Assignment 61: What are the various input/output functions available in C language ?

In C programming, there are several input/output functions available for reading and writing data. Here are some of the commonly used ones along with brief explanations:

1. **printf()**:
    - `printf()` is used to print formatted output to the standard output (usually the console).
    - It takes a format string as input, which specifies how the output should be formatted, and additional arguments corresponding to the format specifiers in the format string.
    - Example: `printf("The value of x is %d\n", x);`

2. **scanf()**:
    - `scanf()` is used to read formatted input from the standard input (usually the keyboard).
    - It takes a format string as input, which specifies the format of the input data, and additional arguments corresponding to the variables where the input data should be stored.
    - Example: `scanf("%d", &x);`

3. **getchar()**
    - `getchar()` is used to read a single character from the standard input.
    - It reads a character from the input buffer and returns it as an integer value.
    - Example: `char ch = getchar();`

4. **putchar()**:
    - `putchar()` is used to output a single character to the standard output.
    - It takes a character as input and writes it to the output buffer.
    - Example: `putchar('A');`

5. **gets()** and **fgets()**:
    - `gets()` and `fgets()` are used to read a string from the standard input.
    - `gets()` is deprecated due to its vulnerability to buffer overflow attacks and should be avoided.
    - `fgets()` is safer to use as it allows you to specify the maximum number of characters to read and prevents buffer overflow.
    - Example: `char str[100]; gets(str);` or `char str[100]; fgets(str, 100, stdin);`

6. **puts()**:
    - `puts()` is used to output a string to the standard output followed by a newline character.
    - It takes a null-terminated string as input and writes it to the output buffer.
    - Example: `puts("Hello, world!");`

7. **scanf()** and **printf()** Family:
    - There are various functions in the `scanf()` and `printf()` families that handle different data types and formatting options, such as `%d` for integers, `%f` for floats, `%s` for strings, etc.
    - These functions allow for formatted input and output of various data types, making them versatile for different programming tasks.

These are just some of the input/output functions available in C language. Each function serves a specific purpose and can be used depending on the requirements of the program.