Returns	Number of input items successfully read and stored. Returns EOF if an input failure occurs before any items could be read.	*22.8*

---

strcat   *String Concatenation*   `<string.h>`

```
char *strcat(char * restrict s1,
             const char * restrict s2);
```

Appends characters from the string pointed to by s2 to the string pointed to by s1.

Returns   s1 (a pointer to the concatenated string).   *13.5, 23.6*

---

strchr   *Search String for Character*   `<string.h>`

```
char *strchr(const char *s, int c);
```

Returns   A pointer to the first occurrence of the character c in the string pointed to by s. Returns a null pointer if c isn't found.   *23.6*

---

strcmp   *String Comparison*   `<string.h>`

```
int strcmp(const char *s1, const char *s2);
```

Returns   A negative, zero, or positive integer, depending on whether the string pointed to by s1 is less than, equal to, or greater than the string pointed to by s2.   *13.5, 23.6*

---

strcoll   *String Comparison Using Locale-Specific Collating*   `<string.h>`
                    Sequence

```
int strcoll(const char *s1, const char *s2);
```

Returns   A negative, zero, or positive integer, depending on whether the string pointed to by s1 is less than, equal to, or greater than the string pointed to by s2. The comparison is performed according to the rules of the current locale's LC_COLLATE category.   *23.6*

---

strcpy   *String Copy*   `<string.h>`

```
char *strcpy(char * restrict s1,
             const char * restrict s2);
```

Copies the string pointed to by s2 into the array pointed to by s1.

Returns   s1 (a pointer to the destination).   *13.5, 23.6*

---

strcspn   *Search String for Initial Span of Characters Not in Set*   `<string.h>`

```
size_t strcspn(const char *s1, const char *s2);
```

Returns   Length of the longest initial segment of the string pointed to by s1 that doesn't contain any character in the string pointed to by s2.   *23.6*

---

strerror   *Convert Error Number to String*   `<string.h>`

```
char *strerror(int errnum);
```

Returns   A pointer to a string containing an error message corresponding to the value of errnum.   *24.2*