

String

Yung-Hsiang Lu

C String = Array of char

- examples of strings:
 - “Hello”
 - “Good Morning”
 - “I am learning C.”
- static string, created when a program is written
 - `char str1[] = “This is a string.”;`
 - `char * str2 = “Learning C Programming”;`
 - `char str3[] = {‘a’,‘b’,‘c’,‘d’,‘e’, ‘\0’}; /* equivalent to string “abcde” */`
 - Every string must end with a special character ‘\0’.**
- array of char, the string can be changed
 - `char str5[80];`

Functions for Strings

`strlen(str)` \Rightarrow length of a string (not counting `'\0'`)

`strcmp(str1, str2)` \Rightarrow compare two strings using
dictionary order (lexicographical order, alphabetic order)
0 if the two strings are the same

1 if `str1 > str2` (i.e. `str1` would appear later in a dictionary)

-1 if `str1 < str2` (i.e. `str1` would appear earlier)

`strcpy(dest, src)` \Rightarrow copy string `src` to `dest`

make sure dest has enough space

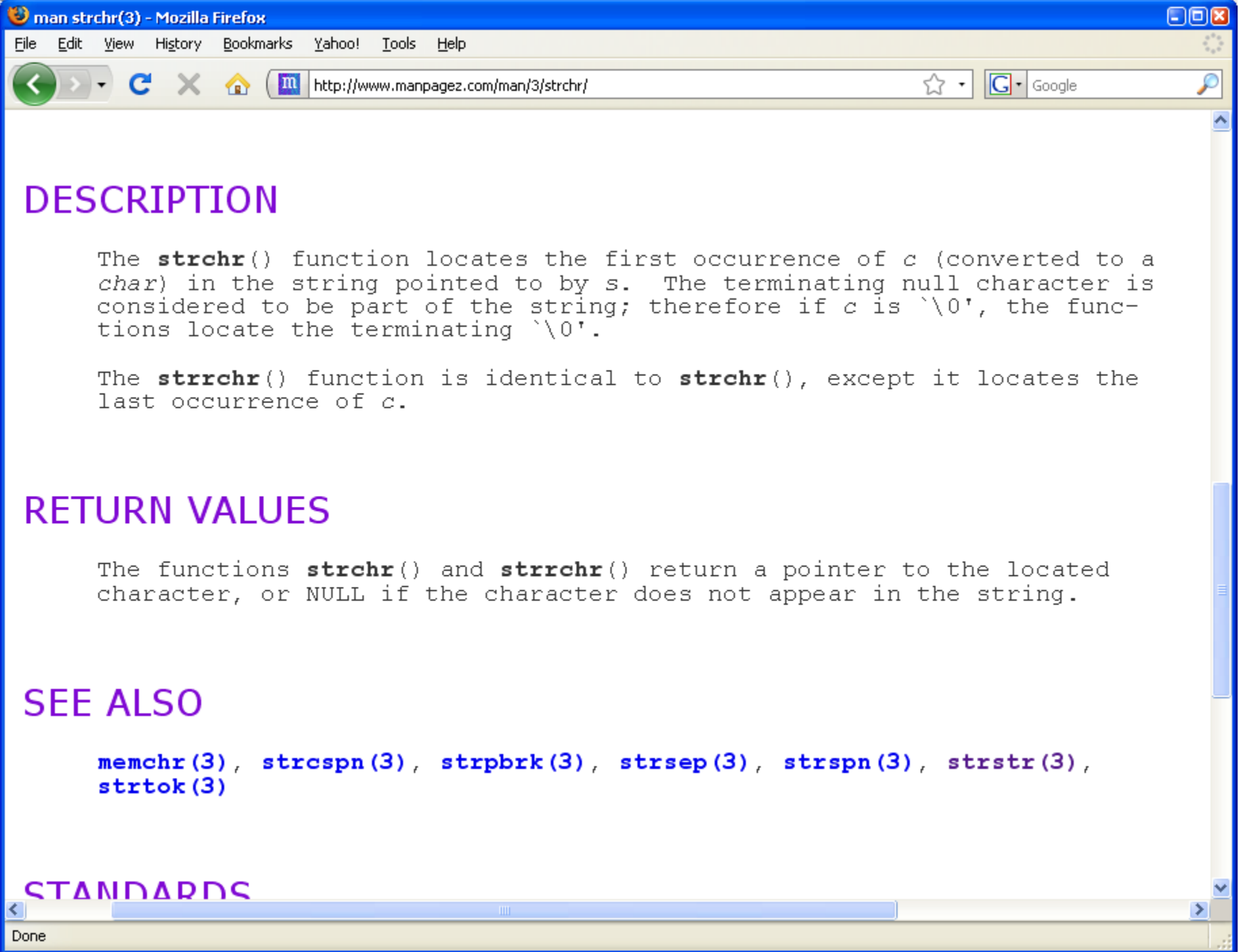
`strcat(dest, src)` \Rightarrow append `src` to `dest`
make sure `dest` has enough space

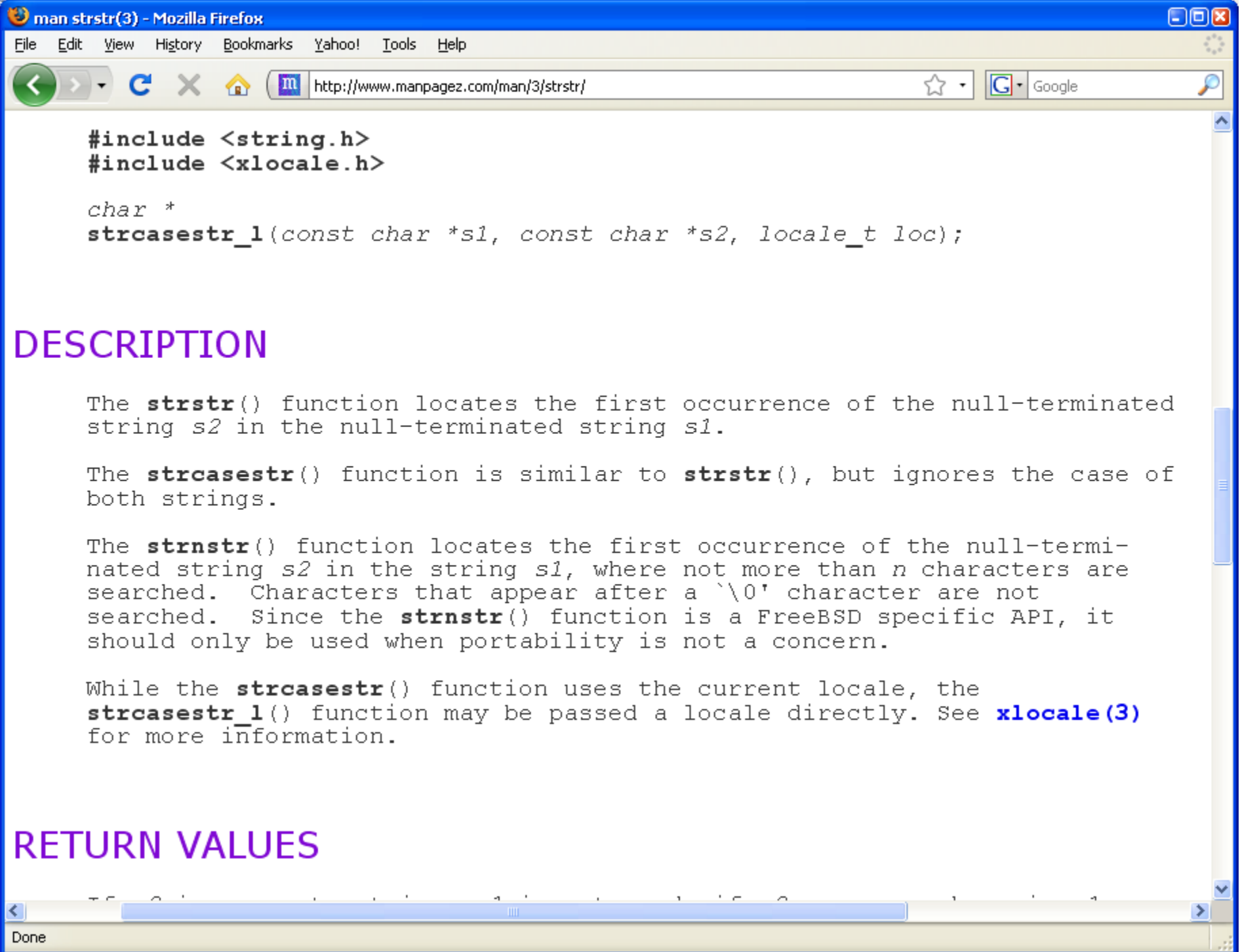
String and Character

`strchr(str, ch)` \Rightarrow return a pointer to the first occurrence of the character. If the string does not contain this character, return `NULL`

`strrchr(str, ch)` \Rightarrow return a pointer to the last occurrence of the character. If the string does not contain this character, return `NULL`

`strstr(str1, str2)` \Rightarrow return a pointer to the first occurrence of the second string. If the string does not contain this string, return `NULL`





1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
L e a r n i n g C P r o g r a m m i n g

string.c

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char * str1 = "Good Morning";
    char str2[] = "Learning C Programming";
    char str3[] = {'a', 'b', 'c', 'd', '\0'};
    printf("%s\n", str1);
    printf("strlen(str2) = %d\n", strlen(str2));
    printf("strchr(str3, 'c') = %s\n", strchr(str3, 'c'));
    char str4[80];
    strcpy(str4, str1);
    strcat(str4, str2);
    printf("str4 = %s\n", str4);
    printf("strstr(str4, \"ing\") = %s\n", strstr(str4, "ing"));
    return 0;
}
```

Console

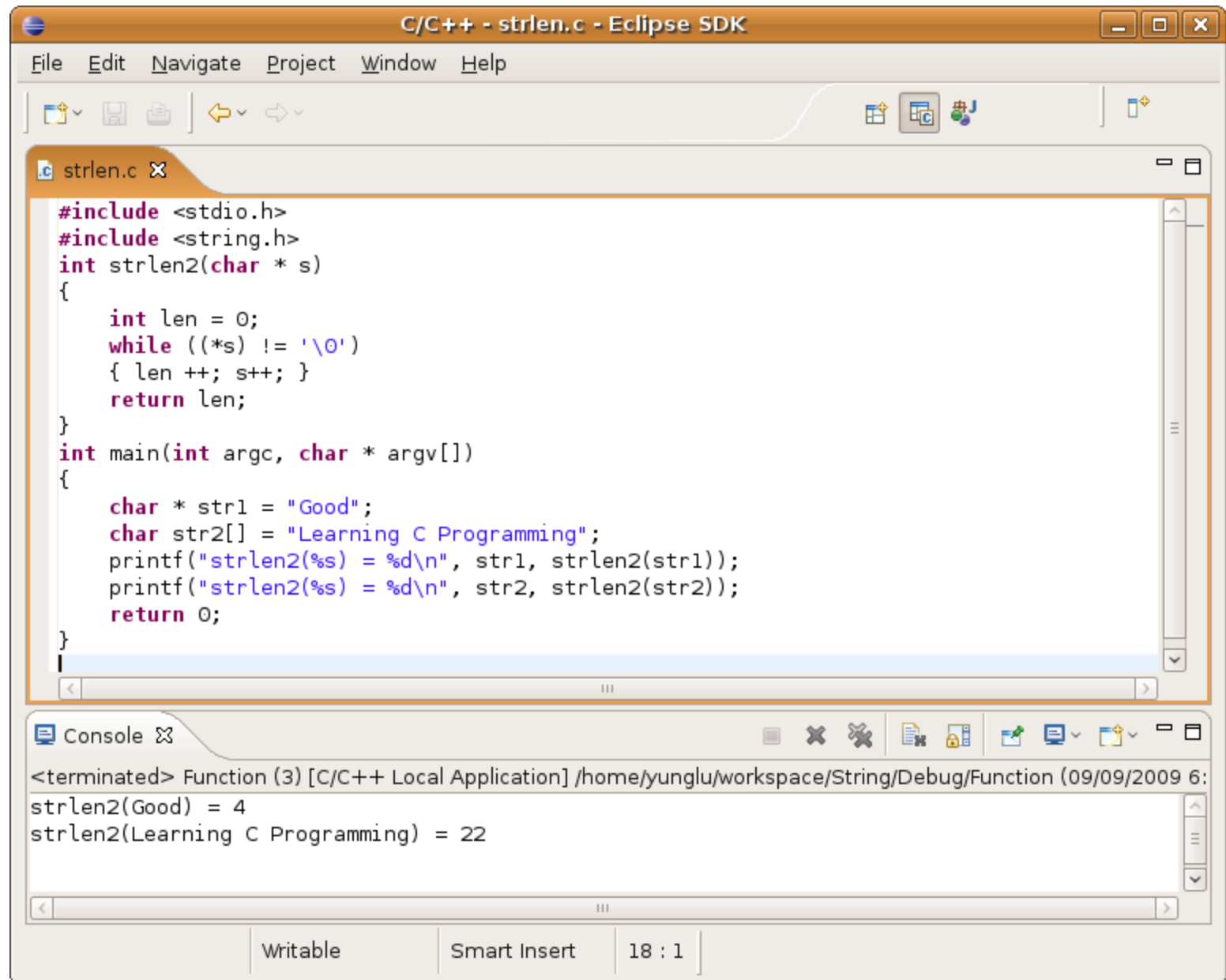
<terminated> Function (3) [C/C++ Local Application] /home/yunglu/workspace/String/Debug/Function (09/08/2009 11:23 PM)

```
Good Morning
strlen(str2) = 22
strchr(str3, 'c') = cd
str4 = Good MorningLearning C Programming
strstr(str4, "ing") = ingLearning C Programming
```

Writable

Smart Insert

18 : 1



Increment a Pointer

Diagram illustrating the memory layout and pointer incrementation in a C program.

The memory layout shows a string "Hello C Program" stored in memory, indexed from 0 to 14. The pointer `str1` points to index 0, and the pointer `sp` points to index 1.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
char	H	e	l	l	o		C		P	r	o	g	r	a	m

The Eclipse IDE screenshot shows the source code for `strlen.c`:

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char str1[] = "Hello C Program";
    char * sp = & str1[0];
    printf("%c\n", (* sp));
    sp ++;
    printf("%c\n", (* sp));
    sp += 8;
    printf("%c\n", (* sp));
    return 0;
}
```

The output window shows the first character 'H' printed, with a yellow arrow pointing to it.

Increment a Pointer

	str1	sp																	
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14				
char	H	e	l	l	o		C		P	r	o	g	r	a	m				

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char str1[] = "Hello C Program";
    char * sp = & str1[0];
    printf("%c\n", (* sp));
    sp ++;
    printf("%c\n", (* sp));
    sp += 8;
    printf("%c\n", (* sp));
    return 0;
}
```

Writable | Smart Insert | 14 : 1

<terminated> Function (3) [I
H
e
r

Increment a Pointer

The screenshot shows the Eclipse IDE with a C program named `strlen.c`. The program defines a character array `str1` containing the string "Hello C Program" and a pointer `sp` that points to the first element of `str1`. The program prints the string using `printf` and then increments the pointer `sp` by 8, printing the string again. The output window shows the results of the program execution.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
char	H	e	l	l	o		C		P	r	o	g	r	a	m

```
#include <stdio.h>
#include <string.h>
int main(int argc, char * argv[])
{
    char str1[] = "Hello C Program";
    char * sp = & str1[0];
    printf("%c\n", (* sp));
    sp ++;
    printf("%c\n", (* sp));
    sp += 8;
    printf("%c\n", (* sp));
    return 0;
}
```

Writable | Smart Insert | 14 : 1

What is the output?

```
char str1[] = "Nice Day."  
char * sp = str1;  
sp += 2;  
printf("%c\n", *(sp));
```

The output is .

Correct - Click anywhere to
continue

Incorrect - Click anywhere to
continue

Your answer:

You did not answer this question

You must answer the question
before continuing

Submit

Clear

String

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz