4.- `ft_isascii.-`

Function based on the definition given in the BSD man pages for “isascii”.
The library associated is <ctype.h>.

Sinopsis: int isascii (int c);

Purpose: Checks for an ASCII character.

Parameters:

- `c`: The character to be checked.

Return value:

- 1 if **`c`** is an ASCII character, and 0 otherwise.

Description:

The **`ft_isascii`** function checks whether the given character is an ASCII character. An ASCII character is any character from the set **`[0, 127]`**. The function uses a simple comparison to check if the character's ASCII value falls within this range. If the character falls within this range, it returns 1; otherwise, it returns 0.

Code:

```
#include "libft.h"

int   ft_isascii(int c)
{
      return (c >= 0 && c <= 127);
}
/*
int   main(void)
{
      char  c;

      printf("Input an ascii character to obtain 1: ");
      scanf("%c", &c);
      printf("%d\n", ft_isascii(c));
      return (0);
}
*/
```

Code explanation:

1. **Include header file:** The **`#include "libft.h"`** statement includes the header file **`libft.h`**, which defines the required libraries for our function.
2. **Define function:** The `int` **`ft_isascii(int c)`** statement defines the `ft_isascii` function. The function takes one argument, `c`, which is the character to be checked.
3. **Return value:** The **`return (c >= 0 && c <= 127);`** statement returns 1 if `c` is an ASCII character, and 0 otherwise. This statement checks if the ASCII value of `c` is between 0 (ASCII value of NUL character) and 127. If `c` is within this range, the statement returns 1; otherwise, it returns 0.

4. Under comments we develop a main function to show how it works:

4.1 Main function: The **`int main(void)`** statement defines the main function, which is the entry point of the program. The main function prompts the user to enter an ASCII character, reads the input character, and prints a message indicating whether the character is ASCII.

4.2 Input and check ascii: The **`scanf("%c", &c);`** statement reads a character from the user's input and stores it in the variable `c`. The **`(ft_isascii(c))`** statement checks whether `c` is an ASCII character. If `c` is an ASCII character, the statement prints a message indicating that `c` is an ASCII character (that is 1); otherwise, it prints a message indicating that `c` is not an ASCII character (that is 0),

4.3 Return value: The **`return (0);`** statement exits the program with a status code of 0, indicating that the program executed successfully.