36.- The library libft.h.-

Code:

#ifndef LIBFT_H

define LIBFT_H

include <ctype.h>

include <string.h>

include <stdlib.h>

include <unistd.h>

include <stdio.h>

//Libc functions.

int   ft_isalpha(int c);//1.

int   ft_isdigit(int c);//2.

int   ft_isalnum(int c);//3.

int   ft_isascii(int c);//4.

int   ft_isprint(int c);//5.

size_t ft_strlen(const char *s);//6.

void   *ft_memset(void *b, int c, size_t len);//7.

void   ft_bzero(void *s, size_t n);//8.

void   *ft_memcpy(void *dst, const void *src, size_t n);//9.

void   *ft_memmove(void *dst, const void *src, size_t len);//10.

size_t ft_strlcpy(char *dst, const char *src, size_t dstsize);//11.

size_t ft_strlcat(char *dst, const char *src, size_t dstsize);//12.

int   ft_toupper(int c);//13.

int   ft_tolower(int c);//14.

char   *ft_strchr(const char *s, int c);//15.

char   *ft_strrchr(const char *s, int c);//16.

int   ft_strncmp(const char *s1, const char *s2, size_t n);//17.

void   *ft_memchr(const void *s, int c, size_t n);//18.

int   ft_memcmp(const void *s1, const void *s2, size_t n);//19.

char   *ft_strnstr(const char *haystack, const char *needle, size_t len);//20.

int   ft_atoi(const char *str);//21.

void   *ft_calloc(size_t count, size_t size);//22.

char   *ft_strdup(const char *s1);//23.

//Additional functions.

char   *ft_substr(char const *s, unsigned int start, size_t len);//24.

char   *ft_strjoin(char const *s1, char const *s2);//25.

char   *ft_strtrim(char const *s1, char const *set);//26.

char   **ft_split(char const *s, char c);//27.

char   *ft_itoa(int n);//28.

char   *ft_strmapi(char const *s, char (*f)(unsigned int, char));//29.

void   ft_striteri(char *s, void (*f)(unsigned int, char*));//30.

void   ft_putchar_fd(char c, int fd);//31.

void   ft_putstr_fd(char *s, int fd);//32.

void   ft_putendl_fd(char *s, int fd);//33.

void   ft_putnbr_fd(int n, int fd);//34.

#endif

/*NOTE: Following functions use next libraries:

<ctype.h>isalpha,isdigit,isalnum,isascii,isprint,toupper,tolower.

<string.h>strlen,memset,bzero,memcpy,memmove,strlcpy,strlcat,strchr,

        strrchr,strncmp,memchr,memcmp,strnstr,strdup.

<stdlib.h>atoi,calloc.

*/

This code defines a header file named `libft.h`, which contains function prototypes for a variety of useful functions. Here's a breakdown of the key elements:

1. `#ifndef LIBFT_H` **and** `#define LIBFT_H`**:**

- These lines guard against including the header file multiple times in a single compilation unit. If `LIBFT_H` is not already defined, it gets defined here, preventing further inclusions. This avoids conflicting definitions and potential errors.

2. `#include` **directives:**

- These lines include standard C libraries like `<ctype.h>`, `<string.h>`, and `<stdlib.h>`, providing access to existing functions like `isalpha`, `strlen`, `atoi`, etc.

3. Function prototypes:

- The entire file defines prototypes for various functions using the syntax `returnType functionName(parameters);`. These prototypes tell the compiler about the functions' existence, their return type, and their parameters, even though the actual function definitions might be elsewhere.

4. `//Libc functions.:` **and** `//Additional functions.:` **comments:**

- These comments categorize the functions into standard library functions (Libc) and custom functions added by the library.

5. Individual function explanations:

- Each function has a name, parameters, and a return type. You can find detailed explanations of each function's purpose and behavior in the previous explanations I provided.

6. `#endif:`

- This line marks the end of the conditional block started by `#ifndef`. It signals that the `LIBFT_H` definition is no longer active, preventing unexpected effects if included multiple times.

7. Comment about used libraries:

- This comment notes which standard libraries are used by the functions listed below. It helps clarify dependencies and potential linking requirements.

I hope this comprehensive explanation clarifies the content of the `libft.h` header file and the use of `#ifndef`, `#define`, `#include`, and other directives.