13.- ft_toupper.-

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

Synopsis:

```
int toupper(int c);
```

Purpose:

Converts a lowercase letter to its uppercase equivalent.

Parameters:

`c`: The character to be converted.

Return Value:

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

Description

- Checks if `c` is a lowercase letter (between 'a' and 'z').
- If it's lowercase, subtracts 32 from its ASCII value to get the uppercase equivalent.
- Returns the uppercase character or the original character.

  The toupper() function converts a lower-case letter to the corresponding upper-case letter. The argument must be representable as an unsigned

Code

```
#include "libft.h"

int ft_toupper(int c)
{
    if (c >= 'a' && c <= 'z')
    {
        c = c - 32;
    }
    return (c);
}
```

Code Explanation

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

Main Function (Optional)

```
int main(void)
```

```
{
    char a;

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

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

Key Points:

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