strxfrm *Transform String* <string.h>

```
size_t strxfrm(char * restrict s1,
               const char * restrict s2, size_t n);
```

Transforms the string pointed to by `s2`, placing the first `n` characters of the result—including the null character—in the array pointed to by `s1`. Calling `strcmp` with two transformed strings should produce the same outcome (negative, zero, or positive) as calling `strcoll` with the original strings. If `n` is zero, `s1` is allowed to be a null pointer.

Returns Length of the transformed string. If this value is `n` or more, the contents of the array pointed to by `s1` are indeterminate. *23.6*

swprintf *Wide-Character Formatted String Write (C99)* <wchar.h>

```
int swprintf(wchar_t * restrict s, size_t n,
             const wchar_t * restrict format, ...);
```

Equivalent to `fwprintf`, but stores wide characters in the array pointed to by `s` instead of writing them to a stream. The string pointed to by `format` specifies how subsequent arguments will be displayed. No more than `n` wide characters will be written to the array, including a terminating null wide character.

Returns Number of wide characters stored in the array, not including the null wide character. Returns a negative value if an encoding error occurs or the number of wide characters to be written is `n` or more. *25.5*

swscanf *Wide-Character Formatted String Read (C99)* <wchar.h>

```
int swscanf(const wchar_t * restrict s,
            const wchar_t * restrict format, ...);
```

Wide-character version of `sscanf`. *25.5*

system *Perform Operating-System Command* <stdlib.h>

```
int system(const char *string);
```

Passes the string pointed to by `string` to the operating system's command processor (shell) to be executed. Program termination may occur as a result of executing the command.

Returns If `string` is a null pointer, returns a nonzero value if a command processor is available. If `string` isn't a null pointer, `system` returns an implementation-defined value (if it returns at all). *26.2*

tan *Tangent* <math.h>

```
double tan(double x);
float tanf(float x);
long double tanl(long double x);
```

Returns Tangent of `x` (measured in radians). *23.3*