사용할 수 있는 2차원 배열을 사용하면
이러한 형태를 쉽게 기록하고 사용할 수 있다. 물론 더 확장한 n차원 배열도 만들 수 있다.

```
예시
int n, i, j, x, y;
int a[20][20]={};
for(i=1; i<=19; i++) //한 줄씩 바둑판 상황 입력 받기
    for(j=1; j<=19; j++)
        scanf("%d", &a[i][j]);

scanf("%d", &n); //좌표 개수 입력받기

for(i=1; i<=n; i++) //좌표의 개수만큼
{
    scanf("%d %d", &x, &y);
    for(j=1; j<=19; j++) //가로 줄 흑<->백 바꾸기
    {
        if(a[x][j]==0) a[x][j]=1;
        else a[x][j] = 0;
    }

    for(j=1; j<=19; j++) //세로 줄 흑<->백 바꾸기
    {
        if(a[j][y]==0) a[j][y]=1;
        else a[j][y] = 0;
    }
}
...
```

◎ 입력 형식
바둑알이 깔려 있는 상황이 19 * 19 로 입력된다.
십자 뒤집기 횟수(n)가 입력된다.
십자 뒤집기 좌표가 횟수(n) 만큼 입력된다. 단, n은 10이하의 자연수이다.

◎ 출력 형식
십자 뒤집기 결과를 출력한다.

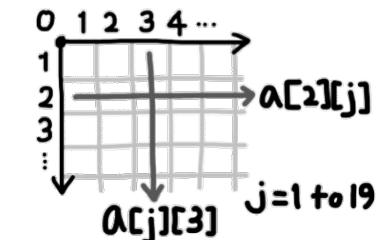
입력 예시

```
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
11111111111111111111
00000000001010000000
11111111111111111111
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
00000000001010000000
2
10 10
12 12
```

출력 예시

```
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000001010000000
00000000000000000000
00000000001010000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
```

input data: 2 3



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

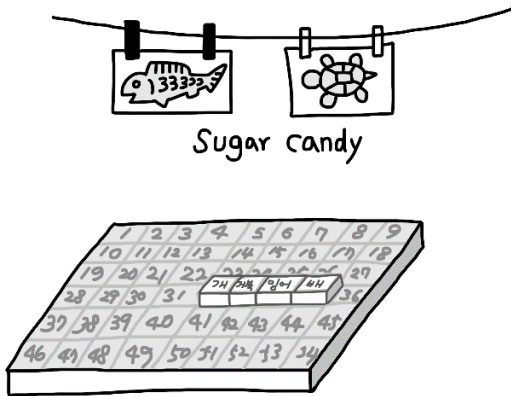
01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

설탕과자 뽑기

부모님과 함께 유원지에 놀러간 영일이는
설탕과자(설탕을 녹여 물고기 등의 모양을 만든 것) 뽑기를 보게 되었다.

길이가 다른 몇 개의 막대를 바둑판과 같은 격자판에 놓는데,

막대에 있는 설탕과자 이름 아래에 있는 번호를 뽑으면 설탕과자를 가져가는 게임이었다.
(잉어, 봉어, 용 등 여러 가지가 적혀있다.)



격자판의 세로(h), 가로(w), 막대의 개수(n), 각 막대의 길이(l),
막대를 놓는 방향(d:가로는 0, 세로는 1)과
막대를 놓는 막대의 가장 왼쪽 또는 위쪽의 위치(x, y)가 주어질 때,

격자판을 채운 막대의 모양을 출력하는 프로그램을 만들어보자.

◎ 입력 형식

첫 줄에 격자판의 세로(h), 가로(w)가 공백을 두고 입력되고,
두 번째 줄에 놓을 수 있는 막대의 개수(n)
세 번째 줄부터 각 막대의 길이(l), 방향(d), 좌표(x, y)가 입력된다.

입력값의 정의역은 다음과 같다.

```
1 <= w, h <= 100
1 <= n <= 10
d = 0 or 1
1 <= x <= 100-h
1 <= y <= 100-w
```

◎ 출력 형식

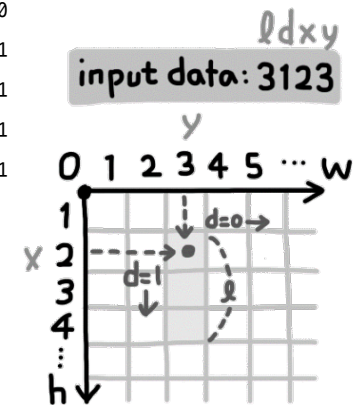
모든 막대를 놓은 격자판의 상태를 출력한다.
막대에 의해 가려진 경우 1, 아닌 경우 0으로 출력한다.
단, 각 숫자는 공백으로 구분하여 출력한다.

입력 예시

```
5 5
3
2 0 1 1
3 1 2 3
4 1 2 5
```

출력 예시

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



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

성실한 개미

영일이는 생명과학에 관심이 생겨 왕개미를 연구하고 있었다.

왕개미를 유심히 살펴보던 중 특별히 성실해 보이는 개미가 있었는데,
그 개미는 개미굴에서 나와 먹이까지 가장 빠른 길로 이동하는 것이었다.

개미는 오른쪽으로 움직이다가 벽을 만나면 아래쪽으로 움직여 가장 빠른 길로 움직였다.
(오른쪽에 길이 나타나면 다시 오른쪽으로 움직인다.)

이에 호기심이 생긴 영일이는 그 개미를 미로 상자에 넣고 살펴보기 시작하였다.

미로 상자에 넣은 개미는 먹이를 찾았거나, 더 이상 움직일 수 없을 때까지
오른쪽 또는 아래쪽으로만 움직였다.

미로 상자의 구조가 0(갈 수 있는 곳), 1(벽 또는 장애물)로 주어지고,
먹이가 2로 주어질 때, 성실한 개미의 이동 경로를 예상해보자.

단, 맨 아래의 가장 오른쪽에 도착한 경우, 더 이상 움직일 수 없는 경우, 먹이를 찾은 경우
에는 더이상 이동하지 않고 그 곳에 머무른다고 가정한다.

미로 상자의 테두리는 모두 벽으로 되어 있으며,
개미집은 반드시 (2, 2)에 존재하기 때문에 개미는 (2, 2)에서 출발한다.

◎ 입력 형식

10*10 크기의 미로 상자의 구조와 먹이의 위치가 입력된다.

◎ 출력 형식

성실한 개미가 이동한 경로를 9로 표시해 출력한다.

입력 예시1

```
1 1 1 1 1 1 1 1 1 1
1 0 0 1 0 0 0 0 0 1
1 0 0 1 1 1 0 0 0 1
1 0 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 1 0 1
1 0 0 0 0 1 0 0 0 1
1 0 0 0 0 1 2 0 0 1
1 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```

출력 예시1

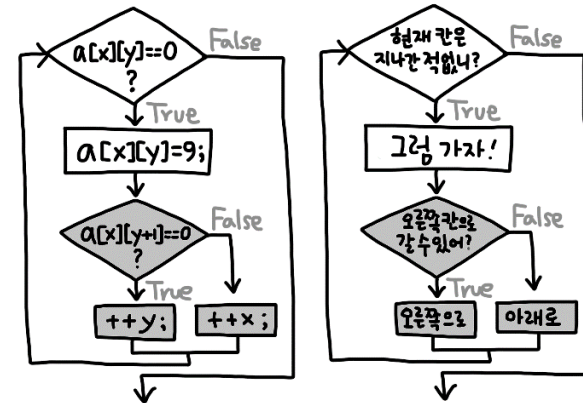
```
1 1 1 1 1 1 1 1 1 1
1 9 9 1 0 0 0 0 0 1
1 0 9 1 1 1 0 0 0 1
1 0 9 9 9 9 9 1 0 1
1 0 0 0 0 0 9 1 0 1
1 0 0 0 0 1 9 9 9 1
1 0 0 0 0 1 2 0 9 1
1 0 0 0 0 1 0 0 9 1
1 0 0 0 0 0 0 0 9 1
1 1 1 1 1 1 1 1 1 1
```

입력 예시2

```
1 1 1 1 1 1 1 1 1 1
1 0 0 1 0 0 0 0 0 1
1 0 0 1 1 1 0 0 0 1
1 0 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 1 0 1
1 0 0 0 0 1 0 1 0 1
1 0 0 0 0 1 2 1 0 1
1 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```

출력 예시2

```
1 1 1 1 1 1 1 1 1 1
1 9 9 1 0 0 0 0 0 1
1 0 9 1 1 1 0 0 0 1
1 0 9 9 9 9 9 1 0 1
1 0 0 0 0 0 9 1 0 1
1 0 0 0 0 1 9 1 0 1
1 0 0 0 0 1 9 1 0 1
1 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

찍어 맞춰라??

답은 정수 0 ~ 255 중 하나이다.
뭘까? 찍어 맞춰보자.

답을 추측해 그 값만 출력하면 된다.
(답을 맞춘 사람은 절대로 다른 사람에게 답을 알려주지 말자! 답은 정기적으로? 바뀐다.)

중요 : 답이 틀리면 답과 비교된 메시지가 출력된다.
메시지(예)
Your Answer is Greater than Correct Answer

운이 아주 아주 좋으면 1번에 맞출 것이고,
0 부터 255까지 차례대로 시도해 본다면, 최악의 경우 256번 만에 맞출 것이다.
답을 추측하는 계획을 잘~ 세우지 않고 이것저것 찍어본다면 매우 오래 걸릴 수도 있다.

"포기하지 않고 도전하는 사람만이 세상을 이끌어갈 수 있습니다."
온라인 채점 사이트에서 많은 문제를 해결한 사람들은 랭킹이 올라간다.
Rank 메뉴에서 다른 사람들의 아이디를 눌러보자.

◎ 입력 형식
입력 없음

◎ 출력 형식
정답을 출력한다.

입력 예시

출력 예시

3



Thought && (Idea || Curiosity)

(Offline Coding) || (Kernel Codes)

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	

- c언어 기초 100제 -

본 오프라인 기초100제 노트북(공책)의

초판 인쇄를 위해 관심과 지원을 해주신 전국 정보 선생님들께 감사드립니다.

김지혜, 남상천, 최웅선, 이승현, 제종철, 이묘정, 김지은, 김춘희, 현태익, 윤만영
김현준, 채길순, 김종순, 김형기, 박미정, 김영주, 노경보, 박정화, 안상진, 맹삼호
류은미, 신혜정, 황은숙, 오윤정, 김태욱, 장윤숙, 김서영, 이화정, 한경철, 이상호
박정희, 조혜영, 이다현, 최인범, 전준호, 박지현, 남상숙, 이은혜, 박경희, 김미정
문지은, 한숙희, 양진영, 조용구, 서은화, 김주리, 조승조, 안문옥, 조은형, 이현아
신미경, 이정철, 이영미, 배성환, 최명진, 김지영, 원인선, 하은영, 이효진, 이시웅
권영아, 이은경, 허민택, 하정숙, 김희정, 서정민, 강 진, 최윤경

- 고작해야 한 명... 너무나도 힘들고 지쳤었지만...

- 지금 바로 이순간까지...

- 끝까지 포기하지 않으시고, 함께 노력하고 공부하며...

- 전세계 어느 정보(컴퓨터과학) 교사도 가보지 않은 길들을 한 걸음씩 걸어온...

- 한국의 모든 중고등학교 정보 선생님들에게 감사드립니다.

- 뭐 늘어죽기 전에는 뭔가 바뀌겠조~ 그때까지 까짓거! 함께 고생하시죠!?

- 2016년 10월 어느 새벽... v0.0 작업을 마무리 하며...



Computer Science & Love ™

Computer Science teachers' League ™

한국 컴퓨터과학교사 교과연구 카페/동호회 Since 2008



- 본 오프라인 노트북에 사용된 모든 그림의 저작권은 김지혜 선생님에게 있습니다.

🔗 A HUNDRED PROBLEMS :: C (ver 1.0)

name
