fclose *Close File* `<stdio.h>`

```
int fclose(FILE *stream);
```

Closes the stream pointed to by `stream`. Flushes any unwritten output remaining in the stream's buffer. Deallocates the buffer if it was allocated automatically.

Returns Zero if successful, `EOF` if an error was detected. *22.2*

---

fdim *Positive Difference (C99)* `<math.h>`

```
double fdim(double x, double y);
```
fdimf `float fdimf(float x, float y);`
fdiml `long double fdiml(long double x, long double y);`

Returns Positive difference of `x` and `y`:

$$\begin{cases} x - y & \text{if } x > y \\ +0 & \text{if } x \le y \end{cases}$$

A range error may occur. *23.4*

---

feclearexcept *Clear Floating-Point Exceptions (C99)* `<fenv.h>`

```
int feclearexcept(int excepts);
```

Attempts to clear the floating-point exceptions represented by `excepts`.

Returns Zero if `excepts` is zero or if all specified exceptions were successfully cleared; otherwise, returns a nonzero value. *27.6*

---

fegetenv *Get Floating-Point Environment (C99)* `<fenv.h>`

```
int fegetenv(fenv_t *envp);
```

Attempts to store the current floating-point environment in the object pointed to by `envp`.

Returns Zero if the environment was successfully stored; otherwise, returns a nonzero value. *27.6*

---

fegetexceptflag *Get Floating-Point Exception Flags (C99)* `<fenv.h>`

```
int fegetexceptflag(fexcept_t *flagp, int excepts);
```

Attempts to retrieve the states of the floating-point status flags represented by `excepts` and store them in the object pointed to by `flagp`.

Returns Zero if the states of the status flags were successfully stored; otherwise, returns a nonzero value. *27.6*

---

fegetround *Get Floating-Point Rounding Direction (C99)* `<fenv.h>`

```
int fegetround(void);
```

Returns Value of the rounding-direction macro that represents the current rounding direction. Returns a negative value if the current rounding direction can't be determined or doesn't match any rounding-direction macro. *27.6*