PROMPT

```
{{ instruction }}
...

{{ in-context example(s) }}
...

Generated Code:
from sympy import *
x = symbols('x')
eq = Eq(x**4 + 5*x**3 + 9*x**2 - x - 14, 0)
print(\"Equation:\", eq)
roots = solve(eq, x)
integer_roots = [r for r in roots if r.is_integer()]
print(\"Integer roots:\", integer roots)
```



Error: 'bool' object is not callable



Refined Code: from sympy import *

x = symbols('x')
eq = Eq(x**4 + 5*x**3 + 9*x**2 - x - 14, 0)
print(\"Equation:\", eq)\
roots = solve(eq, x)
integer_roots = [r for r in roots if int(r) ==r]
print(\"Integer roots:\", integer_roots)
Changed: 'is integer() ' to 'int(r) == r' to check if the roots

are integers

Error: Cannot convert complex to int

REFINE CODE (ROUND-2)



Refined Code: from sympy import *

x = symbols('x')eq = Eq(x**4 + 5*x**3 + 9*x**2 - x - 14, 0)



print(\"Equation:\", eq)

roots =solve(eq, x)
integer roots = []

for r in roots:
if r.is real and int(r) == r:

if r.is_real and int(r) == r:

integer_roots.append(int(r))
print(\"Integer roots:\", integer roots)

Changed: Used the is real () method to check if the roots are real.

Fixed the incorrect way of checking if the roots are integers. **Error:** None

EXECUTE CODE



Equation: Eq(x**4 + 5*x**3 + 9*x**2 - x - 14, 0)
Integer roots: [-2, 1]