vsscanf *Formatted String Read Using Variable Argument List (C99)* <stdio.h>

```
int vsscanf(const char * restrict s,
            const char * restrict format,
            va_list arg);
```

Equivalent to `sscanf` with the variable argument list replaced by `arg`.

Returns Number of input items successfully read and stored. Returns `EOF` if an input failure occurs before any items can be read. *26.1*

---

vswprintf *Wide-Character Formatted String Write Using Variable Argument List (C99)* <wchar.h>

```
int vswprintf(wchar_t * restrict s, size_t n,
              const wchar_t * restrict format,
              va_list arg);
```

Equivalent to `swprintf` with the variable argument list replaced by `arg`.

Returns Number of wide characters stored in the array pointed to by `s`, 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*

---

vswscanf *Wide-Character Formatted String Read Using Variable Argument List (C99)* <wchar.h>

```
int vswscanf(const wchar_t * restrict s,
             const wchar_t * restrict format,
             va_list arg);
```

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

---

vwprintf *Wide-Character Formatted Write Using Variable Argument List (C99)* <wchar.h>

```
int vwprintf(const wchar_t * restrict format,
             va_list arg);
```

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

---

vwscanf *Wide-Character Formatted Read Using Variable Argument List (C99)* <wchar.h>

```
int vwscanf(const wchar_t * restrict format,
            va_list arg);
```

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

---

wcrtomb *Convert Wide Character to Multibyte Character – Restartable (C99)* <wchar.h>

```
size_t wcrtomb(char * restrict s, wchar_t wc,
               mbstate_t * restrict ps);
```

If `s` is a null pointer, a call of `wcrtomb` is equivalent to

```
wcrtomb(buf, L'\0', ps)
```