Names, Scopes, and Bindings
What does the function `test` print if the language uses static scoping? What does it print with dynamic scoping? (otherwise assume C++ syntax and semantics).

```cpp
int n = 100; // global

print_plus_n(int x) {
    cout << x + n;
}
increment_n() {
    n = n + 1;
}

test() {
    int n;
    n = 1;
    print_plus_n(25);

    n = 33;
    print_plus_n(n);

    increment_n();
    cout << n;

    print_plus_n(n);
}

With Static Scoping:

125 133 33 134

With Dynamic Scoping:

26 66 34 68
Parameter Passing Modes.
Show what is printed out by the code below using the parameter passing modes listed below. **print_array(a)** will print out all the elements in array a. (You can have it print ? for any uninitialized values.)

```pascal
array a[1..10] of integer;
integer n;

procedure p(b: integer);
begin
    print(b);
    n := n+2;
    print(b);
    b := b+5;
end;

a[1] := 10;
a[2] := 20;
a[3] := 30;
a[4] := 40;
n := 1;
p(a[n+1]);
new_line;
print_array(a);
```

**call by value:**
```
20 20
```

**call by reference:**
```
20 20
```

**call by name:**
```
20 40
```