strpbrk *Search String for One of a Set of Characters* `<string.h>`

```
char *strpbrk(const char *s1, const char *s2);
```

Returns A pointer to the leftmost character in the string pointed to by `s1` that matches any character in the string pointed to by `s2`. Returns a null pointer if no match is found. *23.6*

---

strrchr *Search String in Reverse for Character* `<string.h>`

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

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

---

strspn *Search String for Initial Span of Characters in Set* `<string.h>`

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

Returns Length of the longest initial segment in the string pointed to by `s1` that consists entirely of characters in the string pointed to by `s2`. *23.6*

---

strstr *Search String for Substring* `<string.h>`

```
char *strstr(const char *s1, const char *s2);
```

Returns A pointer to the first occurrence in the string pointed to by `s1` of the sequence of characters in the string pointed to by `s2`. Returns a null pointer if no match is found. *23.6*

---

strtod *Convert String to Double* `<stdlib.h>`

```
double strtod(const char * restrict nptr,
              char ** restrict endptr);
```

Skips white-space characters in the string pointed to by `nptr`, then converts subsequent characters into a `double` value. If `endptr` isn't a null pointer, `strtod` modifies the object pointed to by `endptr` so that it points to the first leftover character. If no `double` value is found, or if it has the wrong form, `strtod` stores `nptr` in the object pointed to by `endptr`. If the number is too large or small to represent, it stores `ERANGE` in `errno`. *C99 changes:* The string pointed to by `nptr` may contain a hexadecimal floating-point number, infinity, or NaN. Whether `ERANGE` is stored in `errno` when the number is too small to represent is implementation-defined.

Returns The converted number. Returns zero if no conversion could be performed. If the number is too large to represent, returns plus or minus `HUGE_VAL`, depending on the number's sign. Returns zero if the number is too small to represent. *C99 change:* If the number is too small to represent, `strtod` returns a value whose magnitude is no greater than the smallest normalized positive `double`. *26.2*

---

strtof *Convert String to Float (C99)* `<stdlib.h>`

```
float strtof(const char * restrict nptr,
             char ** restrict endptr);
```