14.- ft_tolower.-

Function based on the definition given in the BSD man pages for “tolower(3)”.
The library associated is <ctype.h> (standard C library).

Synopsis:

```
int tolower(int c);
```

Purpose:

Converts an uppercase letter to its lowercase equivalent.

Parameters:

- `c`: The character to be converted.

Return Value

- Returns the lowercase equivalent of `c` if it's an uppercase letter.
- Returns `c` unchanged if it's not an uppercase letter.

Description

- Checks if `c` is an uppercase letter (between 'A' and 'Z').
- If it's uppercase, adds 32 to its ASCII value to get the lowercase equivalent.
- Returns the lowercase character or the original character.

Code

```
#include "libft.h"

int ft_tolower(int c)
{
    if (c >= 'A' && c <= 'Z')
    {
        c = c + 32;
    }
    return (c);
}
```

Code Explanation

- **Checks for uppercase:**
  - Sees if `c` falls within the ASCII range for uppercase letters (65 to 90).
- **Converts to lowercase:**
  - If `c` is uppercase, adds 32 to reach the ASCII range for lowercase letters (97 to 122).
- **Returns character:**
  - Returns either the lowercase character or the original `c` if it wasn't uppercase.

Main Function (Optional)

```
int main(void)
{
    char a;

    a = 'Y';
    a = ft_tolower(a);
    printf("%c\n", a);  // Output: y
    return (0);
}
```

The above main function it is used to check the ft_tolower function – it converts ‘Y’ to ‘y’.

Key Points:

- **ASCII Values:** Characters have numerical codes called ASCII values.
- **Uppercase Letters:** ASCII values 65 to 90 represent uppercase letters.
- **Lowercase Letters:** ASCII values 97 to 122 represent lowercase letters.
- **Adding 32:** Moving from uppercase to lowercase involves adding 32 to the ASCII value.