Copilot

The `%p` format specifier in C is used to print the **address or pointer** of a variable. When you use `%p` in a `printf()` statement, it displays the memory address to which the pointer refers. Here are some key points about `%p`:

1. **Pointer Address**: `%p` prints the hexadecimal representation of the memory address. It's useful for debugging and understanding memory layout.
2. **Void Pointer**: The corresponding argument for `%p` must be a `void *` value (i.e., a pointer). You can cast between a pointer and an integer since a pointer is essentially a 32-bit or 64-bit number representing a chunk of memory.
3. **Hexadecimal Format**: Since it deals with memory addresses, `%p` prints the address in hexadecimal format. For example, if the address is `0x55`, it corresponds to decimal `85`.

Here's an example:

```c
#include <stdio.h>

int main() {
    int i = 85;
    printf("Address of i: %p\n", &i); // Prints the address of 'i'
    return 0;
}
```

The output will be something like:

```
Address of i: 0x7ffdbb8f6a4c
```

Remember that the actual address will vary based on your system architecture and memory layout. If you're working with pointers, `%p` is a handy specifier to display memory addresses. 😊

For more information, you can refer to the GeeksforGeeks article on format specifiers in C programming. It covers various other format specifiers as well! 🚀