12.- ft_strlcat.-

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

Synopsis:

```
size_t strlcat(char * restrict dst, const char * restrict src, size_t dstsize);
```

We shall use:

```
size_t ft_strlcat(char *dst, const char *src, size_t dstsize);
```

Purpose:

Appends a string to another, preventing buffer overflows and ensuring null-termination.

Parameters:

- `dst`: Pointer to the destination string buffer.
- `src`: Pointer to the source string to be appended.
- `dstsize`: Size of the destination buffer, including space for the null terminator.

Return Value

- Returns the total length of the string that would have been created if there had been enough space in `dst`.

Description

- Appends up to `dstsize - strlen(dst) - 1` bytes from `src` to `dst`.
- Always null-terminates the destination string, even if truncation occurs.
- Returns the total length of the string that would have been created if there had been no truncation.

Code

```
#include "libft.h"

size_t ft_strlcat(char *dst, const char *src, size_t dstsize)
{
    size_t i;
    size_t l;

    l = ft_strlen(dst) + ft_strlen(src);
    if (dstsize <= ft_strlen(dst))
        return (ft_strlen(src) + dstsize);
    while (*dst)
        dst++;
    i = 0;
    while ((i < dstsize - (l - ft_strlen(src)) - 1) && src[i])
    {
        dst[i] = src[i];
        i++;
    }
    dst[i] = '\0';
    return (l);
```

Code Explanation:

1. **Calculates total string length:**
    - Finds the combined length of `dst` and `src` (`l`).
2. **Checks for insufficient space:**
    - If `dstsize` is less than or equal to the current length of `dst`, returns the total length that would have been needed (`l`).
3. **Finds end of `dst`:**
    - Iterates through `dst` until the null terminator is found.
4. **Appends characters (if space allows):**
    - Copies characters from `src` to `dst` until:
        - The end of `src` is reached.
        - The available space in `dst` is filled (`dstsize - strlen(dst) - 1`).
5. **Null-terminates `dst`:**
    - Ensures `dst` ends with a null terminator.
6. **Returns total string length:**
    - Returns the total length of the string that would have been created (`l`).

Main Function (Optional)

```
int main(void)
{
    char src[20] = " friend";
    char dst[20] = "Hello";

    printf("%zu\nsrc: %s\ndst: %s\n", ft_strlcat(dst, src, 13), src, dst);
    return (0);
}
```