**04\_variables**

**What's the difference between:**

1. printf("You are %d years old.", age);

2. printf("You are %i years old.", age);

The %d and %i format specifiers are interchangeable when dealing with integers in printf. However, there is a difference between %d and %i when used with scanf for input. %d expects the input to be in decimal format, while %i allows for input in decimal, octal, or hexadecimal format, based on prefixes like 0x for hexadecimal and 0 for octal.

**30\_nested\_loops**

**If I try to use scanf to read a character after having read a numeric value, it fails.**

The issue you're encountering is due to the behavior of scanf when reading characters after reading numeric values.

When you use scanf to read numeric values (%d for integers in this case), it reads the numeric value but leaves the newline character ('\n') in the input buffer. When you then use scanf to read a character (%c), it reads the next character in the input buffer, which is often the newline character left over from the previous input operation.

To fix this issue, you need to consume the newline character from the input buffer before reading the character. You can do this by adding a space before %c in the format string, like this:

scanf(" %c", &symbol); // Note the space before %c

The space before %c in the format string tells scanf to skip any whitespace characters (including newline characters) before reading the character. This ensures that scanf reads the correct character input from the user.