sinhf `float sinhf(float x);`
sinhl `long double sinhl(long double x);`

Returns Hyperbolic sine of x. A range error occurs if the magnitude of x is too large. *23.3*

---

snprintf *Bounded Formatted String Write (C99)* <stdio.h>

```
int snprintf(char * restrict s, size_t n,
             const char * restrict format, ...);
```

Equivalent to `fprintf`, but stores characters in the array pointed to by `s` instead of writing them to a stream. No more than `n` – 1 characters will be written to the array. The string pointed to by `format` specifies how subsequent arguments will be displayed. Stores a null character in the array at the end of output.

Returns Number of characters that would have been stored in the array (not including the null character) had there been no length restriction. Returns a negative value if an encoding error occurs. *22.8*

---

sprintf *Formatted String Write* <stdio.h>

```
int sprintf(char * restrict s,
            const char * restrict format, ...);
```

Equivalent to `fprintf`, but stores 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. Stores a null character in the array at the end of output.

Returns Number of characters stored in the array, not including the null character. In C99, returns a negative value if an encoding error occurs. *22.8*

---

sqrt *Square Root* <math.h>

```
double sqrt(double x);
float sqrtf(float x);
long double sqrtl(long double x);
```

Returns Nonnegative square root of x. A domain error occurs if x is negative. *23.3*

---

srand *Seed Pseudo-Random Number Generator* <stdlib.h>

```
void srand(unsigned int seed);
```

Uses `seed` to initialize the sequence of pseudo-random numbers produced by calling `rand`. *26.2*

---

sscanf *Formatted String Read* <stdio.h>

```
int sscanf(const char * restrict s,
           const char * restrict format, ...);
```

Equivalent to `fscanf`, but reads characters from the string pointed to by `s` instead of reading them from a stream. The string pointed to by `format` specifies the format of the items to be read. The arguments that follow `format` point to objects in which the items are to be stored.