Harbour Guide

GT_ASCPOS()

Return the ascii value of a specified character in a string

Syntax

GT_Ascpos(<cStr>, <nPos>) --> nAscVal

Arguments

<cStr> - The string <nPos> - The position in <cStr>

Returns

<nAscVal> - The ascii value of substr(<cStr>, <nPos>, 1)

Description

Return the ascii value of a specified character in a string Equivalent (but much faster) to asc(substr(cStr, nPos, 1)

NOTE: invalid parameters will return -1 nPos > len(cStr) will return -2

This last behaviour is different to the Funcky function of the same name. I changed the behaviour because some of the strings I process contain embedded NULs.

Examples

```
? gt_ascpos("the cat sat on the mat", 3) // prints e
```

Status

Ready

Files

Library is libgt

GT_ASCIISUM()

Sum the ascii values in a string.

Syntax

GT_AsciiSum(<cStr>) --> nSum

Arguments

<cStr> - The string to sum

Returns

<nSum> - The sum of all ascii values in <cStr>.

Description

Sum the ascii value of every character in the passed string and return the result.

Status

Ready

Files

Library is libgt

GT_ATDIFF()

Return the position where two strings begin to differ

Syntax

GT_AtDiff(<cStr1>, <cStr2>) --> nPos

Arguments

<cStr1> - A character string to compare <cStr2> - The string to compare with

Returns

<nPos> - The position in <cStr2> where <cStr1> begins to differ

Description

Return the position in <cStr2> where <cStr1> begins to differ. If the strings differ in the first character GT_AtDiff() will return 1. If the two strings are identical (or identical upto the last character in <cStr2>) the function will return 0.

NOTE: invalid parameters will return -1

Examples

```
? gt_atDiff("the cat", "the rat")          // prints 5
? gt_atDiff("the cat", "the ")             // prints 0
```

Status

Ready

Files

Library is libgt

GT_CHAREVEN()

Return a string of all the characters in even positions

Syntax

GT_CharEven(<cStr>) --> cRet

Arguments

<cStr> - A character string to extract chars from

Returns

<cRet> - A string of all the chars in even positions

Description

Return a string consisting of all the characters in even positions in <cStr1>.

NOTE: invalid parameters will return ""

Examples

```
? gt_CharEven("abcdefghijklm")            // prints "bdfhjl"
```

Status

Ready

Files

Library is libgt

GT_CHARMIX()

Amalgamate two strings to form the return value

Syntax

GT_CharMix(<cStr1>, <cStr2>) --> cRet

Arguments

<cStr1> - A character string to mix <cStr2> - A character string to mix with

Returns

<cRet> - A string consisting of all the characters in <cStr1> mixed with all the characters in <cStr2>

Description

Return a string consisting of all the characters in <cStr1> mixed with the characters from <cStr2>.

NOTE: invalid parameters will return ""

Examples

```
? gt_CharMix("abc", "123")              // prints "a1b2c3"
? gt_CharMix("abcde", "123")            // prints "a1b2c3de"
? gt_CharMix("abc", "12345")            // prints "a1b2c345"
```

Status

Ready

Files

Library is libgt

GT_CHARODD()

Return a string of all the characters in odd positions

Syntax

GT_CharOdd(<cStr>) --> cRet

Arguments

<cStr> - A character string to extract chars from

Returns

<cRet> - A string of all the chars in odd positions

Description

Return a string consisting of all the characters in odd positions in <cStr1>.

NOTE: invalid parameters will return ""

Examples

```
? gt_CharOdd("abcdefghijklm")            // prints "acegikm"
```

Status

Ready

Files

Library is libgt

GT_CHRCOUNT()

Count the number of times a character appears in a string

Syntax

GT_ChrCount(<cChr>, <cStr>) --> nFreq

Arguments

<cChr> - The character to find the frequence of <cStr> - The string in which to find the character

Returns

Description

GT_ChrCount() counts how many times a specified character appears in a string.

NOTE: invalid parameters will return -1

Examples

```
? GT_ChrCount("t", "the cat sat on the mat")      // prints 4
```

Status

Ready

Files

Library is libgt

GT_CHRFIRST()

Find which character occurs first in a string

Syntax

GT_ChrFirst(<cChars>, <cStr>) --> nAsc

Arguments

<cChars> - The set of characters to find <cStr> - The input string

Returns

<nAsc> - The ASCII value of the first character in <cChars> which appears first in <cStr>

Description

Return the ascii value of a character in <cChars> which appears first in <cStr>.

Examples

```
? chr(GT_ChrFirst("sa ", "This is a test"))  // prints "s"
? chr(GT_ChrFirst("et",  "This is a test"))   // prints "t"
```

Status

Ready

Files

Library is libgt

GT_CHRTOTAL()

Find number of times a set of characters appears in a string

Syntax

GT_ChrTotal(<cChrs>, <cStr>) --> nTotOcc

Arguments

<cChrs> - The set of characters <cStr> - The string to search

Returns

<nTotOcc> - The number of times the characters specified in <cChrs> appears in <cStr>

Description

Returns the numnber of occurrences of characters belonging to the set <cChrs> in the string <cStr>. If no characters in <cChrs> appears in <cStr> GT_ChrTotal() will return 0.

NOTE: invalid parameters will return -1

Examples

```
local cStr1 := "the cat sat on the mat"

? GT_ChrTotal("tae", cStr1)           // prints 10
? GT_ChrTotal("zqw", cStr1)           // prints  0
```

Status

Ready

Files

Library is libgt

GT_STRCOUNT()

Count the number of times a substring appears in a string

Syntax

GT_StrCount(<cChrs>, <cStr>) --> nFreq

Arguments

<cChrs> - The substring to find the frequence of <cStr> - The string in which to find the character

Returns

<nFreq> - The number of times <cChrs> occurs in <cStr>

Description

GT_StrCount() counts how many times a specified substring appears in a string. If the substring does NOT appear in <cStr> this function will return 0. If the substring is a single character use GT_ChrCount() as it will be faster.

NOTE: invalid parameters will return -1

Examples

```
? GT_StrCount("the", "the cat sat on the mat")      // prints 2
```

Status

Ready

Files

Library is libgt

GT_STRCSPN()

Return length of prefix in string of chars NOT in set.

Syntax

GT_strcspn(<cString>, <cSet>) --> nLength

Arguments

<cString> - The string to find the prefix in <cSet> - The set of characters

Returns

<nLength> - The length of a string upto a character in the set

Description

Return the number of characters in the leading segment of a string that consists solely of characters NOT in the set.

Examples

```
? GT_strcspn("this is a test", "as ")      // prints 3
? GT_strcspn("this is a test", "elnjpq")   // prints 11
```

Status

Ready

Files

Library is libgt

GT_STRDIFF()

Return a string where it begins to differ from another

Syntax

GT_StrDiff(<cStr1>, <cStr2>) --> cRet

Arguments

<cStr1> - A character string to compare <cStr2> - The string to compare with

Returns

<cRet> - A string beginning at the position in <cStr2> where <cStr1> begins to differ from <cStr1>

Description

Return a string beginning at the position in <cStr2> where <cStr1> begins to differ from <cStr1>. If the two strings are identical (or identical upto the last character in <cStr2>) the function will return "".

NOTE: invalid parameters will return ""

Examples

```
? gt_strDiff("the cat", "the rat")          // prints "rat"
? gt_strDiff("the cat", "the ")             // prints ""
```

Status

Ready

Files

Library is libgt

GT_STREXPAND()

Insert fillers between characters in a passed string

Syntax

GT_StrExpand(<cStr>, [<nNum>], [<cChar>]) --> cRet

Arguments

<cStr1> - A character string to insert chars into <nNum> - The number of fill characters to insert (default 1) <cChar> - The fill chararacter (default space)

Returns

<cRet> - The input string with fill characters inserted between every character in the original.

Description

Inserts fill characters into a string.

NOTE: invalid parameters will return ""

Examples

```
? gt_strexpand("abc")                  // prints "a b c"
? gt_strexpand("abc", 2)               // prints "a  b  c"
? gt_strexpand("abc", 2, 'þ')          // prints "aþþbþþc"
```

Status

Ready

Files

Library is libgt

GT_STRLEFT()

Find length of prefix of a string

Syntax

GT_StrLeft(<cStr>, <cChars>) --> nLen

Arguments

<cStr> - The input string <cChars> - The set of characters to find

Returns

Description

Return the length of the leading segment in the passed string <cStr> that consists solely of the characters in the character set <cChars>.

If no characters in the the search set are found, the function shall return 0

Examples

```
? GT_StrLeft("this is a test", "hsit ")       // prints 8
? GT_StrLeft("this is a test", "hit a")       // prints 3
? GT_StrLeft("this is a test", "zxy")         // prints 0
```

Status

Ready

Files

Library is libgt

GT_STRPBRK()

Return string after 1st char from a set

Syntax

GT_StrpBrk(<cStr>, <cSet>) --> cString

Arguments

<cStr> - The input string <cSet> - The set of characters to find

Returns

<cString> - The input string after the first occurance of any character from <cSet>

Description

Return a string after the first occurance of any character from the input set <cSet>.

Examples

```
? GT_Strpbrk("This is a test", "sa ")  // prints "s is a test"
? GT_Strpbrk("This is a test", "et")   // prints "test"
```

Status

Ready

Files

Library is libgt

GT_STRRIGHT()

Find length of a suffix of a string

Syntax

GT_StrRight(<cStr>, <cChars>) --> nLen

Arguments

<cStr> - The input string <cChars> - The set of characters to find

Returns

<nLen> - The length of the prefix found.

Description

Return the length of the trailing segment in the passed string <cStr> that consists solely of the characters in the character set <cChars>.

If no characters in the the search set are found, the function shall return 0

Examples

```
? GT_StrRight("this is a test", "teas ")       // prints 8
? GT_StrRight("this is a test", "tes h")       // prints 5
? GT_StrRight("this is a test", "zxy")         // prints 0
```

Status

Ready

Files

Library is libgt