

```

// This requires Igor 9.00
WAVE/B/U w1 = StringToUnsignedByteWave(theStr)
Print w1

// This works in older versions of Igor
Variable len = strlen(theStr)
Make/B/U/FREE/N=(len) w1
w1 = char2num(theStr[p])
Print w1
End

```

**See Also**

**WaveDataToString**, **MoveWave**, **Free Waves** on page IV-91, **Working With Binary String Data** on page IV-175

**strlen****strlen(*str*)**

The strlen function returns the number of bytes in the string expression *str*.

strlen returns NaN if the *str* is NULL. A local string variable or a string field in a structure that has never been set is NULL. NULL is not the same as zero length. Use **numtype** to test if the result from strlen is NaN.

**Examples**

```

String zeroLength = ""
String neverSet
Print strlen(zeroLength), strlen(neverSet)

// Test if a string is null
Variable len = strlen(neverSet) // NaN if neverSet is null
if (numtype(len) == 2)          // strlen returned NaN?
    Print "neverSet is null"
endif

```

**See Also**

**Characters Versus Bytes** on page III-483, **Character-by-Character Operations** on page IV-173

**strsearch****strsearch(*str*, *findThisStr*, *start* [, *options*])**

The strsearch function returns the byte position of the string expression *findThisStr* in the string expression *str*.

**Details**

strsearch performs a case-sensitive search.

strsearch returns -1 if *findThisStr* does not occur in *str*.

The search starts from the byte position in *str* specified by *start*; 0 references the start of *str*.

strsearch clips *start* to one less than the length of *str* in bytes, so it is useful to use Inf for *start* when searching backwards to ensure that the search is from the end of *str*.

*options* is an optional bitwise parameter specifying the search options:

- 1: Search backwards from *start*.
- 2: Ignore case.
- 3: Search backwards and ignore case.

**Examples**

```

String str="This is a test isn't it?"
Print strsearch(str,"test",0)           // prints 10
Print strsearch(str,"TEST",0)           // prints -1
Print strsearch(UpperStr(str),"TEST",0) // prints 10
Print strsearch(str,"TEST",0,2)         // prints 10
Print strsearch(str,"is",0)             // prints 2

```