String HandlingFunctions

Examples with description

void *memchr(const void *str, int c, size_t n)

Searches for the first occurrence of the character c (an unsigned char) in the first n bytes of the string pointed to, by the argument str.

```
#include <stdio.h>
#include <string.h>

int main () {
   const char str[] = "Hello. All ."Techies;
   const char ch = '.';
   char *ret;

   ret = memchr(str, ch, strlen(str));

   printf("String after |%c| is - |%s|\n", ch, ret);

   return(0);
}
```

int memcmp(const void *str1, const void *str2, size_t n)

compares the first n bytes of memory area str1 and memory area str2.

```
#include <stdio.h>
#include <string.h>
int main () {
   char str1[15];
   char str2[15];
   int ret;
   memcpy(str1, "abcdef", 6);
   memcpy(str2, "ABCDEF", 6);
   ret = memcmp(str1, str2, 5);
   if(ret > 0) {
      printf("str2 is less than str1");
   } else if(ret < 0) {
      printf("str1 is less than str2");
   } else {
      printf("str1 is equal to str2");
   }

   return(0);
}
```

void *memcpy(void *dest, const void * src, size_t n)

The C library function void *memcpy(void *dest, const void *src, size_t n) copies n characters from memory area src to memory area dest.

```
#include <stdio.h>
#include <string.h>

int main () {
   const char src[50] = "Hello All Coders";
   char dest[50];
   strcpy(dest,"Heloooo!!");
   printf("Before memcpy dest = %s\n", dest);
   memcpy(dest, src, strlen(src)+1);
   printf("After memcpy dest = %s\n", dest);

   return(0);
}
```

void *memmove(void *str1, const void *str2, size_t n)

copies n characters from str2 to str1, but for overlapping memory blocks, memmove() is a safer approach than memcpy().

```
#include <stdio.h>
#include <string.h>

int main () {
   char dest[] = "oldstring";
   const char src[]  = "newstring";

   printf("Before memmove dest = %s, src = %s\n", dest, src);
   memmove(dest, src, 9);
   printf("After memmove dest = %s, src = %s\n", dest, src);

   return(0);
}
```

```
void *memset(void *str, int c, size_t
n)
```

copies the character c (an unsigned char) to the first n characters of the string pointed to, by the argument str.

```
#include <stdio.h>
#include <string.h>

int main () {
   char str[50];

   strcpy(str,"This is string.h library
function");
   puts(str);

   memset(str,'$',7);
   puts(str);

   return(0);
}
```

char *strstr(const char *haystack, const char *needle)

Function finds the first occurrence of the substring needle in the string haystack. The terminating '\0' characters are not compared.

```
#include <stdio.h>
#include <string.h>


int main () {
   const char haystack[20] = "My Favourite Book";
   const char needle[10] = "Favourite";
   char *ret;

   ret = strstr(haystack, needle);

   printf("The substring is: %s\n", ret);

   return(0);
}
```

char *strtok(char *str, const char *delim)

breaks string str into a series of tokens using the delimiter delim.

```
#include <string.h>
#include <stdio.h>

int main () {
   char str[80] = "Learning -With - Fun";
   const char s[2] = "-";
   char *token;

   /* get the first token */
   token = strtok(str, s);

   /* walk through other tokens */
   while( token != NULL ) {
      printf( " %s\n", token );

      token = strtok(NULL, s);
   }

   return(0);
}
```