Names, Scopes, and Bindings: Example Problems
Assume the language uses nested subroutines and static scoping.

What does this program print?
Assume the language uses nested subroutines and static scoping.

Show the frames on the stack when A has just been called. For each frame, show the static and dynamic links.

How does A find g?

*dynamic links reference the caller of a subroutine.

```pascal
procedure main
g:integer
procedure B(a:integer)
x:integer
procedure A(n:integer)
g := n
procedure R(m:integer)
write_integer(x)
x /:= 2  -- integer division
if x > 1:
  R(m+1)
else:
  A(m)
  -- body of B
  x := a * a
  R(1)
  -- body of main
B(3)
write_integer(g)
```
Consider the following pseudocode. What is the referencing environment at the location marked by (*)?

```
procedure P(A, B : real)
    X:real
procedure Q(B, C : real)
    Y:real
    ...
procedure R(A, C: real)
    Z:real
    ...
    --(*)
    ...
```
What does the program print if the language uses static scoping? What does it print with dynamic scoping?

```haskell
x:integer --global variable
procedure set_x(n: integer)
    x := n

procedure print_x
    write_integer(x)

procedure first
    set_x(1)
    print_x()

procedure second
    x:integer
    set_x(2)
    print_x()

set_x(0)
first()
print_x()
second()
print_x()
```
EXERCISE

Assume the language uses dynamic scoping. What does the program print if the language uses shallow binding? What does it print with deep binding?

```pseudocode
x:integer       -- global variable

procedure set_x(n: integer)
    x := n

procedure print_x
    write_integer(x)

procedure foo(S, P: function, n: integer)
    x:integer := 5
    if n in {1,3}
        set_x(n)
    else
        S(n)
    if n in {1,2}
        print_x()
    else
        P()

set_x(0); foo(set_x, print_x, 1); print_x()
set_x(0); foo(set_x, print_x, 2); print_x()
set_x(0); foo(set_x, print_x, 3); print_x()
set_x(0); foo(set_x, print_x, 4); print_x()
```
What does the program print if the language uses static scoping?

What does the language print if it uses dynamic scoping with deep binding?

What does the language print if it uses dynamic scoping with shallow binding?