

Forum:[Hack Forums](#)

Thread: [Recursion error](#)

Sentience - July 16, 2022, 3:16 pm

This gives me a recursion error:

```
Code def f(n: int) -> bool:
```

```
    return f(n) >= f(n+1)
```

How can I write this so it runs?

Thanks,
Oliver

SIM-574702 - July 16, 2022, 3:36 pm

Function `f` is defined with `int` parameter so you can call it with `f(0)` or any other integer. If you really want to call it as `f(n)` you need to define `n`.

```
Code n=0
```

```
f(n)
```

I'm not sure what you're trying to do but this will always give you maximum recursion depth exceeded because you don't have an exit condition.

Athr - July 16, 2022, 3:38 pm

(Jul 16, 2022 03:16 PM)Sentience Wrote: [\[CITE\]](#) This gives me a recursion error:

```
Code def f(n: int) -> bool:
```

```
    return f(n) >= f(n+1)This says n is undefined:
```

```
Code f(n)
```

How can I write this so it runs?

Thanks,
Oliver

You have not defined `n` before calling the function.

Write the following line to define `n`.

```
Code #Sample n value
```

```
n = 5
```

```
print(f(n))
```

EDIT:

[SIM-574702](#) is right about the exit condition.

It's like a base case for an induction problem in math.

Sentience - July 18, 2022, 10:02 am

Code # We represent elements of $\mathbb{N}$? as functions `f : int -> bool`, where we only care about

the values of those functions for nonnegative inputs `n`, and the functions are

assumed to satisfy `f(n) >= f(n+1)` for all `n >= 0`.

This is an excerpt of what I am talking about

It's not supposed to call itself...

How can I write this:

Code def f(n: int) -> bool:
return f(n) >= f(n+1)
Without recursion?

Edit: Solved...

Code def g(f: typing.Callable[[int], bool], n: int) -> bool:
return f(n) >= f(n + 1)

Sentience - July 19, 2022, 10:26 am

Wait wtf how do call g()?

Turns out I can call g just by going

g

SIM-4763645 - July 25, 2022, 3:46 am

```

```
def f():
def g():
 f()
then call g() inside main thread
```
```

Sentience - July 25, 2022, 1:43 pm

(Jul 19, 2022 10:26 AM)Sentience Wrote: [\[CITE\]](#) Wait wtf how do call g()?

Turns out I can call g just by going

g

Edit: This thread is solved