

질문 잘하는 법

파트

목차

- 1부: 컴퓨터 질문 전반
 - 기술적인 문제를 해결하는 방법
- 2부: 백준 질문
 - 문제 풀이에서 막혔을 때 해결하는 방법

1부: 컴퓨터 질문 전반

기술적인 문제를 해결하는 방법을 알아보시다.

질문하기 전에

스스로 답을 찾기 위한 시도를 하셨습니까?

아직 하지 않았다면, **반드시 먼저 하세요.**

스스로 답을 찾아봅시다.

구글 검색을 꼭 해봅시다.

당신이 겪는 대부분의 문제는...

- 수많은 이가 이미 겪었습니다.
- 수많은 이가 이미 질문했습니다.
- 수많은 이가 거기에 답변했습니다.
- 심지어 공식 매뉴얼이 있는 경우도 있습니다.
- 이것을 대부분 **검색으로 찾을 수 있습니다.**

검색해도 안 찾아져요!

높은 확률로 검색을 잘못된 경우입니다.

- **구글**에서 검색하세요.
 - 국내 포털은 기술자료 검색에 적합하지 않습니다.
- 영어로 검색하면 훨씬 좋습니다.
 - 영어 자료의 질과 양이 압도적입니다.
 - 어렵다면 번역기의 도움을 받아봅시다.

검색해도 안 찾아져요!

- 구체적으로 검색하세요.
 - 특정 환경에서만 생기는 문제가 많습니다.
- 정확한 용어를 알면 검색이 훨씬 수월합니다.
- 에러가 발생했다면 **에러 메시지를 그대로 복사**해서 검색하세요.

믿을만한 자료를 찾아보세요.

- 블로그 글은 부정확한 경우가 종종 있습니다.
- 공식 문서가 있다면 최우선으로 참고하세요.
 - 검색어 뒤에 'documentation'을 붙여서 검색해보세요.
- StackOverflow를 참고하세요.
 - 특히 추천을 많이 받은 글은 신뢰도가 높습니다.

분석을 더 해봅시다.

당신이 문제를 해결하기 위해 시도하고 분석한 모든 것이 문제 해결에 도움이 됩니다.

스스로 해결하지 못했더라도 그렇습니다.

스스로 해결이 안되었다면
이제 질문의 시간입니다.

스스로 답을 찾지 못했대구요?

당연히 스스로 답을 찾지 못할 수 있습니다.

그럴 때 질문하는 것은 부끄러운 일이 아닙니다.

자신감을 가지고 질문합시다.

질문자와 답변자는 한 팀입니다.

문제 해결을 위해 눈물나게 고생하신 걸 압니다.

답변자에게 그 **과정을 전부 공유해주세요.**

이는 '노력하는 모습을 보고 싶다'는 감정적인 이유가 아닙니다.

그 과정이 실제로 답변자가 문제를 파악하고 답변하는 데에 **꼭 필요한 정보를 제공하기** 때문입니다.

불필요한 서두를 빼세요.

~~"혹시 질문해도 되나요?"~~

~~"~~ 잘 아는 분 있나요?"~~

시간을 절약합니다.

온라인 상의 대화에는 필연적으로 **매우 긴 딜레이**가 발생합니다.

당신이 불필요한 서두를 던지는 순간 답변자는 당신이 원래 하고자 했던 질문을 하기까지 무한정 기다릴 수밖에 없습니다.

양쪽 모두에게 불필요한 시간 낭비입니다.

사람을 찾지 마세요.

~~ 아는 분 있냐고 말문을 여는 경우가 많습니다.

제발 하지 마세요.

설령 있다고 하더라도 나오지 않을 겁니다.

당신이 질문하고 싶었던 내용을 바로 질문하세요.

해결하고 싶은
근본적인 문제를
질문하세요.

이런 상황 겪어보지 않으셨나요?

파일의 확장자를 출력하고 싶습니다.

".txt", ".exe", ".png" 등등 확장자는 다 세 글자니까,
"파일 이름 뒤 세 글자를 얻는 방법"만 알면 해결될
것 같습니다. (정말 그럴까요?)

그런데 이건 어떻게 할까요? 모르니 질문합니다.

이제 답변자 입장에서...

"파일 이름 뒤 세 글자를 출력하는 방법"을 누군가 질문했습니다.

도대체 이걸 어디에 쓰려는 걸까요? 잘 모르겠지만 답변은 얼마든지 해줄 수 있습니다.

```
$ foo=abcdef  
$ echo ${foo: -3}  
def
```

다시 질문자 입장에서...

으악, 이걸 예상하지 못했는데요!

```
$ foo=dog.jpeg  
$ echo ${foo: -3}  
peg
```

근본적인 문제를 질문합니다.

해결하고 싶었던 근본적인 문제가 무엇이었죠?

"파일의 확장자를 출력하기"

그러면, 그걸 그대로 질문했으면 좋았을텐데요!

```
$ foo=dog.jpeg  
$ echo ${foo##*.  
jpeg
```

최대한 많은 정보를 주세요.

"Y를 어떻게 하나요?" 라고 질문하면...

Y를 하는 방법에 대한 답변을 받을 수 있습니다.

"X를 해결하기 위해 Y를 하면 될 것 같은데 안됩니다. 어떻게 하나요?" 라고 질문하면...

X를 해결하는 일반적으로 더 나은 방법 Z가 있다면 Z를 안내받을 수도 있습니다.

Y만 질문했을 때는 전혀 얻지 못하는 것이죠!

최대한 많은 정보를 주세요.

" x 를 해결하기 위해 일반적으로 z 를 하는 것을 알지만 이러한 이유로 z 가 안됩니다. 대신 y 를 하면 될 것 같은데 이러한 에러 때문에 잘 안됩니다. 어떻게 하나요?" 라고 질문하면...

답변자는 당신의 제약 상황을 전부 이해한 상황에서 x 를 해결하는 데 도움을 줄 수 있습니다. y 를 더 파고들거나, 제약 상황에 적합한 w 를 제시할 수도 있죠.

검색할 때와 마찬가지로...

- 에러가 발생했다면 **에러 메시지를 그대로 복사** 해서 알려주세요.
- 정확한 용어를 안다면 이를 사용해주세요.
- 잘 모르겠다면 현상을 정확하게 설명해주세요.
- 어떤 환경에서 문제가 발생했는지 알려주세요.

무엇을 검색했는지 알려주세요.

- 어떤 키워드로 검색했는지 알려주세요.
 - 키워드가 잘못되어 검색이 안되는 일이 흔합니다.
 - 답변자가 올바른 키워드를 알려줄 수 있습니다.
- 찾아본 페이지의 링크를 주면 더욱 좋습니다.
 - 답변자의 검색 시간을 획기적으로 줄여줍니다.

검색의 결과를 공유해주세요.

당신이 찾아본 문서가 왜 당신의 문제를 해결하지 못했는지 그 이유를 알려주세요.

그 이유는 당신만 알고 있는 제약 상황, 또는 당신의 컴퓨터가 놓인 환경에 대한 귀중한 정보를 제공합니다.

이해가 안된 부분을 설명하세요.

검색으로 찾은 문서가 이해가 안갈 수 있습니다.

이때 그냥 '이해가 안된다'고만 하면 곤란합니다.

구체적으로 **어느 부분이 어떤 이유로** 이해가 안갔는지를 설명해야 답변자도 맞춰서 설명할 수 있습니다.

답변자를 존중해주세요.

답변을 받았다면 경청해주세요.

더 많은 정보를 요청받았다면 제공해주세요.

무언가를 해보길 요청받았다면 그대로 하세요.

답변자는 당신을 괴롭히려는 것이 아니라,

당신의 문제를 해결하는 것을 돕고 싶은 겁니다.

결과를 알려주세요.

해결이 되었다면 해결이 되었음을 알려주세요.

해결이 안 되었다면 해결이 안됐다고 알려주세요.

답변자는 본인과는 아무런 상관도 없는 당신의 문제에 시간을 쏟았습니다.

감사인사를 덧붙여주면 정말 고맙겠지만,

최소한 문제가 해결이 되었는지는 좀 알려주세요.

2부: 백준 질문

문제 풀이에 막혔을 때 해결하는 방법을 알아보시다.

1부의 내용을 먼저 읽어주세요.

"문제를 해결하는 것"이 목표인 상황에서 할 일은 1부의 내용과 크게 다르지 않습니다.

2부는 온라인 저지의 특성과 "학습"이라는 측면에서 1부와 달라지는 문제 해결 방법에 집중합니다.

이를 위해서는 1부의 내용을 숙지해야 합니다.

질문하기 전에

스스로 답을 찾기 위한 시도를 하셨습니까?

아직 하지 않았다면, 반드시 먼저 하세요.

이번에도 다르지 않습니다.

백준 시스템을 이해합니다.

공지사항을 읽어보세요.

<https://help.acmicpc.net/question>

전부 정독하세요.

아래에서 설명할 내용은 위 링크에서 찾을 수 있는
내용의 부분집합일 뿐입니다.

채점 결과를 정확히 이해합니다.

내가 제출한 코드는 단순히 맞거나 틀릴 뿐일까요?

아닙니다.

어떻게 틀렸는지에 따라 접근을 달리해야 합니다.

틀렸습니다, 시간 초과, 메모리 초과, 출력 초과, 런타임 에러, 컴파일 에러 등의 채점 결과가 제공되며, 이는 전부 다릅니다.

컴파일 에러는 클릭하세요.

5 1000번 제출 맞힌 사람 슛코딩 재채점 결과 채점 현황 **내 제출** 난

[NEW] 코딩 테스트 준비 온라인 강의 할인 중 / 코딩 테스트 및 인터뷰 온라인 강의

1000 lcr7324

제출 번호	아이디	문제	결과
73470293	lcr7324	5 1000	컴파일 에러
73470255	lcr7324	5 1000	맞았습니다!!

1

클릭!

2

클릭!

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main() {
6     int a, b;
7     cin >> a >> b;
8     cout << a + b << '\n'
9
10     return 0;
11 }
```

! error: expected ';' before 'return'

```
Main.cc: In function 'int main()':
Main.cc:8:26: error: expected ';' before 'return'
   8 |     cout << a + b << '\n'
     |                                     ^
     |                                     ;
   9 |
  10 |     return 0;
     |     ~~~~~
```

컴파일 에러의 원인을 확인할 수 있습니다.

에러 메시지가 무슨 뜻인지 모르겠다고요?

1부에서 말했듯이, 에러 메시지를 그대로 복사해서 검색하세요.

틀렸습니다: 예제는 맞나요?

- 내가 제출한 코드를 **그대로** 실행해보았나요?
 - 실행은 정말 해보신거죠?
 - 디버깅을 위해 변경한 것이 있지는 않나요?
- 예제를 **그대로** 입력했나요?
 - 띄어쓰기와 줄바꿈을 구별했나요?
- 예제 출력과 **완전히** 일치하나요?
 - 한 글자라도 다르면 **틀립니다**. (스페셜 저지는 예외)

실행 환경이 올바른가요?

- 특히 윈도우 환경에서 Visual Studio를 사용하는 경우, 백준 채점 환경인 gcc에서 사용할 수 없는 `scanf_s` 함수를 사용하는 등의 문제가 종종 있습니다.
- 본인의 개발 환경이 [백준 채점 환경](#)과 다르다면, [tio.run](#) 등의 온라인 컴파일러에서 한번 더 돌려보세요.

예제를 그대로 입력하는 방법

N+2번째 줄부터 N+3번째 줄까지, T번째 줄의 정답을 출력 예제의 정답에 모두 출력한다.

N+7번째 줄에, 게임의 결과를 본문에 설명한 형식대로 출력한다.

아래 예제 중 첫 세 개는 가능한 세 가지 게임 결과에 대한 출력 형식을 모두 담고 있다.

예제 입력 1 복사

```
7 8
.&....&
..B.&..&
B...&...
.B@.B#..
.&....M.
.B...B..
..B^^&..
RRRUULLLULUDDDLDRDRDRRRURRULUULLU
3 5 One 4 2 10 3
2 5 Two 10 2 8 2
```

클릭!

예제를 그대로 입력하는 방법

tio.run

```
C++ (gcc)
▶ Compiler flags
▶ Header
▼ Code
// 여기에 작성한 코드 그대로 붙여넣기
▶ Footer
▼ Input
7 8
.&...&
..B.&..&
B...&...
.B@.B#..
.&...M.
.B...B..
..B^^&..
RRRUULLULUDDDLDRDRRRRURRULUULLU
3 5 One 4 2 10 3
2 5 Two 10 2 8 3
1 2 Three 20 2 14 7
```

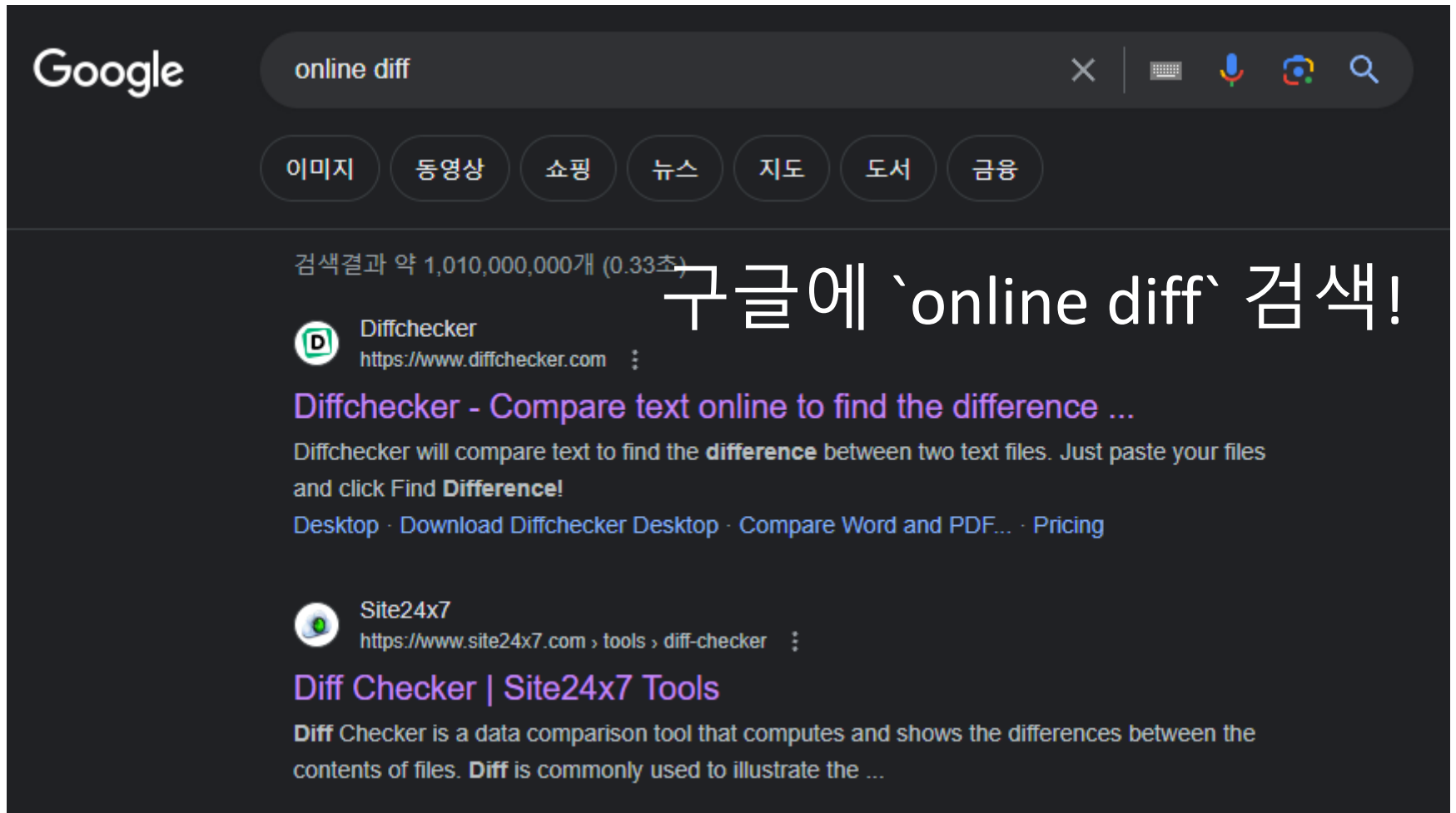
클릭 후 Ctrl + V!

출력이 일치하는지 확인하는 법

```
cout << H << M;  
return 0;  
}  
▶ Footer  
▼ Input  
10 10  
▶ Arguments  
▼ Output  
925  
▼ Det  
Real time: 0.554 s  
User time: 0.450 s  
Sys. time: 0.095 s  
CPU share: 98.35 %  
Exit code: 0
```

전체
복사!

출력이 일치하는지 확인하는 법



Google online diff

이미지 동영상 쇼핑 뉴스 지도 도서 금융

검색결과 약 1,010,000,000개 (0.33초)

구글에 `online diff` 검색!

Diffchecker
<https://www.diffchecker.com>

Diffchecker - Compare text online to find the difference ...
Diffchecker will compare text to find the **difference** between two text files. Just paste your files and click Find **Difference!**
Desktop · Download Diffchecker Desktop · Compare Word and PDF... · Pricing

Site24x7
<https://www.site24x7.com/tools/diff-checker>

Diff Checker | Site24x7 Tools
Diff Checker is a data comparison tool that computes and shows the differences between the contents of files. Diff is commonly used to illustrate the ...

tio.run

```
cout << H << M;  
return 0;  
}  
▶ Footer  
▼ Input  
10 10  
▶ Arguments  
▼ Output  
925  
▼ Debug  
Real time: 0.554 s  
User time: 0.450 s  
Sys. time: 0.095 s  
CPU share: 98.35 %  
Exit code: 0
```

1 복사!

예제 출력 1 복사

3 클릭!

```
9 25
```

Edit texts ...

Switch texts

Compare!

Clear all

Paste one version of a text here.

Paste another version of the text here.

2 붙여넣기!

5
클릭!

4 붙여넣기!

text-compare.com

```
1 어느 한 컴퓨터공학과 학생이 유명한 교수님을 찾아가 물었다.
2 "재귀함수가 뭐가요?"
3 "잘 들어보게. 옛날옛날 한 산 꼭대기에 이세상 모든 지식을 통달한 선인이 있었어.
4 마을 사람들은 모두 그 선인에게 수많은 질문을 했고, 모두 지혜롭게 대답해 주었지.
5 그의 답은 대부분 옳았다고 하네. 그런데 어느 날, 그 선인에게 한 선비가 찾아와서 물었어."
6 ---- "재귀함수가 뭐가요?"
7 ---- "잘 들어보게. 옛날옛날 한 산 꼭대기에 이세상 모든 지식을 통달한 선인이 있었어.
8 ---- 마을 사람들은 모두 그 선인에게 수많은 질문을 했고, 모두 지혜롭게 대답해 주었지.
9 ---- 그의 답은 대부분 옳았다고 하네. 그런데 어느 날, 그 선인에게 한 선비가 찾아와서 물었어."
10 ----- "재귀함수가 뭐가요?"
11 ----- "재귀함수는 자기 자신을 호출하는 함수라네"
12 ----- "라고 답변하였지."
13 ----- "라고 답변하였지."
14 "라고 답변하였지."
```

```
1 어느 한 컴퓨터공학과 학생이 유명한 교수님을 찾아가 물었다.
2 "재귀함수가 뭐가요?"
3 "잘 들어보게. 옛날옛날 한 산 꼭대기에 이세상 모든 지식을 통달한 선인이 있었어.
4 마을 사람들은 모두 그 선인에게 수많은 질문을 했고, 모두 지혜롭게 대답해 주었지.
5 그의 답은 대부분 옳았다고 하네. 그런데 어느 날, 그 선인에게 한 선비가 찾아와서 물었어."
6 ---- "재귀함수가 뭐가요?"
7 ---- "잘 들어보게. 옛날옛날 한 산 꼭대기에 이세상 모든 지식을 통달한 선인이 있었어.
8 ---- 마을 사람들은 모두 그 선인에게 수많은 질문을 했고, 모두 지혜롭게 대답해 주었지.
9 ---- 그의 답은 대부분 옳았다고 하네. 그런데 어느 날, 그 선인에게 한 선비가 찾아와서 물었어."
10 ----- "재귀함수가 뭐가요?"
11 ----- "재귀함수는 자기 자신을 호출하는 함수라네"
12 ----- "라고 답변하였지."
13 ----- "라고 답변하였지."
14 "라고 답변하였지."
```

```
6 ---- "재귀함수가 뭐가요?"
7 ---- "잘 들어보게. 옛날옛날
8 ---- 마을 사람들은 모두 그 선
9 ---- 그의 답은 대부분 옳았다.
10 ----- "재귀함수가 뭐가요?"
11 ----- "재귀함수는 자기 자
12 ----- "라고 답변하였지."
13 ----- "라고 답변하였지."
14 "라고 답변하였지."
```

```
6 ---- "재귀함수가 뭐가요?"
7 ---- "잘 들어보게. 옛날옛날
8 ---- 마을 사람들은 모두 그
9 ---- 그의 답은 대부분 옳았다
10 ----- "재귀함수가 뭐가요
11 ----- "재귀함수는 자기 자
12 ----- "라고 답변하였지.
13 ----- "라고 답변하였지.
14 "라고 답변하였지."
```

두 텍스트가 다른 부분을 강조하여 표시해줍니다.

예제만 넣어본 건 아니가요?

- "맞았는데 왜 틀리죠?" (boj.kr/15820)
 - 예제만 맞았다고 다가 아닙니다.
- 입력 조건에 맞는 가장 큰 입력을 넣어보세요.
- 입력 조건에 맞는 가장 작은 입력을 넣어보세요.
- 입력 조건에 맞는 다양한 입력을 만들어보세요.
- 질문 게시판에서 반례를 찾아보세요.

시간 초과: 다음을 확인하세요.

- 시간복잡도가 무엇인지 알고 있나요?
 - 모른다면, 지금이 바로 학습이 필요한 시점입니다.
- 빠른 입출력을 하고 있나요?
 - 모른다면, "빠른 A+B" (boj.kr/15552)를 풀어봅시다.
- 혹시 Python3로 제출하고 있나요?
 - PyPy3로 제출하세요. [Python3는 느립니다.](#)

질문 게시판을 확인하세요.

3 2884번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

난이도 기여 ↗

질문 게시판

알람 시계

성공

다국어

클릭!

시간 제한

메모리 제한

제출

정답

1 초

128 MB

456542

172451

문제

상근이는 매일 아침 알람을 듣고 일어난다. 알람을 듣고 바로 일어나면 다행이겠지만, 항상 조금만 더 자려는 마음 때문

상근이는 모든 방법을 동원해보았지만, 조금만 더 자려는 마음은 그 어떤 것도 없앨 수가 없었다.

이런 상근이를 불쌍하게 보던 창영이는 자신이 사용하는 방법을 추천해 주었다.

바로 "45분 일찍 알람 설정하기"이다.

제목	카테고리	언어	글쓴이	댓글	👍	작성일
node.js 런타임 에러	공지		startlink	0	62	2년 전
질문글을 지우지 말아주세요.	오타/오역/요청		portableangel	14	355	5년 전
반례를 다해봤지만 99%에서 틀렸습니다가 뜨네요,,, 도와주세요ㅠㅠ	3 2884번 질문	C++17		4	0	1일 전
어디서 틀렸는지 모르겠어요 ㅜㅜ 해결	3 2884번 질문	C99		2	0	10일 전
어디서 틀렸는지 모르겠습니다! 해결	3 2884번 질문	C++17		1	0	10일 전
답은 맞게 나오는데 틀렸다고 나옵니다 ㅠㅠ 해결	3 2884번 질문	Python 3		3	0	13일 전
런타임 에러가 뜹니다...	3 2884번 질문	Python 3		1	0	17일 전
어디가 틀린걸까요? 해결	3 2884번 질문	C99		2	0	24일 전
이거 왜 틀린건가요?	3 2884번 질문	PyPy3		1	0	27일 전
if else 에서 if 로 바꾼거 뿐인데	3 2884번 질문	C++17		1	0	28일 전
반례좀 찾아주세요 ㅠㅠ	3 2884번 질문	Java 11		3	0	28일 전
정말로.. 틀린 이유를 못 찾겠습니다..! 해결	3 2884번 질문	node.js		2	0	29일 전
30번은 넘게 수정한거같은데 뭐가 틀린건지 모르겠습니다. 도와주십쇼	3 2884번 질문	C++17		3	0	29일 전
어느 부분에서 틀린건지 모르겠습니다ㅠㅠ	3 2884번 질문	C++17		1	0	29일 전
이거 어디서 틀린건가요?? 해결	3 2884번 질문			0	0	30일 전
왜 틀린건가요? 해결	3 2884번 질문	Python 3		2	0	1달 전
어디가 틀린지 모르						1달 전
틀린 이유좀 알려주						1달 전
왜 틀렸을까요 해결	3 2884번 질문	C99		2	0	1달 전
틀린 부분과 void 함수 질문	3 2884번 질문	C99		1	0	1달 전
틀린이유	3 2884번 질문	Python 3		2	0	1달 전
c언어 왜 안되는걸까요..	3 2884번 질문	C++17		1	0	1달 전

이 문제는 30 페이지 가량의 질문이 있습니다.

다른 사람의 질문을 읽어보세요.

코드를 읽지 말고 답변을 유심히 살펴보세요.

심지어 모르는 언어여도 괜찮습니다.

답변에 반례가 있다면, 그걸 그대로 돌려보세요.

답변에 틀린 부분에 대한 설명이 있다면, 그 설명이 본인의 코드에도 해당하는지 살펴보세요.

코드는 달라도 틀리는 이유는 대개 비슷합니다.

틀리는 이유는 대개 비슷합니다.

정말로요!

"알람 시계" (boj.kr/2884) 문제의 질문 297개를 분석해본 결과는 다음과 같았습니다.

- 예제를 통과하지 못함: **83개** (28%)
- 예제는 통과하지만 "0 45" 반례에서 틀림: **141개** (47%)
- 모든 질문을 커버할 수 있는 반례: **10개**

읽어보면 좋은 블로그

- 자주 하는 답변 모음집 ([링크](#))
 - 비슷한 패턴의 "틀리는 이유"가 모아서 정리된 곳
 - PPT 작성자 또한 백준 답변할 때 매우 많이 인용 중

스스로 해결이 안되었다면
질문의 시간입니다.

채점 결과를 정확히 알려주세요.

- 문제를 "틀렸다"는 표현은 "틀렸습니다"를 받았을 때만 사용해야 합니다.
- 채점 결과를 정확히 알려줘야 답변자도 이에 맞추어 코드를 분석할 수 있습니다.

제출한 코드를 그대로 주세요.

코드 전체가 있어야 합니다.

위 글을 인용하자면,

1. 오답을 받은 코드를
2. 한 글자의 수정도 없이
3. 전체를 그대로 가져와야 합니다.

코드를 캡처하지 마세요.

코드를 캡처해서 공유하면 답변자가 할 수 있는 것은 눈으로 읽는 것 뿐입니다.

자신이 작성한 코드도 눈으로만 읽으면 틀린 부분을 찾기 힘든데, 남이 작성한 코드는 오죽할까요?

답변자가 코드를 실행해볼 수 있도록 해주세요.

코드를 그대로 주는 방법

3 2884번 제출 맞힌 사람 슛코딩 재채점 결과 채점 현황 **내 제출** 난이도 기여 ↗ 질문 게시판

[NEW] 코딩 테스트 준비 온라인 강의 할인 중 / 코딩 테스트 및 인터뷰 온라인 강의

1

클릭!

2884

lcr7324

모든 언어

모든 결과

제출 번호	아이디	문제	결과	메모리	시간	언어
49834602	lcr7324	3 2884	맞았습니다!!	1112 KB	0 ms	C11 / 수정
49834597	lcr7324	3 2884	틀렸습니다			C11 수정
49833958	lcr7324	3 2884	맞았습니다!!	1112 KB	0 ms	C11 / 수정

클릭!

2

코드를 그대로 주는 방법

```
19
20
21
22
23     printf("%d %d", H, M);
24
25     return 0;
26 }
```

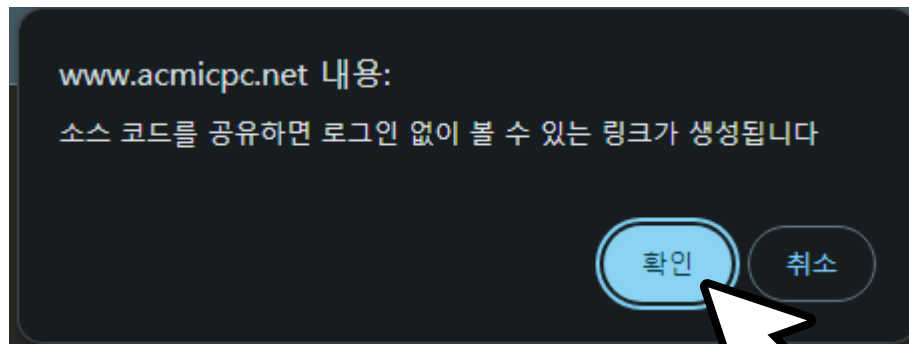
소스 코드 공개 공개 비공개 맞았을 때만 공개 [수정](#)

[복사](#) [다운로드](#) [공유](#)

제출 번호	아이디	문제	문제 제목	결과	메모리	시간	언어	코드 길이	제출 시간
49834597	lcr7324	2884	알람 시계	틀렸습니다			C11	479	11년 전

클릭!

코드를 그대로 주는 방법



1 클릭!

<http://boj.kr/19de46099eba4ff595d25c63ee1d6c60>



C11

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3
4 int main(){
```

2 클릭하여 링크 복사

⇒ 복사된 링크를 공유하세요!

코드를 그대로 주는 다른 방법

- ohmy.codes, pastebin.com 등의 코드 공유 사이트를 이용해도 좋습니다.
- 하지만 코드를 복사해서 채팅창에 그대로 붙여 넣는 등의 테러는 하지 말아주세요.

덧붙이면 좋은 것

- 본인이 접근한 방식에 대한 설명
- 본인이 시도해본 입력
 - 질문 게시판의 반례는 당연히 다 넣어놔야 합니다.
 - 추가로 테스트해본 케이스가 있다면 알려주세요.

어떤 입력에서 틀리는지 안다면

다음 두 가지 경우 중 하나입니다.

1. 풀이는 논리적으로 올바르나 구현을 잘못함
2. 풀이가 논리적으로 틀림

둘 중 어느 것인지 생각해 보세요.

2번의 경우로 의심된다면 질문해도 좋지만,

1번의 경우라면 스스로 디버깅하세요.

디버깅하는 법

디버거를 사용할 줄 알면 제일 좋습니다.

잘 모른다면, 코드 중간에 출력문을 추가하여 각 시점의 변수 상태를 일일이 확인하세요.

본인이 생각했던 풀이의 논리를 생각했을 때 각 시점에서 변수 및 식의 값이 어떠해야 하는지를 생각해보고, 실행 결과가 그것과 일치하는지 보세요.

코드가 예상치 않게 동작한다면

언어의 문법에 대한 이해가 필요할 수 있습니다.

해당하는 입력과 함께, "코드의 어느 부분에서 어떻게 동작할 것이라 예상했는데 실제로는 이렇게 동작했다"고 구체적으로 설명하세요.

해당 상황에 대한 설명을 받을 수 있습니다.