

## Review: Pointers (creation to initialization)

1. What is the purpose of RAM in a computer system?
2. What does it mean when we say a pointer holds the “address” of a variable?
3. How is memory access different in languages like Java or Python compared to C?
4. What does the `%p` format specifier print in a C `printf` statement?
5. Why is pointer manipulation considered both powerful and dangerous in C?
6. What is a pointer in C, and how is it declared? Declare an integer variable `x` and a pointer `p` pointing to it.
7. What does the indirection (dereferencing) operator `*` do? Declare an integer variable `x`, a pointer `p` pointing to it, and a print statement to print the value of `p` through indirection.
8. What happens if you assign a literal integer (like 1) to a pointer?
9. What does the `sizeof` operator tell us about arrays? Declare an array `a` of 100 values, and store the size of `a` in `size` computed using `sizeof`.
10. Why must you use `&i` in a call to `scanf("%d", &i);`?