setlocale — *Set Locale* — <locale.h>

```
char *setlocale(int category, const char *locale);
```

Sets a portion of the program's locale. `category` indicates which portion is affected. `locale` points to a string representing the new locale.

Returns If `locale` is a null pointer, returns a pointer to the string associated with `category` for the current locale. Otherwise, returns a pointer to the string associated with `category` for the new locale. Returns a null pointer if the operation fails.

25.1

setvbuf — *Set Buffer* — <stdio.h>

```
int setvbuf(FILE * restrict stream,
            char * restrict buf,
            int mode, size_t size);
```

Changes the buffering of the stream pointed to by `stream`. The value of `mode` can be either `_IOFBF` (full buffering), `_IOLBF` (line buffering), or `_IONBF` (no buffering). If `buf` is a null pointer, a buffer is automatically allocated if needed. Otherwise, `buf` points to a memory block that can be used as the buffer; `size` is the number of bytes in the block. *Note:* `setvbuf` must be called after the stream is opened but before any other operations are performed on it.

Returns Zero if the operation is successful. Returns a nonzero value if `mode` is invalid or the request can't be honored.

22.2

signal — *Install Signal Handler* — <signal.h>

```
void (*signal(int sig, void (*func)(int)))(int);
```

Installs the function pointed to by `func` as the handler for the signal whose number is `sig`. Passing `SIG_DFL` as the second argument causes default handling for the signal; passing `SIG_IGN` causes the signal to be ignored.

Returns A pointer to the previous handler for this signal; returns `SIG_ERR` and stores a positive value in `errno` if the handler can't be installed.

24.3

signbit — *Sign Bit (C99)* — <math.h>

```
int signbit(real-floating x);
```

macro

Returns A nonzero value if the sign of `x` is negative and zero otherwise. The value of `x` may be any number, including infinity and NaN.

23.4

sin — *Sine* — <math.h>

```
double sin(double x);
float sinf(float x);
long double sinl(long double x);
```

Returns Sine of `x` (measured in radians).

23.3

sinh — *Hyperbolic Sine* — <math.h>

```
double sinh(double x);
```