```
.globl starts_with            # Make the `starts_with` routine
globally accessible.

.globl prefix_match           # Global label for successful prefix
match.

.globl prefix_diff            # Global label for prefix mismatch.

.globl starts_with_char1      # Special case to compare a single
character.


# Main routine to check if a string (pointed to by t0) starts with a
given prefix (pointed to by t1).

starts_with:

    lb t2, 0(t1)              # Load the first byte of the prefix
string into t2.

    beqz t2, prefix_match     # If the prefix is empty (null terminator
reached), it's a match.

    lb t3, 0(t0)              # Load the first byte of the input string
into t3.

    beqz t3, prefix_diff      # If the input string ends before the
prefix, it's a mismatch.

    bne t2, t3, prefix_diff   # If the characters do not match, it's a
mismatch.

    addi t0, t0, 1            # Move to the next character in the input
string.

    addi t1, t1, 1            # Move to the next character in the
prefix string.

    j starts_with             # Repeat for the next character.


# Special case: Check if a single character (pointed to by t1) matches
the start of a string (pointed to by t0).

starts_with_char1:

    lb t2, 0(t1)              # Load the single character into t2.

    lb t3, 0(t0)              # Load the first character of the input
string into t3.

    beq t2, t3, prefix_match # If they match, jump to `prefix_match`.

    j prefix_diff             # Otherwise, it's a mismatch.


# Label for a successful match.
```

```
prefix_match:
    li a0, 0                # Set return value to 0 (indicates a
match).
    ret                     # Return to the caller.


# Label for a mismatch.
prefix_diff:
    li a0, 1                # Set return value to 1 (indicates a
mismatch).
    ret                     # Return to the caller.
```